



HIGHARC

EUVIP 2026 · LUXEMBOURG · ORAL SESSION IV

BIM-Native Tokenization for Constraint-Aware Room Layout Synthesis

Manuel Ladron de Guevara¹ · Jinmo Rhee² · Ardavan Bidgoli¹ · Vaidas Razgaitis¹ · Michael Bergin¹

¹ Higharc, United States ² University of Calgary, Canada

manuelladron.github.io/sbm



A master bedroom generated by DDEP inside a BIM room envelope



Layout synthesis in BIM is a representation problem

1 Outputs must stay BIM-editable

Every element keeps its category, hosting wall, parametric position, dimensions and rotation, not an image-space box.

2 Constraints are hard, not stylistic

Doors and windows live on walls; door swings and egress paths must stay clear; the inventory of a room type must be met.

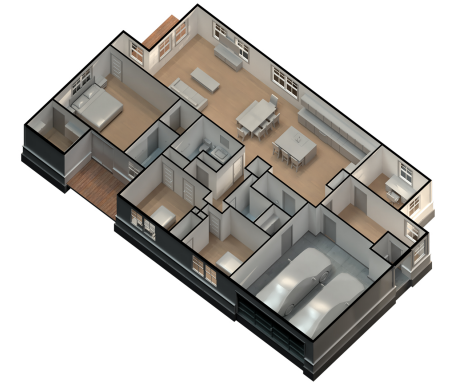
3 Strong 3D generators skip this

Voxel, mesh, point-cloud and image-conditioned diffusion models treat the scene as unstructured geometry: plausible, but hard to edit and easy to break.

Our claim. Make the representation sequence-native, mixed-type and wall-referenced; then a modestly sized Transformer becomes a strong, constraint-aware layout primitive.



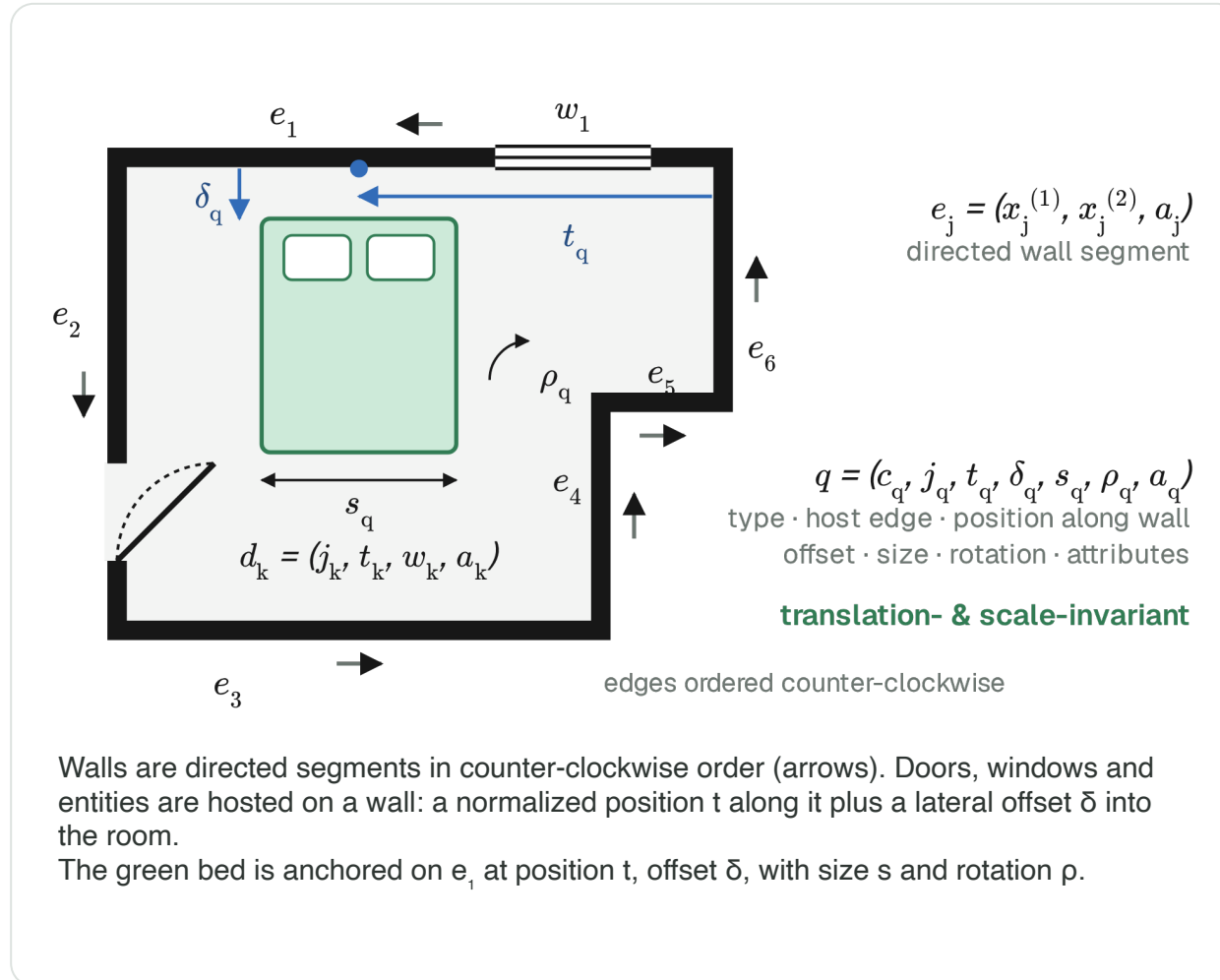
(a) Top-down View



(b) Axonometric View

Professional residential BIM: typed rooms, wall-hosted doors and windows, casework and wall-hosted furniture.

