UniFPDesign: Unified Floorplan and Scene Generation via LLMs

Anonymous ACL submission

Abstract

The fragmentation between topological floorplan partitioning and local scene synthesis has long hindered end-to-end automated residential design. We argue that a truly functional space requires a holistic reasoning chain that aligns immutable structural constraints with diverse user personas. To bridge this gap, we formalize the task of **Constraint-Aware Unified Floorplan Design** and propose **UniFPDesign**, formulating the problem as "Geometry-as-Code" generation. To overcome the scarcity of aligned data, we introduce **Persona-Guided Specification Inversion**. By masking finalized layouts to reverse-engineer demands, this method utilizes structural invariants as implicit supervision and shifts from geometric rule-reversal to persona-centric synthesis. To eliminate spatial hallucinations, we develop a hierarchical CPT-SFT-RL strategy: CPT and SFT establish spatial syntax and instruction adherence, while GRPO ensures **Geometric Habitability**. Experiments on our proposed **UniFPDesign-2K** benchmark demonstrate superior performance in integrating structural integrity with functional placement.

1 Introduction

The automation of residential design necessitates translating abstract user needs into precise physical configurations. However, current research remains bifurcated: *floorplan generation* focuses on topological partitioning (Nauata et al., 2020; Yin et al., 2025a), while *scene generation* concentrates on furniture placement (Feng et al., 2023; Yang et al., 2025b). This fragmentation prevents the realization of end-to-end habitable design, which demands a holistic reasoning chain—from adhering to immutable structural constraints to interpreting complex user personas. To address this, we formalize the challenge of **Constraint-Aware Unified Floorplan Design** (see Figure 1).

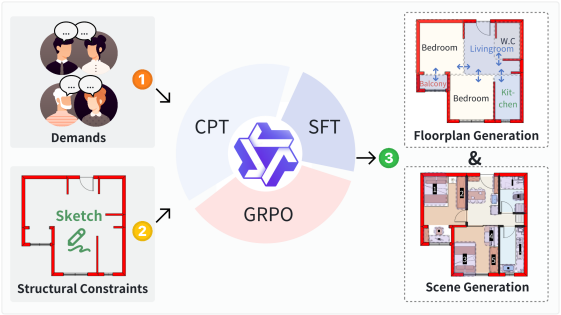

Figure 1: Overview of the proposed **UniFPDesign** framework. Our method unifies floorplan and scene generation by transforming demands and structural constraints into high-quality results via a three-stage training pipeline (CPT, SFT and GRPO) using LLMs.

Achieving this unification faces two fundamental bottlenecks. **First**, the "**Data Scarcity Paradox**": while user personas are the functional origin of design (Kuma, 1986), historical datasets rarely preserve aligned triplets of *(Structural Constraints, User Demands, Final Design)*. Most existing works attempt to reverse-engineer design rules from geometry alone, overlooking the "Human-Centric" origin of spatial logic. **Second**, the **Trade-off between Semantic Depth and Geometric Precision**: despite their semantic prowess, LLMs consistently struggle with rigorous geometric reasoning. Our investigation reveals a critical gap: while domain-specific models (e.g., GAN or Diffusion planners) effectively optimize structural partitions (Nauata et al., 2020, 2021), they lack the flexible reasoning to interpret open-ended personas. In contrast, general-purpose LLMs, though superior in instruction following, suffer from "spatial blindness"—failing to maintain collision-free boundaries or load-bearing integrity (Rudman et al., 2025; Feng et al., 2023). This trade-off necessitates a staged learning approach to bridge user demands with low-level physical validity.

To bridge these gaps, we present **UniFPDesign**,

a unified framework designed to reconcile user demands with geometric rigor. Our core insight is that *spatial configuration is the physical manifestation of resident personas*, a principle that allows us to treat geometric layouts as high-level semantic anchors.

Guided by this insight, we first address the data scarcity bottleneck through a **Persona-Guided Specification Inversion** pipeline. Unlike existing methods that rely on rigid rule-based geometry augmentation (Nauata et al., 2020) or simplistic template-based demand matching (Feng et al., 2023), our approach treats massive unlabeled finalized floorplans as "ground truth" and employs **masking-based inversion** to reverse-engineer latent user demands. By leveraging structural invariants as implicit supervision, this method not only ensures that synthesized demands are physically grounded but also enables the **automated scaling** of high-quality, aligned training triplets.

To further resolve the trade-off between semantic depth and geometric precision, we develop a hierarchical training strategy: **CPT** aligns the model with "Geometry-as-Code" syntax to establish coordinate consciousness; **SFT** maps instructions to spatial primitives for constraint adherence; and **RL** synergistically enforces geometry habitability. Our contributions are summarized as follows:

- We formalize **Constraint-Aware Unified Floorplan Design** and propose **UniFPDesign**, a framework that treats spatial design as a "Geometry-as-Code" generation task. This bridges the gap between structural partitioning and functional scene synthesis within a unified LLM reasoning chain.