A room: A) envelope for encoder, B) contents for decoder



ENVELOPE AND CONTENTS

$$r_{\text{env}} = (y^{\text{topo}}, y^{\text{layout}}, \mathcal{E}, \mathcal{D}, \mathcal{W}), \quad r_{\text{ent}} = (\mathcal{P}, \mathcal{C})$$

WALLS AND OPENINGS

$$e_j = (x_j^{(1)}, x_j^{(2)}, a_j^{\text{wall}}), \quad d_k = (j_k, t_k, w_k, a_k^{\text{door}}), \quad t_k \in [0, 1]$$

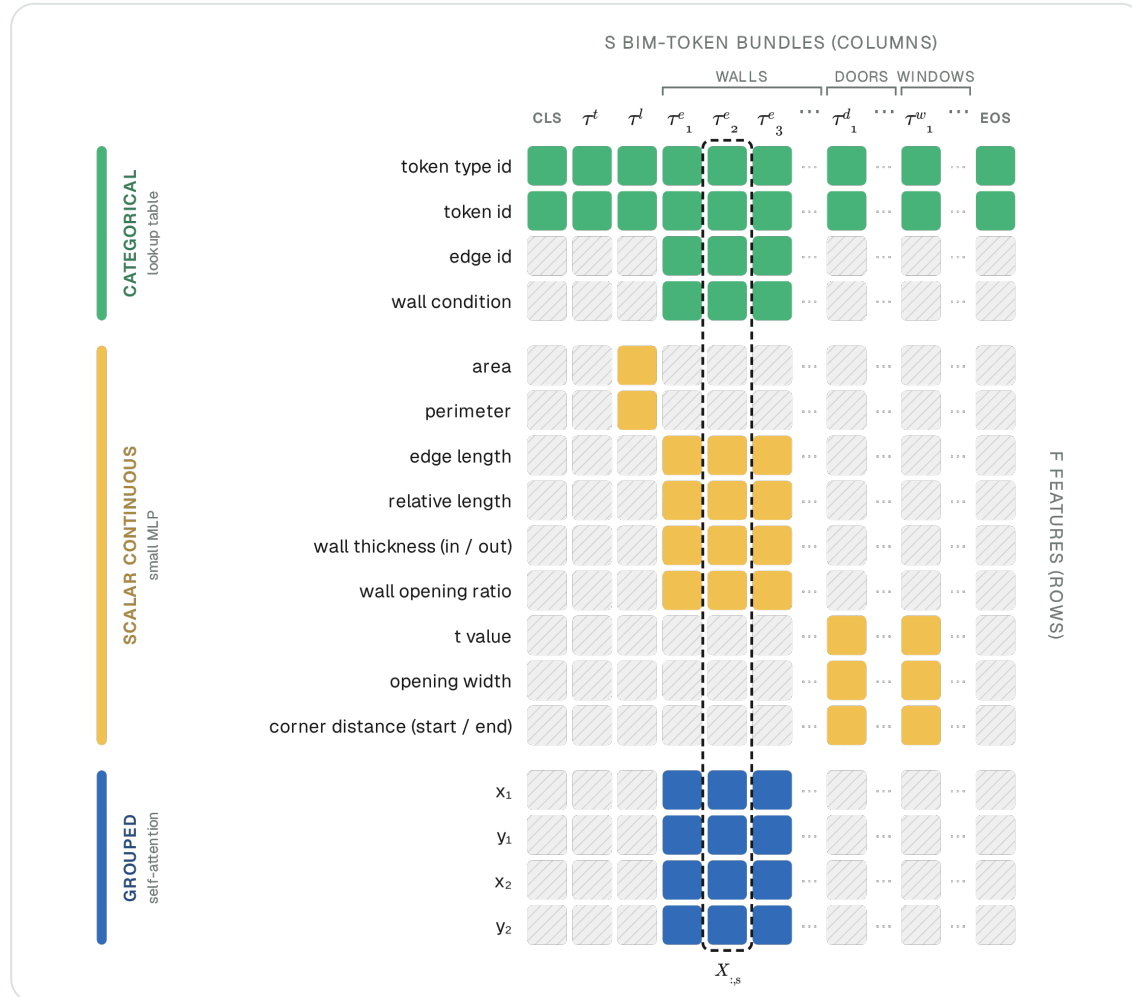
ENTITIES – PROPS AND CASEWORK

$$q = (c_q, j_q, t_q, \delta_q, s_q, \rho_q, a_q^{\text{ent}})$$

type · host wall · position along wall · lateral offset · size · rotation · attributes

Why wall-referenced? The same coordinates BIM tools already edit: invariant to absolute translation and scale, and every output token stays an editable object hosted on a wall.

BIM-Token Bundles: one sparse attribute-feature matrix per room



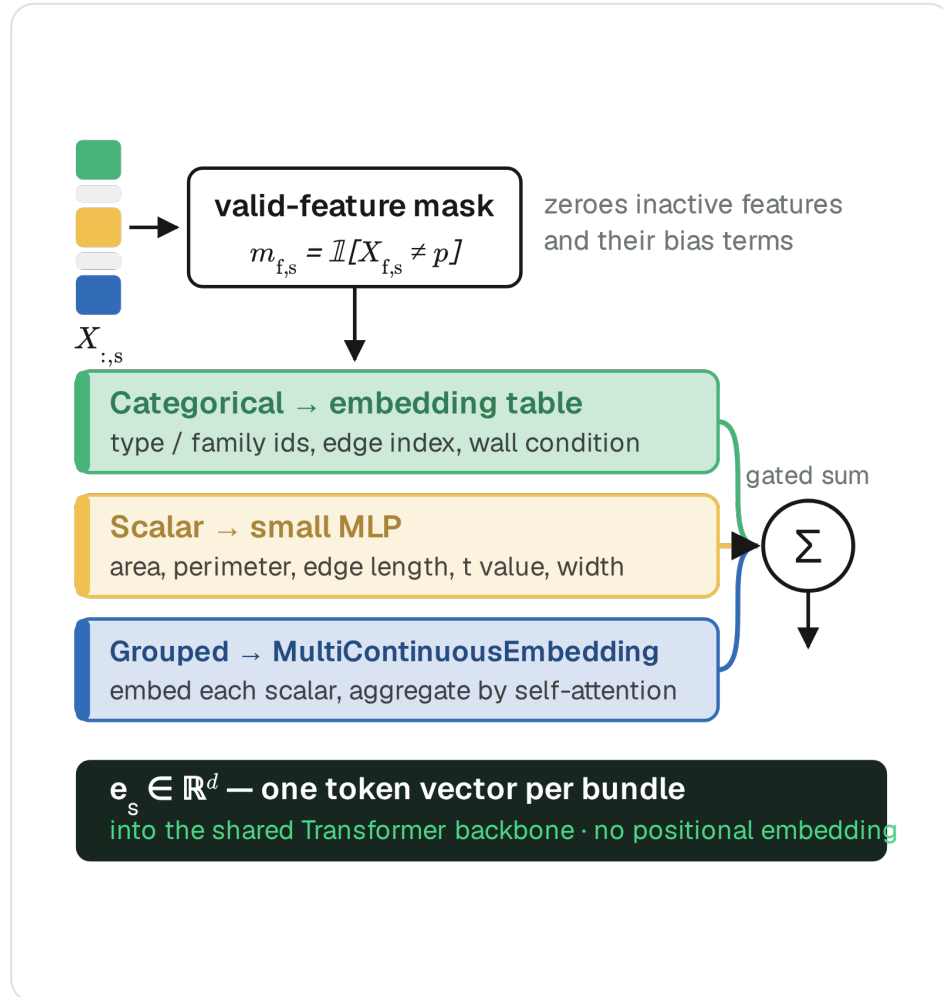
ONE COLUMN PER ELEMENT, ONE ROW PER FEATURE