- We introduce a **Persona-Guided Specification Inversion** pipeline to synthesize high-quality training triplets by reverse-engineering user demands from spatial layouts. We also release **UniFPDesign-2K**, the first unified benchmark for evaluating structural and functional habitability.

- We develop a robust training pipeline comprising CPT for domain alignment, SFT for instruction following, and Reinforcement Learning (RL). By leveraging GRPO, our approach synergistically optimizes geometry habitability.

2 Related Work

In the field of residential floorplan design, generating high-quality layouts has been a pivotal research direction. However, performing unified design under specific constraints remains a non-trivial task. We review existing methods across floorplan generation, spatial reasoning, mask modeling, and instruction synthesis to position our work.

2.1 Floorplan Generation

Early raster-based floorplan generation relied on Generative Adversarial Networks (Nauata et al., 2020, 2021). To enhance controllability and topological consistency, recent research has shifted toward diffusion-based methods for geometric precision (Shabani et al., 2023; Chen et al., 2023; Qin et al., 2024) and discrete denoising for topological learning (Su et al., 2024; Zeng et al., 2024). More recently, Large Language Models (LLMs) have been leveraged to address numerical constraints and semantic alignment: frameworks like DStruct2Design (Luo et al., 2024) and ChatDesign (Li et al., 2024a) enable iterative refinement from natural language, while other approaches facilitate interactive workflows via next-room prediction (Yin et al., 2025a) and bubble-diagram augmentation (Wei and Li, 2024).

2.2 Scene Generation & Spatial Reasoning

LLMs have been successfully adapted as visual planners, progressing from simple object placement to the integration of semantic planning and geometric reasoning. In the domain of semantic layout synthesis, LayoutGPT (Feng et al., 2023) ensures 2D spatial fidelity, while Holodeck (Yang et al., 2024) and OptiScene (Yang et al., 2025b) extend these capabilities to 3D asset retrieval and construction. However, these methods are primarily restricted to unconstrained or single-room scenarios, neglecting the rigid structural constraints, such as preserving load-bearing walls, which is critical to floorplan design.

To align semantic generation with geometric precision, recent approaches incorporate explicit constraints ranging from differentiable optimization (Sun et al., 2025) to parametric generation (Wang et al., 2025). These mechanisms address ambiguities in model reasoning, where perspectives diverge: while Ivanitskiy et al. (Ivanitskiy et al., 2023) suggest models possess emergent topological representations, (Rudman et al., 2025) argue they lack

innate visual reasoning, necessitating structured guidance.

2.3 Mask Modeling for Structural Completion

Mask modeling facilitates self-supervised structure completion by learning layout representations through reconstruction. In topological generation, MaskPlan (Zhang et al., 2024a) utilizes graph masking to infer global adjacencies. In the raster domain, FloorPlanMAE (Yin et al., 2025b) captures spatial and load-bearing structures via patch reconstruction, while FlexCAD (Zhang et al., 2024b) extends this paradigm to CAD sequences to model construction dependencies. Collectively, these studies demonstrate the efficacy of mask-based reconstruction for structural inference and data synthesis in design contexts.

2.4 Instruction Synthesis & Persona Modeling

Instruction back-translation provides a practical way to synthesize supervision when paired *(user demand, floorplan)* data is scarce (Li et al., 2024b). Persona-conditioned generation has been explored to scale synthetic diversity and model user-specific patterns (Chan et al., 2024; Sengupta et al., 2023). Our persona hub is additionally motivated by evidence that living environments reflect household traits and life stage (Gosling et al., 2002; South et al., 2016; Kuma, 1986). We also draw inspiration from perturbation-based augmentation for robustness (Kaushik et al., 2020). In contrast to prior work, we integrate **layout-grounded** back-translation, a **sociologically informed** persona hub, and **verified wishlist injection** that encodes optional improvements as concessive constraints under feasibility.

Limitations of Prior Art. Despite substantial progress, existing literature suffers from three primary limitations: (1) *Task Fragmentation*: Most works are restricted to either topological room partitioning (Nauata et al., 2020, 2021) or isolated scene synthesis (Feng et al., 2023), failing to address the holistic reasoning required for **Constraint-Aware Unified Floorplan Design**. (2) *Static Data Paradigms*: Current datasets rely heavily on rigid rule-based augmentation or simplistic template-matching, which overlooks the "Human-Centric" origin of design. Consequently, they fail to capture the nuanced link between diverse resident personas and complex spatial configurations, limiting the **scaling** of demand-driven training data. (3)

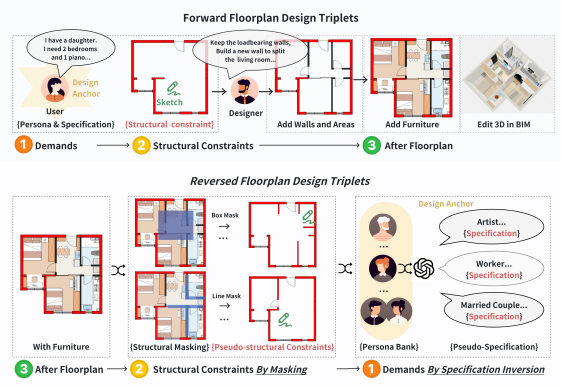

Figure 2: Construction of Forward and Reversed Floorplan Design Triplets. The forward process (top) maps demands to layouts, while the reversed process (bottom) synthesizes pseudo-data via structural masking and specification inversion.

Semantic-Geometric Disconnect: General-purpose LLMs lack the coordinate consciousness to satisfy rigid load-bearing and collision-free constraints, while domain-specific generative models lack the semantic depth to interpret open-ended user demands. A unified framework that synergistically optimizes persona alignment, structural integrity, and functional habitability remains absent.

3 The UniFPDesign

3.1 Problem Formulation

We formalize floorplan design as a *Constraint-Aware Unified Floorplan Design* problem. Unlike generation-from-scratch, floorplan design requires mapping a partial structural state to a complete semantic layout under strict physical boundaries. Let a floorplan be defined as a set of geometric entities $\mathcal{X} = \mathcal{S} \cup \mathcal{F}$, where $\mathcal{S}$ represents immutable structural elements (e.g., load-bearing walls, shafts) and $\mathcal{F}$ denotes designable functional units (e.g., partition walls, furniture).

Given an initial structural skeleton $\mathcal{S}_{src}$ and user demands $\mathcal{I}$, our goal is to synthesize an optimal target layout $\mathcal{X}^*$. We model this as maximizing the conditional likelihood parameterized by θ, subject to hard structural constraints:

$$\begin{aligned} \mathcal{X}^* &= \underset{\mathcal{X}}{\operatorname{argmax}}\, P_\theta(\mathcal{X} \mid \mathcal{S}_{src}, \mathcal{I}) \\ \text{s.t.} \quad & \mathcal{S}_{src} \subseteq \mathcal{X}, \mathcal{V}(\mathcal{X}) = 1 \end{aligned} \tag{1}$$

where $\mathcal{V}(\mathcal{X})$ acts as an indicator function for topological correctness (e.g., closed room loops, collision-free placement). We solve this optimization by casting $\mathcal{X}$ as a sequence of executable code

tokens, transforming the spatial planning problem into a constraint-aware autoregressive generation task.

3.2 Dataset Construction

(1) Persona-Guided Specification Inversion. Paired $(\mathcal{I}, \mathcal{X})$ supervision is scarce, as final layouts reflect compromises. Inspired by instruction back-translation (Li et al., 2024b), we synthesize demands from finalized layouts, grounded in the insight that *spatial configuration is the physical manifestation of resident personas* (Kuma, 1986). We propose a three-stage inversion pipeline: fact extraction → persona conditioning → wishlist injection.

Fact extraction. Given a floorplan $\mathcal{X}$, we deterministically extract grounded facts C_{fact} (e.g., room types, adjacency) from geometry. These facts act as anchors to prevent hallucination and ensure synthesized demands remain consistent with the layout support.

Persona hub. To model the one-to-many mapping of layouts to lifestyles, we build a persona hub (Appendix A) parameterizing household composition and life stage (Gosling et al., 2002). Personas serve as controllable priors for demand synthesis, guiding the inference of unmet needs beyond physical facts ($P(\mathcal{I} \mid \mathcal{X}, \text{Persona})$), while remaining consistent with layout constraints.

Wishlist injection. Real users often express preferences that cannot be fully realized due to spatial or structural constraints. Our goal is to achieve high alignment with real user demands while explicitly modeling how aspirational preferences are negotiated and potentially compromised in practice. Guided by the sampled persona, we propose persona-consistent wishlist items as candidate demand augmentations, typically reflecting improvement-oriented desires (e.g., a kitchen island or bar counter). Each injected item is then verified against the extracted layout facts C_{fact}. If an item conflicts with physical constraints, it is injected as a concessive constraint rather than a contradiction: the preference is explicitly acknowledged but marked as secondary to physical feasibility. Importantly, this prioritization does not override users' core living necessities but reflects the common trade-off where *optional or atmosphere-enhancing demands* are relaxed after basic functional needs are satisfied. We further apply two lightweight checks—persona-demand coherence and option perturbation (Kaushik et al., 2020)—to reduce spurious injections and ensure stable demand reasoning (see Figure 3).

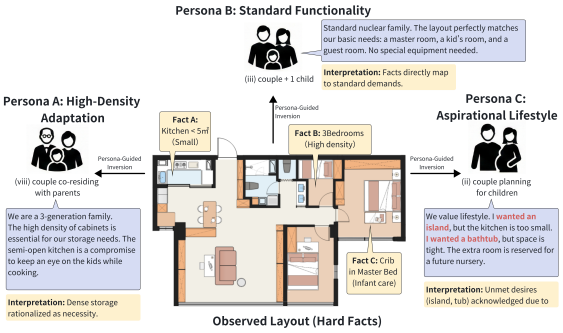

Figure 3: **Qualitative analysis of Persona-Guided Demand Synthesis.** Given identical layout facts, distinct personas induce divergent narratives: the *High-Density Adaptation* persona rationalizes constraints as functional necessities (left), while the *Aspirational Lifestyle* persona explicitly articulates unmet desires (in red) rejected by spatial feasibility (right).