- Columns are BIM-Token Bundles:** CLS, topology, layout, every wall, door and window, EOS. Each is a single logical element.
- Rows are features.** Only the rows that exist for that element type are active; the rest hold a sentinel $p = -100$ and are masked out.
- Rows come in three kinds: categorical ids, scalar continuous values, and correlated continuous groups such as the wall endpoints (x_1, y_1, x_2, y_2) .

$$X \in \mathbb{R}^{F \times S}, \quad X_{f,s} = p = -100 \text{ if feature } f \text{ is not defined for token } s$$

Not a flat serialization. A token stays a complete BIM element: its active rows are exactly the typed attributes you need to edit it later. Encoder and decoder use separate feature schemas. The entity bundles carry type, host wall, t , δ , size and rotation.

Mixed-type embedding: one dense vector per bundle



SUM OF MASKED FEATURE EMBEDDINGS

$$e_s = \sum_f m_{f,s} E_f(X_{f,s}) \in \mathbb{R}^d, \quad m_{f,s} = \mathbf{1}[X_{f,s} \neq p]$$

Categorical

type and family ids, edge index, wall condition → learnable embedding table

Scalar continuous

area, perimeter, length, t, width → small MLP

Grouped continuous

edge endpoints, opening corner distances → embedded jointly, so correlated geometry is learned together

GATED SUMMATION

$$e_s \leftarrow e_s + \sigma(W_g[e_s; \tilde{u}_f]) \odot \tilde{u}_f, \quad \tilde{u}_f = m_{f,s} E_f(X_{f,s})$$

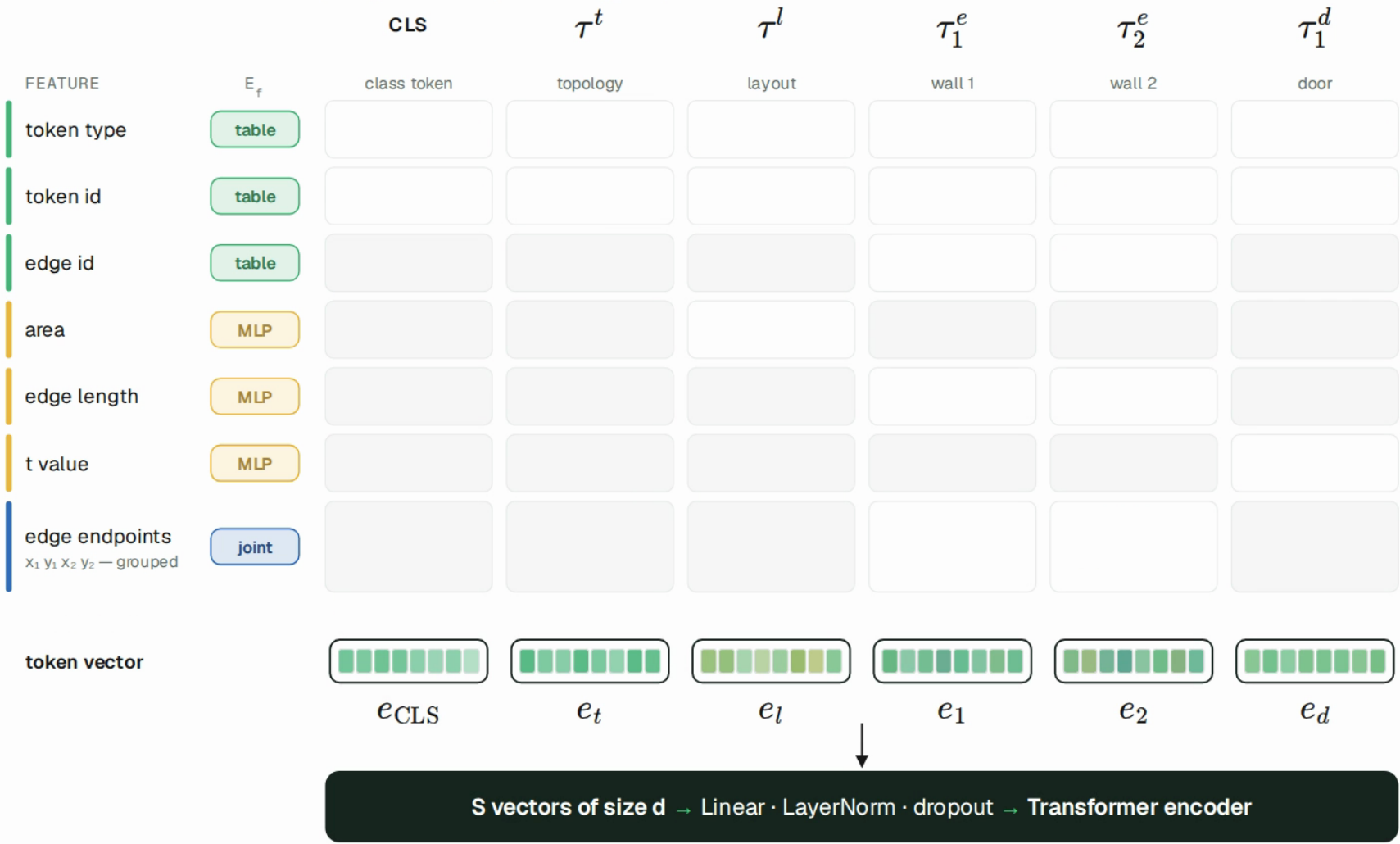
Each incoming feature embedding is scaled element-wise by a sigmoid gate computed from the running total. Inactive features are already zero, including bias terms, so the mask is exact. No positional embedding: geometry carries position.