Demand Synthesis Setting	Jaccard
Layout-only inference (baseline)	0.643
+ Persona hub	0.764
+ Wishlist injection + verification (full)	0.816

Table 1: Ablation on demand alignment measured by Jaccard similarity between synthesized and real user requirement questionnaires.

Ablation. As shown in Table 1, we measure user demand alignment using the Jaccard similarity between synthesized questionnaires and real user demand questionnaires. As final layouts are the result of practical compromises, inferring user demand solely from realized layouts yields limited alignment (0.643). Conditioning on a persona hub significantly improves alignment (0.764), and adding wishlist injection with verification further narrows the gap to real user demands (0.816).

(2) Building the Structural Masking Input $\mathcal{S}_{src}$: We create self-supervised training pairs by corrupting a complete floorplan to obtain a partial structural skeleton as input. Following our formulation, $\mathcal{X} = \mathcal{S} \cup \mathcal{F}$, where $\mathcal{S}$ are immutable structures (e.g., load-bearing walls, shafts) and $\mathcal{F}$ are designable elements (e.g., partitions, furniture). We sample $\omega \sim \pi(\omega)$ and construct

$$\tilde{\mathcal{X}} = \mathcal{C}(\mathcal{X}; \omega), \qquad \mathcal{S}_{src} = \Pi_{\mathcal{S}}(\tilde{\mathcal{X}}), \qquad (2)$$

where $\Pi_{\mathcal{S}}(\cdot)$ projects a plan onto its structural subset. We train a denoising model to reconstruct the full plan from $\mathcal{S}_{src}$ (Vincent et al., 2008; Devlin

et al., 2019; He et al., 2022; Pathak et al., 2016):

$$\max_{\theta} \log P_\theta(\mathcal{X} \mid \mathcal{S}_{src}). \tag{3}$$

In practice, $\mathcal{C}$ deletes a subset of designable inner walls via either region masking or rule-guided wall-chain removal, followed by a deterministic repair to enforce representation validity (i.e., $\mathcal{V}(\tilde{\mathcal{X}}) = 1$; Appendix B). At inference (and supervised fine-tuning), we additionally condition on user demand $\mathcal{I}$.

(3) UniFPDesign-2K Benchmark. Unlike existing benchmarks that decouple renovation from furnishing, UniFPDesign-2K addresses their inherent coupling in real-world design. It serves as a high-quality testbed for Constraint-Aware Unified Floorplan Design.

Composition & Complexity. Comprising **2,000** millimeter-precision samples in a structured Code format, the benchmark supports (a) Renovation, (b) Furnishing, and (c) Unified End-to-End tasks. It presents a high-density challenge, averaging 7.34 functional zones and 28.23 furniture items per plan, alongside detailed structural models (avg. 7.14 doors, 6.61 windows), forcing models to maintain long-horizon consistency beyond simple object placement.

Diversity & Logic. The dataset prevents overfitting via a 20.7% long-tail distribution of complex typologies (e.g., multi-terrace layouts) beyond standard configurations. Quantitative analysis validates robust architectural logic, evidencing realistic "Service-Served" flows (e.g., 86.4% Kitchen-Living connectivity) and privacy buffers.

3.3 Continual Pre-Training (CPT)

General-purpose LLMs often struggle to comprehend the specialized vector representations required for floorplan design. To bridge this domain gap, we employ CPT, which effectively aligns the model with these geometric code constraints (Qi et al., 2025).

We conduct CPT on a ∼40B-token corpus derived from approximately 10 million floorplans, incorporating two key strategies: Geometry-as-Code, which represents floorplans as code to leverage the spatial reasoning of LLMs for floorplan design, and a **Data Mixing Strategy** that combines mathematical (Fujii et al., 2025) and general data to mitigate catastrophic forgetting. (See Appendix C).

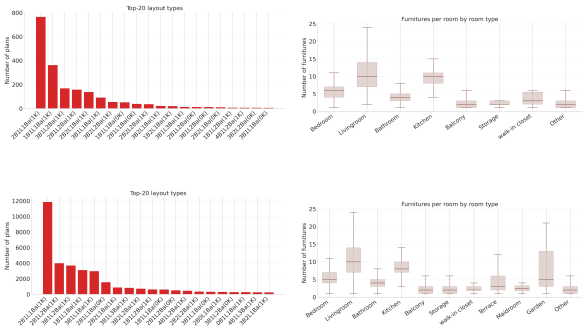

Figure 4: **Dataset statistics for UniFPDesign. Top row:** UniFPDesign-2K Benchmark split. **Bottom row:** training split. **Left column:** distribution of common residential program combinations (shown as #Bedrooms, #Living rooms, #Bathrooms, #Kitchens). **Right column:** distribution of the number of furniture items per room, grouped by room type, reflecting furnishing complexity and long-tail variability across functional spaces.

3.4 Supervised Fine-Tuning (SFT) with Feature Enhancement

Despite acquiring geometric priors via CPT, the model lacks alignment with the complex semantic constraints of design. We address this by employing SFT on a curated real-world floorplan corpus comprising approximately 34,000 floorplans (See Figure 4 bottom).

(1) Bounding Box Constraints: To mitigate spatial conflicts such as furniture overlap and doorway occlusion, we abstract functional entities (e.g., furniture, doors) as Bounding Boxes. By explicitly modeling these boundary coordinates, the model captures precise spatial relationships, ensuring valid inter-object and object-to-door relative positioning.

(2) Graph Planning CoT (Bubble Diagram): Direct floorplan design frequently suffers from illogical topology (e.g., placing bathrooms at main entrances) and implausible sizing. Therefore, we introduce a coarse-to-fine approach operating in two steps (see Figure 5). In the *graph planning* step, the model first predicts a topological adjacency graph (i.e., Bubble Diagram (Nauata et al., 2020; Luo et al., 2024)) to establish high-level spatial logic. Subsequently, conditioned on this structural prior, the model generates specific *spatial actions* as the second step to construct the detailed layout, ensuring adherence to the planned constraints.

(3) Sketch-Derived Point Constraints: Floorplan design presents dual challenges: preserving structural rigidities imposed by piping in functional

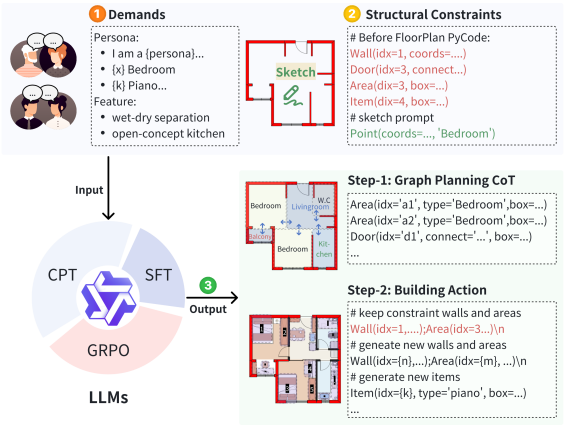

Figure 5: The training and inference pipeline of the proposed UniFPDesign. LLMs encode inputs into structured PyCode prompts and execute a two-stage generation: Step-1 Graph Planning CoT for spatial reasoning and Step-2 Building Action for final layout synthesis.

areas (e.g., kitchens, bathrooms) and satisfying user-defined position preferences. We propose a Sketch-to-Point projection strategy that treats user sketches as spatial priors to extract precise anchors. To guarantee topological validity, the derived point is strictly constrained to the polygon interior. This method serves as a hard constraint in the optimization objective, enforcing layout alignment without violating boundary conditions.

3.5 Reinforcement Learning

On-policy Optimization via GRPO. To ensure geometry habitability, we adopt GRPO (Shao et al., 2024) for on-policy optimization. GRPO updates the model parameters using samples generated from the current policy, allowing for direct feedback on geometry habitability. We define a composite reward R_{hard} to strictly penalize violations:

$$R_{hard} = \mathbb{I}_{syn} \left(\omega_{crit} R_{crit} + \omega_{gen} R_{gen} \right) \quad (4)$$

where $\mathbb{I}_{syn} \in \{0, 1\}$ is a binary indicator for code executability. We implement a hierarchical weighting scheme: R_{crit} enforces essential topology (e.g., *Room Closure*) with weight ω_{crit}, while R_{gen} penalizes local violations (e.g., *Overlap*) with weight ω_{gen}. By setting $\omega_{crit} > \omega_{gen}$, we ensure that the on-policy updates prioritize global structure, guiding the generation towards physically valid solutions.

4 Experiments

4.1 Metrics and Training Details

Metrics: We evaluate the generated floorplans on geometric habitability. We employ Pass@1 to quantify generation success rates and analyze failures on the test set using four specific metrics: Furniture Non-overlap ($\mathcal{P}_{\text{furn}}$), Topological Connectivity ($\mathcal{P}_{\text{topo}}$), Object Reachability ($\mathcal{P}_{\text{reach}}$), and Vector Validity ($\mathcal{P}_{\text{vec}}$). It is worth noting that we adopt a strict validity protocol, rejecting any samples that suffer from vector errors, inaccessibility (of either rooms or furniture), or furniture overlaps exceeding a threshold of 7%. To ensure statistical reliability, all reported results are averaged over three independent runs.

Training Details. Our model is built upon Qwen3-8B(Yang et al., 2025a). In the CPT stage, the model was trained for 2 epochs with a global batch size of 32 and a learning rate of 1e-5. Subsequently, during SFT, we reduced the batch size to 16 and the learning rate to 9e-6, training for 2 epochs. For the reinforcement learning stages, we train GRPO for 3 epochs with a global batch size of 8 and a learning rate of 1e-6. To maintain training stability, we set the group size (G) to 8, the KL coefficient (β_{KL}) to 0.001, and the sampling temperature to 0.3.