RESULT

S tokens in, S vectors out — one per BIM element

The sum runs over features (rows), so the sequence length is preserved for the Transformer

TENSOR SHAPE
 $E \in \mathbb{R}^{S \times d}$



WHAT JUST HAPPENED

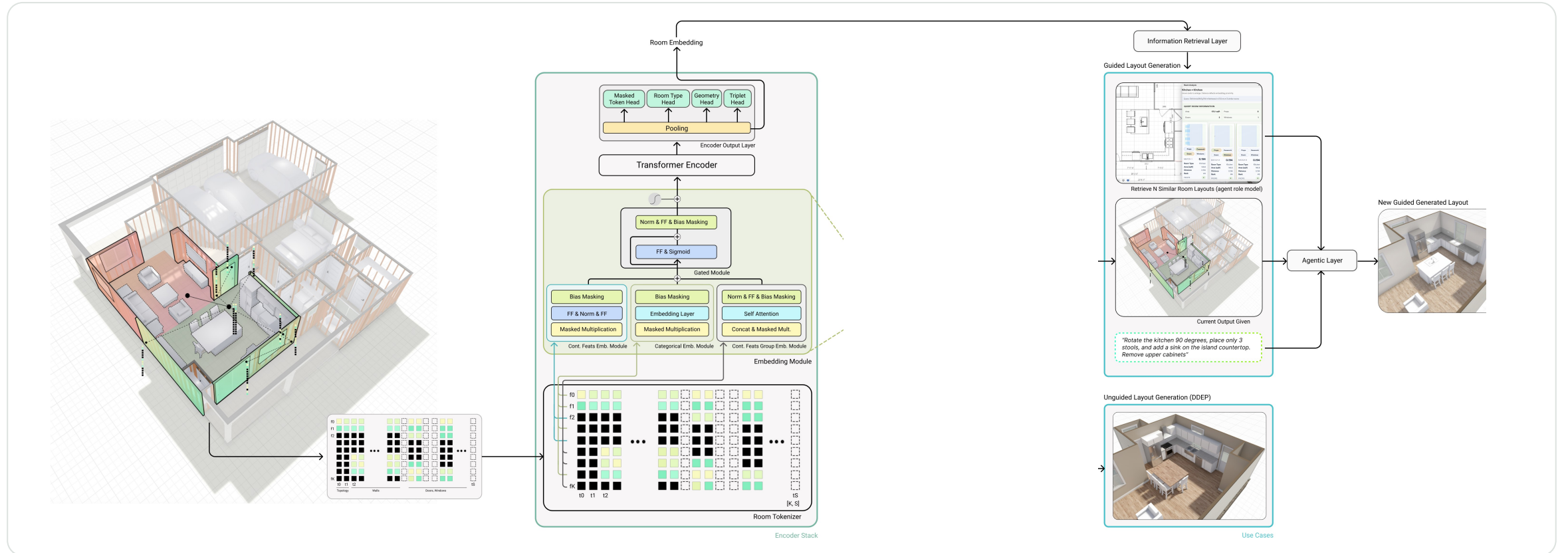
$$e_s = \sum_f \text{gate} \odot \tilde{u}_{f,s}$$

The sum runs **over features** (down a column), **never over tokens**.

$$F \times S \rightarrow F \times S \times d \rightarrow S \times d$$

So the sequence length S is preserved: one vector per wall, door, window and summary token.

One Transformer for two modes



ENCODER-ONLY — EMBEDDINGS & RETRIEVAL

$$M = \text{Enc}_\theta(E^{\text{enc}}) \in \mathbb{R}^{S_{\text{enc}} \times d}, \quad z(r_{\text{env}}) = M_0$$

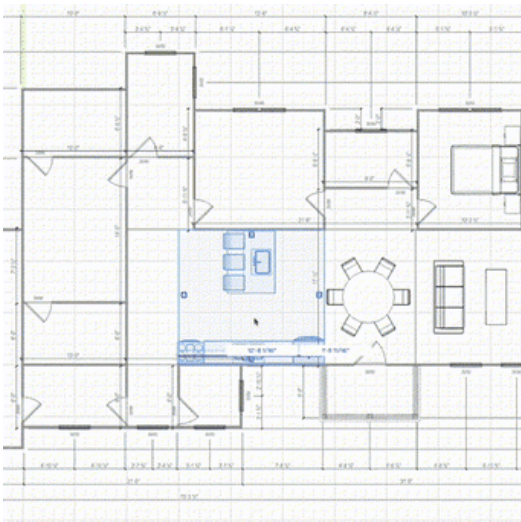
CLS-pooled room embedding z for clustering and retrieval; auxiliary room-type and masked-token heads.