4.2 Feature Analysis

Settings				Metrics				
CPT	Point	Box	CoT	Pass@1(%)↑	$\mathcal{P}_{\text{vec}}$(%)↑	$\mathcal{P}_{\text{topo}}$(%)↑	$\mathcal{P}_{\text{reach}}$(%)↑	$\mathcal{P}_{\text{furn}}$(%)↑
	✓			7.10	97.73	79.26	46.91	11.81
	✓	✓		15.79	96.65	76.98	41.92	29.60
	✓	✓	✓	17.93	97.73	77.01	43.64	33.14
✓	✓			15.49	97.95	**91.29**	**67.03**	20.50
✓	✓	✓		34.06	98.42	88.67	61.07	49.98
✓	✓	✓	✓	**36.95**	**99.02**	89.86	64.86	**52.39**

Table 2: **Ablation study on input features and CPT.** The incorporation of additional features leads to consistent improvements in pass@1.

To validate the efficacy of the proposed features, we conduct an incremental ablation study (As Table 2 shows).

Qualitative analysis. As shown in Figure 6, CoT enhances general spatial planning, while bounding boxes significantly augment the model's spatial awareness.

Effectiveness of CPT. Table 2 demonstrates the effectiveness of the CPT. CPT improves performance across all feature settings. Notably, under the full feature configuration, the inclusion of CPT improves Pass@1 by a substantial margin of **19.02**

(from 17.93% to 36.95%). Furthermore, CPT significantly improves other metrics, with $\mathcal{P}_{\text{furn}}$ increasing from 33.14% to 52.39% and $\mathcal{P}_{\text{vec}}$ increasing to 99.02%, verifying its capability in enforcing geometric habitability.

Impact of Bounding Box. The baseline exhibits poor spatial planning and reasoning, yielding a Pass@1 of only 15.49% and showing severe furniture overlap. The integration of explicit bounding boxes significantly improves performance, raising Pass@1 to 34.06% and reducing overlap by ∼29.48. The improvement indicates that explicit coordinates are critical for geometric understanding and reasoning.

Efficacy of CoT. The integration of CoT achieves the highest structural stability, peaking at a Pass@1 of 36.95%. This result validates the efficacy of the coarse-to-fine generation paradigm: by resolving topological dependencies (e.g., room adjacency, room size) prior to detailed geometry synthesis, the model effectively eliminates logical inconsistencies, ensuring rational spatial planning (as visualized in Figure 6).

4.3 Ablation on Different RL Strategies: DPO vs. GRPO

Method(with Data Aug.)	**Pass@1**(%)↑	$\mathcal{P}_{\text{vec}}$(%) ↑	$\mathcal{P}_{\text{topo}}$(%) ↑	$\mathcal{P}_{\text{reach}}$(%) ↑	$\mathcal{P}_{\text{furn}}$(%) ↑
SFT Baseline	42.33	98.60	92.54	67.21	59.95
SFT + DPO	43.64	98.50	**94.40**	68.76	60.60
SFT + GRPO	**50.46**	**98.95**	94.20	**71.93**	**68.20**

Table 3: Ablation analysis of RL strategies. GRPO achieved substantial improvements over the baseline.

To validate the effectiveness of reinforcement learning, we conducted experiments using both GRPO and DPO. As shown in Table 3, both strategies yield performance gains compared to the SFT baseline. However, GRPO achieves a significantly higher improvement than DPO. Notably, prior to the RL stage, we implemented rotational data augmentation on the baseline to ensure model stability.

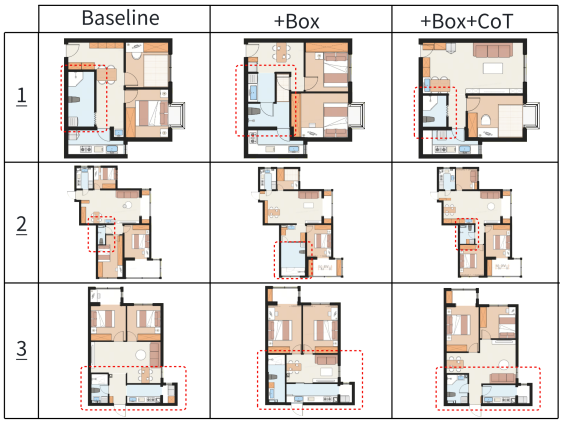

Figure 6: Visualization of feature attribution. Cases 1–3 show that Box and CoT significantly enhance spatial planning performance.

GRPO. As Table 3 shows, compared to the baseline, SFT + GRPO improves Pass@1 by 8.13% (42.33% → 50.46%) and improves $\mathcal{P}_{\text{furn}}$ by 8.25% (59.95% → 68.20%). This demonstrates that online optimization with rule-based rewards is helpful for aligning the model's reasoning with geometric habitability.