ENCODER-DECODER — DATA-DRIVEN ENTITY PREDICTION (DDEP)

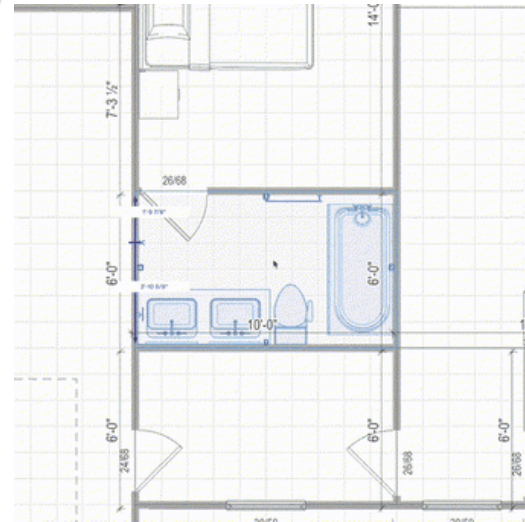
$$H = \text{Dec}_\theta(E^{\text{dec}}, M), \quad p_\theta(Y | r_{\text{env}}) = \prod_{t=1}^T p_\theta(y_t | y_{<t}, r_{\text{env}})$$

Next-entity prediction with cross-attention to the envelope memory M . Same tokenization, same weights.

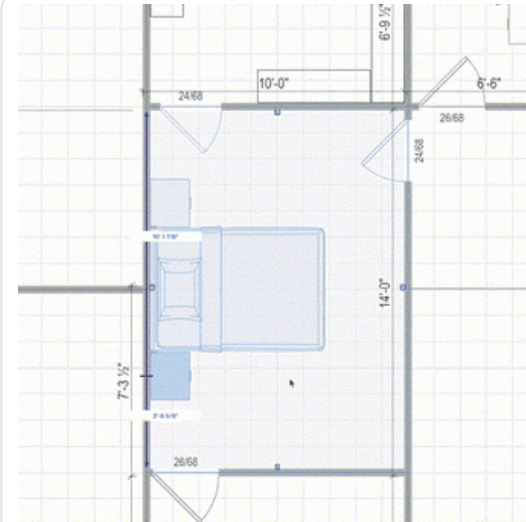
DDEP places entities one at a time — inside the design tool



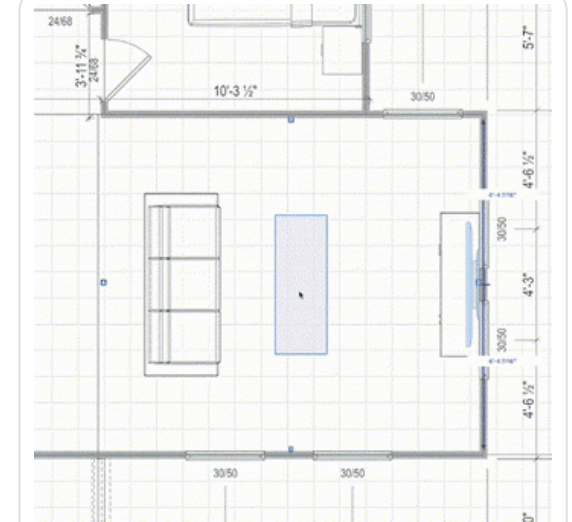
Kitchen



Full bath



Bedroom



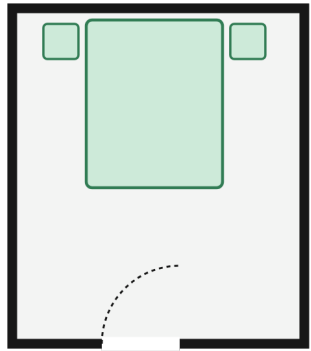
Living

Autoregressive, BIM-native output. Each step predicts type, host wall, t , δ , size and rotation, so every placed item is immediately a BIM object the designer can move or delete.

Constrained decoding. At inference a validity layer filters the candidate tokens — egress clearance, door-swing zones, BIM hosting rules. A guard over candidates, not a post-hoc layout optimiser.

How we score a layout — three views, read together

Coverage ↑



✓ bed (essential)

✓ nightstand ×2

✗ dresser

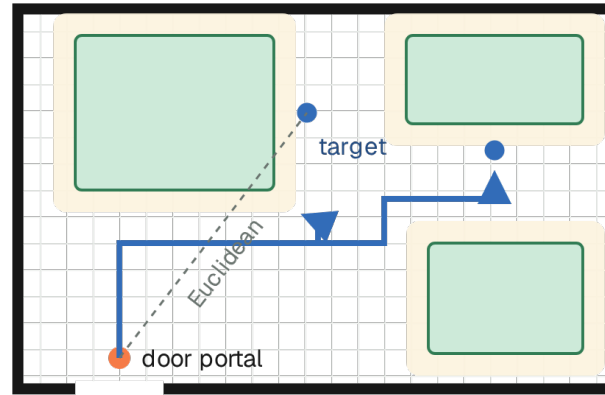
✗ bench (optional)

1.0 essential · 0.5 optional

$$s_i = 1 - \text{clip}(d_i + 0.5 o_i, 0, 1)$$

Inventory satisfaction. Missing fraction d_i and overfill o_i per category; essential items weigh 1.0, optional 0.5; alternative groups like tub-or-shower count once; hallucinated extras are penalised. Measures what was placed — not whether it is reachable.

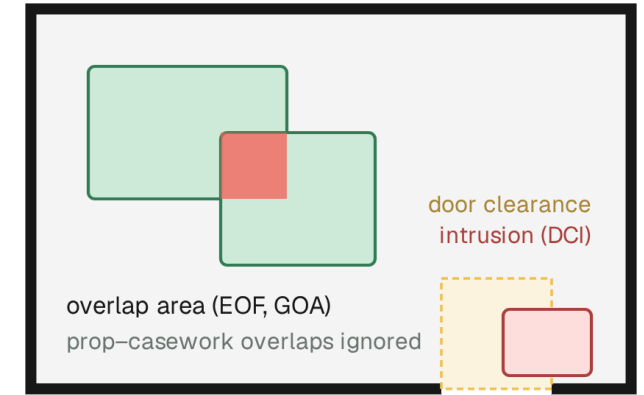
Navigability ↑



$$\text{Nav} = 100 (\text{SR} - 0.35 \text{DF})$$

SR: fraction of door-portal → essential-target pairs reachable by A* on a clearance-inflated grid. DF: capped detour of the path versus the straight line. One blocked doorway drops a room to the floor of the scale, -35.

Overlap-clearance ↓



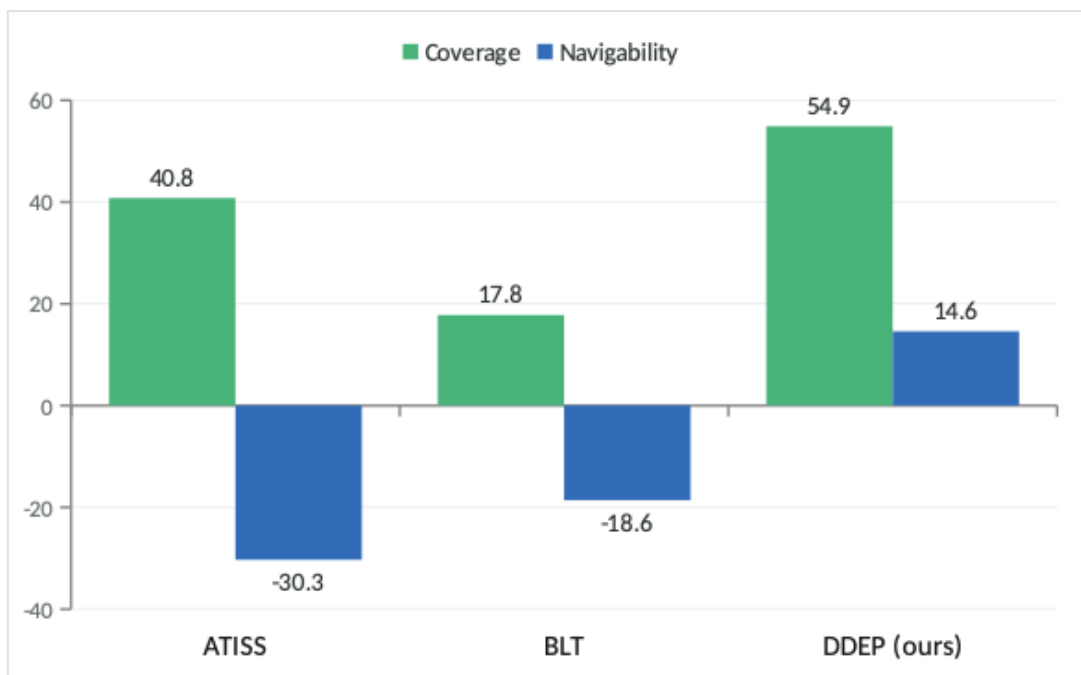
$$\text{OC} = 100 (0.5 \text{EOF} + 0.2 \text{GOA} + 0.3 \text{DCI})$$

On exact footprints: per-entity overlap with eligible neighbors (EOF), multiply-occupied floor area (GOA), door-clearance intrusion (DCI). Prop-casework overlaps are ignored: they usually encode support.

Same data, same budget, same harness: DDEP vs ATISS and BLT

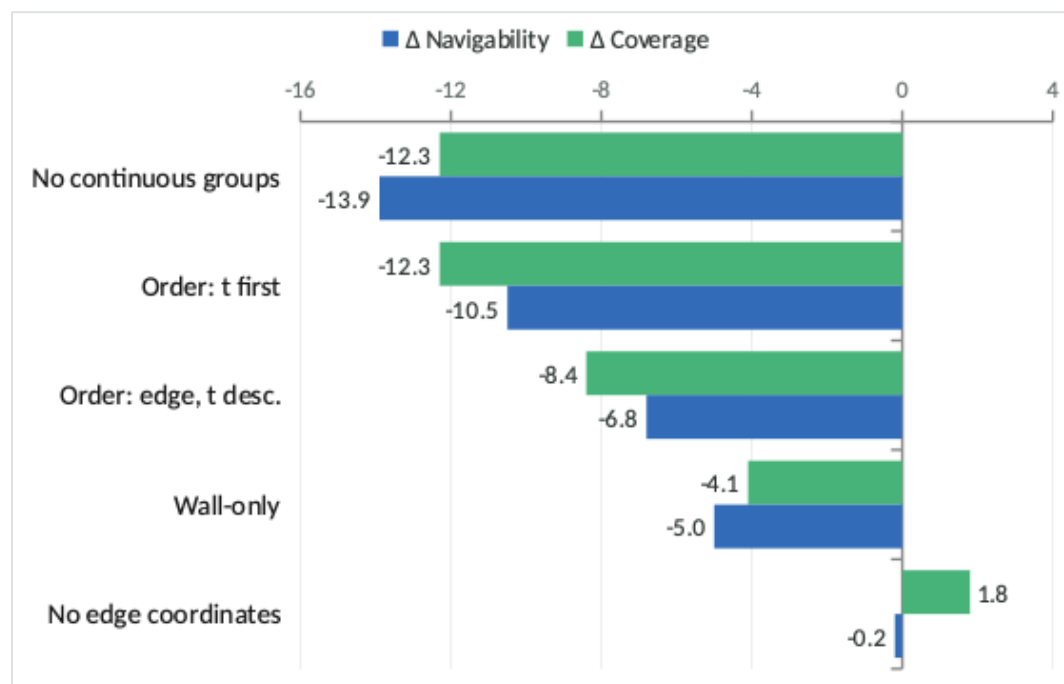
All three trained from scratch on one frozen split (15,740 / 2,009 / 1,826 rooms, 5 room types), 200 epochs, shared ontology; scored on the shared 1,225-room subset.

Coverage and navigability (higher is better)



SR 0.36 vs 0.03 / 0.12 · DF 0.60 vs 0.97 / 0.88 · OC 13.4 vs 0.5 / 35.7 — ATISS keeps overlaps near zero by leaving targets unreachable; BLT fails the opposite way.