DPO. For fairness comparison, we adopt DPO (Rafailov et al., 2023) for **off-policy refinement**, utilizing a fixed dataset to guide the physical generation. We construct preference pairs (y_w, y_l), where the positive sample y_w is drawn from a **mixed dataset of expert designs and selected high-quality model outputs**, while y_l is a standard generated sample. Applying DPO (SFT + DPO) produced only slight improvements over the baseline, with Pass@1 increasing slightly from 42.33% to 43.64%. As shown in Table 4, DPO achieves a Pass@1 score of 43.64%, representing only a minor improvement over the *SFT Baseline* (42.33%). In contrast, GRPO achieves a significantly higher Pass@1 of 50.46%, outperforming the direct DPO approach by a large margin.

We attribute the gap to the limits of off-policy learning. DPO relies on a fixed dataset. Notably, it achieves a Topological Consistency ($\mathcal{P}_{topo}$) of 94.40%, surpassing the 94.20% achieved by GRPO. We argue that GRPO uses *explicit* reward feedback, which directly guides the model to satisfy strict constraints. In contrast, DPO relies on *implicit* guidance through preference pairs. This implicit signal is harder for the model to capture compared to direct feedback, resulting in lower Pass@1.

4.4 Comparison with domain-specific models

To the best of our knowledge, our method presents the first unified framework for floorplan design, bridging the gap between floorplan generation and scene generation. We compare the evaluation results of our model with domain-specific models of floorplan generation and scene generation.

4.4.1 Floorplan Generation

We evaluate our zero-shot floorplan generation capabilities against ChatHouseDiffusion (Qin et al., 2024) on the Tell2Design (Leng et al., 2023) dataset. Unlike the baseline, which relies on ex-

Method	Floorplan Generation		Scene Generation	
	$\mathcal{P}_{\text{vec}}(\%)\uparrow$	$\mathcal{P}_{\text{topo}}(\%)\uparrow$	$\mathcal{P}_{\text{reach}}(\%)\uparrow$	$\mathcal{P}_{\text{furn}}(\%)\uparrow$
ChatHouseDiffusion	90.90	82.50	N/A	N/A
LayoutGPT(GPT-5.2)	N/A	N/A	38.50	55.90
LayoutGPT(Gemini-3-Flash)	N/A	N/A	37.42	53.70
Ours	**97.56**	**92.26**	**72.13**	**68.08**

Table 4: Main results on floorplan and scene generation. Our method achieves competitive performance across both tasks. Unlike domain-specific baselines, our unified framework demonstrates robust capabilities in both floorplan generation and scene generation.

plicit room constraints, our method utilizes LLMs to autonomously derive spatial arrangements from boundary contours. We employ $\mathcal{P}_{\text{topo}}$ to evaluate the results. To ensure fairness, for ChatHouseDiffusion, we evaluate $\mathcal{P}_{\text{topo}}$ the generated image with a raster-based segmentation method. For downstream usage, like editing 2D or 3D in CAD (Zhang et al., 2024b)/BIM, we also convert the image to vector when calculating $\mathcal{P}_{\text{inv}}$, in order to measure the precision loss caused by the R2V task. Experimental results demonstrate that our method significantly outperforms the baseline (See Table 4). As illustrated in Figure 7, ChatHouseDiffusion frequently generates inaccessible rooms or ambiguous doorways (spanning only a few pixels). Furthermore, the R2V process exacerbates these issues and introduces additional errors, severely limiting their applicability in downstream tasks.

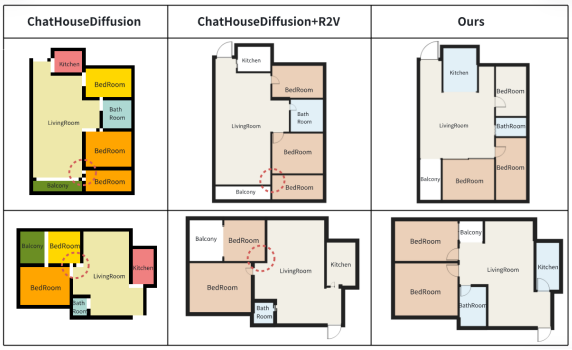

Figure 7: Qualitative comparison of floorplan generation. The left column displays raster outputs from ChatHouseDiffusion, highlighting topological defects such as inaccessible rooms and ambiguous door boundaries (red circles). The middle column shows that the subsequent Raster-to-Vector (R2V) introduces additional errors. In contrast, our method (right) generates floorplans with correct topology.

4.4.2 Scene Generation

Table 4 also reports the results of LayoutGPT (Feng et al., 2023) on our UniFPDesign-2K benchmark, powered by GPT-5.2 (OpenAI, 2025) and Gemini-3-Flash (Google, 2025). Following standard practice (Feng et al., 2023), we limit the evaluation to generating furniture layout in one bedroom and one living room per sample. In the scene generation task, our method significantly outperforms LayoutGPT variants powered by state-of-the-art LLMs. The results show a substantial improvement in $\mathcal{P}_{\text{reach}}$ to 72.13%—approximately twice that of the strongest baseline—while maintaining a $\mathcal{P}_{\text{furn}}$ of 68.08%. These results demonstrate our model's robust capability in ensuring furniture reachability and optimizing spatial arrangements to minimize invalid overlaps (See Figure 8).