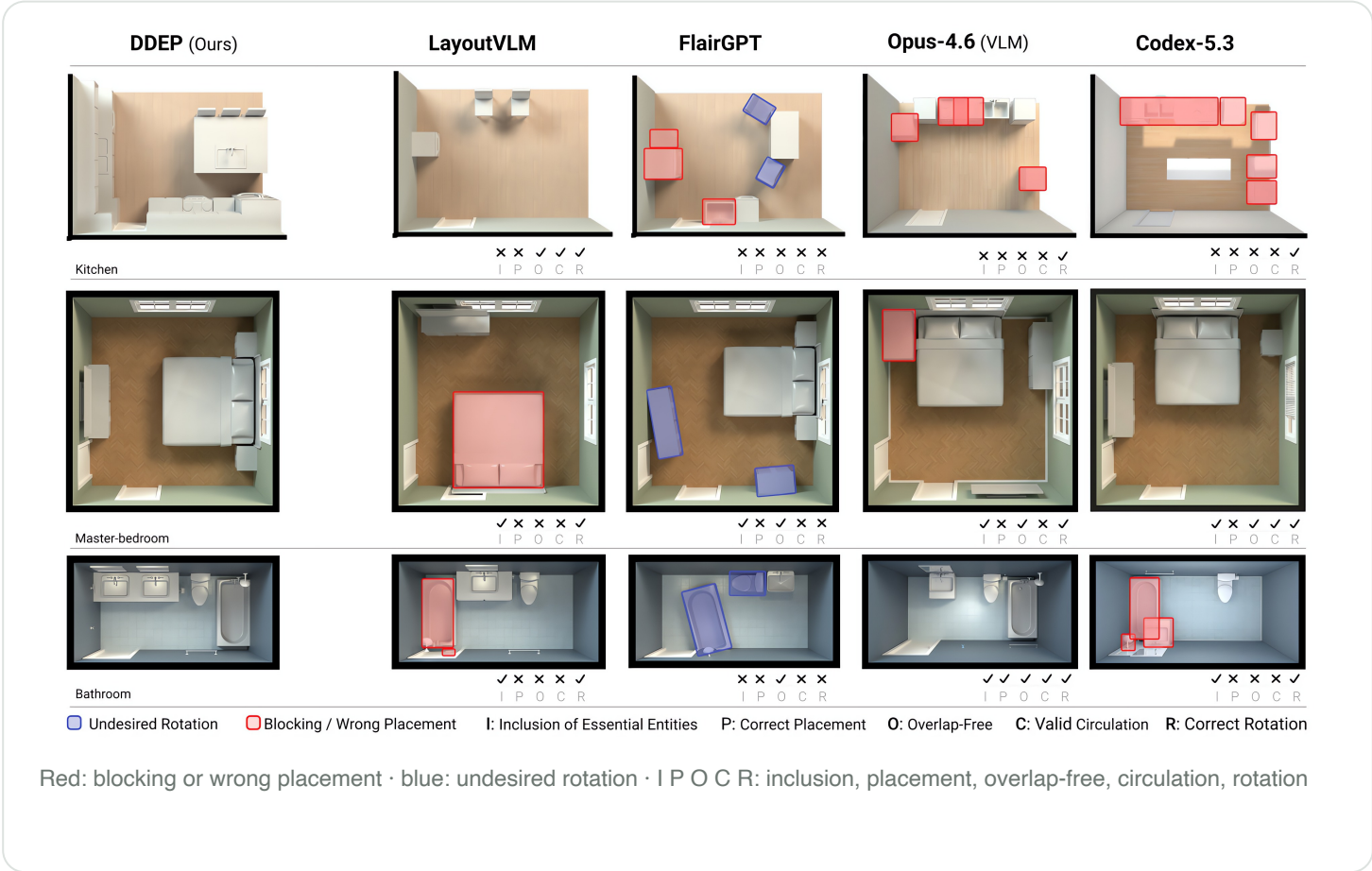
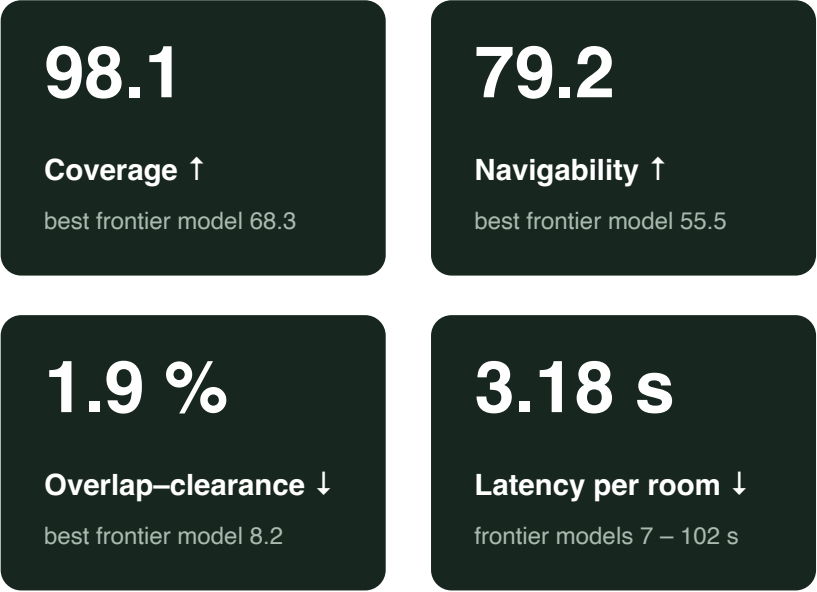
Ablations — change vs. full DDEP (points)



Removing the joint continuous-group embedding is the single most damaging change; entity ordering is second. Part of the ATISS/BLT gap may be conversion loss from bridging their outputs into the wall-referenced format.

Against zero-shot frontier LLMs, VLMs and domain-specific systems

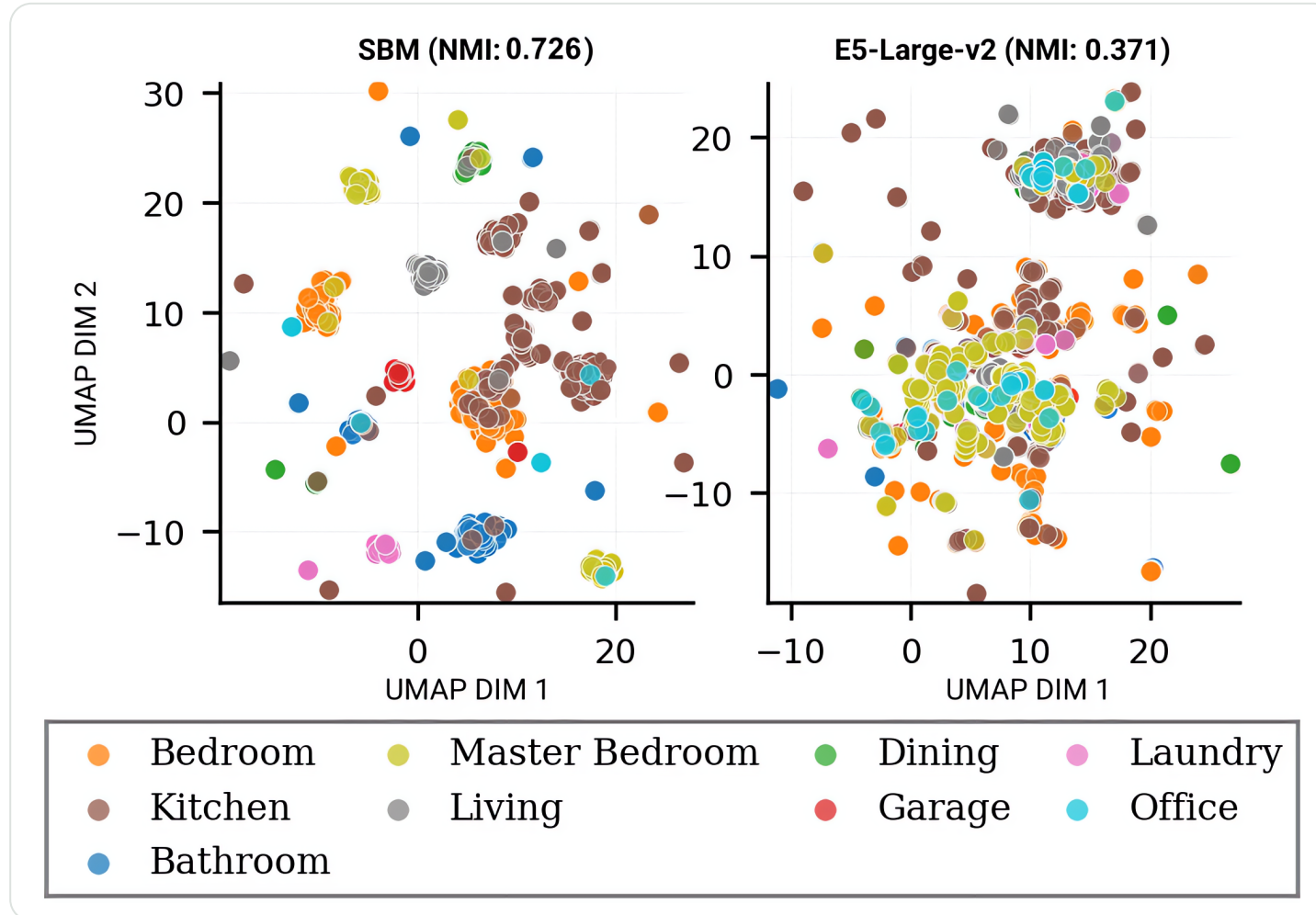
50 held-out production rooms. Every model receives the same tuned prompt, explicit coordinate system, wall-condition guidance, room-type-filtered entity catalogue and a matched output example.



Text LLMs: Claude Opus / Sonnet 4.6, Haiku 4.5, GPT-5.2, Gemini 3.1 Pro, Codex 5.3, Qwen 3.5. VLMs: the same families with the rendered floorplan. LayoutVLM: Nav 40.3, OC 3.8 (needs a furniture list). FlairGPT: Cov 46.6, Nav 28.2, OC 3.9.

A deployment floor, not an architectural ceiling: one run per room, no best-of-k or chain-of-thought, no in-domain fine-tuning of open models.

Encoder-only: room embeddings cluster by type



Against E5-Large-v2, BGE-Large-v1.5 and GTE-Large-v1.5 fed a serialized text description of the same room.

0.726

NMI

vs 0.371 for E5 — about 2x tighter type clusters

0.437

ARI

vs ≤ 0 for every text encoder

99.9 %

Triplet accuracy

type-constrained retrieval; 77 % entity overlap

The honest trade-off: text encoders keep the edge on within-type ranking (nDCG 86.7 vs 68.2). SBM wins on the property retrieval-augmented generation needs type structure.

What we hope you take away

Representation drives it

- Sparse attribute-feature columns, wall-referenced coordinates and joint embedding of correlated continuous features (not model scale) drive the gains.
- Every output token is an editable BIM element, so the model plugs into the design tool directly.

One backbone, two jobs

- Encoder-only for retrieval and clustering; encoder-decoder for constraint-aware generation with a validity guard at decode time.
- A ~50M-parameter model beats zero-shot frontier LLM/VLMs on layout quality at 3 s per room.

Limits and what is next

- Residential rooms from a proprietary corpus; the LLM/VLM track is trained vs. zero-shot.
- Ablations test within-tokenization choices: a flat-serialization baseline is the next experiment. Non-residential typologies, building scale and vertical constraints remain open.



HIGHARC

Thank you

Tokenize the building, not the pixels.

Paper, supplementary, video and per-model results are on the project page. Come find us at the poster session as well.

manuelrodriguez@higharc.com · jinmo.rhee@ucalgary.ca

Ladron de Guevara, Rhee, Bidgoli, Razgaitis, Bergin — Proc. 14th European Conference on Visual Information Processing (EUVIP), 2026



Project page · paper · video

manuelladron.github.io/sbm



GitHub · website repository

github.com/manuelladron/sbm