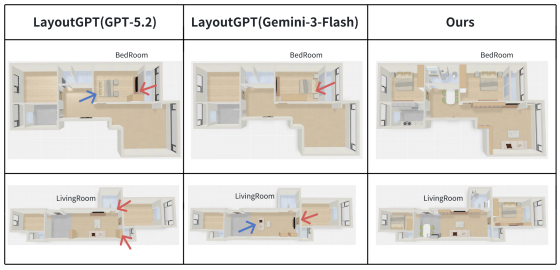

Figure 8: Visual comparison with LayoutGPT. Red arrows indicate furniture's reachable problems, and blue arrows show furniture misaligned with walls. While LayoutGPT often produces these errors, our method successfully generates valid layouts for rooms.

5 Conclusion

In this work, we present UniFPDesign, a unified framework for floorplan design. To address data scarcity, we introduce a **Persona-Guided Specification Inversion** pipeline that synthesizes aligned triplets of user demand, constraints, and layouts. Our scalable training pipeline begins with CPT and SFT enhanced by feature augmentation. Subsequently, we employ GRPO to solve the physical conflict. Experiments on floorplan and scene generation tasks demonstrate the effectiveness of our method in generating valid and high-quality designs. Empirical results on our proposed benchmark **UniFPDesign-2K** for strictly constrained floorplan design, demonstrate that our approach establishes a strong baseline.

Limitations and Future Work. UniFPDesign currently relies exclusively on symbolic "Geometry-as-Code" representations, lacking the capability to directly process raw visual inputs such as reference images or pixel-level sketches. This absence restricts the model from leveraging intuitive visual

cues that are intrinsic to architectural design. Future work will address this by integrating specialized, geometry-aware visual backbones to enable true multi-modal spatial reasoning. Furthermore, our reliance on predefined sociological personas, while effective for resolving ambiguity, may inherently oversimplify complex human behaviors into fixed archetypes. Future iterations aim to mitigate this risk by exploring data-driven persona discovery from large-scale user logs, thereby capturing more granular and long-tail lifestyle distributions.

Furthermore, while our persona-guided demand synthesis effectively resolves ambiguity, it presents intrinsic limitations. First, the reliance on predefined sociological personas may oversimplify complex, fluid human behaviors into fixed archetypes. For instance, in one failure case involving an elderly couple (aging-friendly design), the model misinterpreted a "bunk bed" in the living room—likely intended for a caregiver or visiting grandchildren—as definitive evidence of a "Family with Two Children" persona. This illustrates the difficulty of mapping non-standard furniture usage to correct user identities without finer-grained, data-driven archetypes. Second, employing general-purpose LLMs for reasoning introduces a risk of hallucination, where synthesized demands might occasionally diverge from strict geometric logic despite our constraint verification mechanisms. Third, our evaluation relies primarily on intrinsic similarity metrics as proxies, which cannot fully capture the subjective nuances of dwelling satisfaction in real-world scenarios. Future iterations aim to mitigate these risks by exploring data-driven persona discovery from large-scale user logs, thereby capturing more granular and long-tail lifestyle distributions.

Safety and Responsibility. UniFPDesign is intended as a decision-support system rather than an automated design authority. All generated floorplans should be reviewed by licensed architects or structural engineers before any real-world deployment. In particular, recommendations involving load-bearing structures, plumbing shafts, or fire-safety designs require expert validation. We view this work as a step toward assistive design tools, not a replacement for professional judgment.

Ethical Considerations and Data Consent. The data used in the evaluation part primarily stems from the RPLAN dataset, which is a publicly available benchmark for academic research. We strictly adhere to its licensing terms. Regarding the "Persona-Guided Specification Inversion", all user personas and design requirements were synthetically generated by LLMs based on predefined architectural templates. No real personal data was scraped or collected from human subjects.

References

Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. In *arXiv preprint arXiv:2406.20094*.

Jiacheng Chen, Ruizhi Deng, and Yasutaka Furukawa. 2023. Polydiffuse: Polygonal shape reconstruction via guided set diffusion models. *Advances in Neural Information Processing Systems*, 36:1863–1888.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. 2023. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36:18225–18250.

Kazuki Fujii, Yukito Tajima, Sakae Mizuki, Hinari Shimada, Taihei Shiotani, Koshiro Saito, Masanari Ohi, Masaki Kawamura, Taishi Nakamura, Takumi Okamoto, Shigeki Ishida, Kakeru Hattori, Youmi Ma, Hiroya Takamura, Rio Yokota, and Naoaki Okazaki. 2025. Rewriting pre-training data boosts llm performance in math and code. *Preprint*, arXiv:2505.02881.

Google. 2025. Gemini 3 Flash: Best for frontier intelligence at speed. `https://deepmind.google/models/gemini/flash/`. Accessed: 2025-12-25.

Samuel D. Gosling, Sei Jin Ko, Thomas Mannarelli, and Margaret E. Morris. 2002. A room with a cue: Personality judgments based on offices and bedrooms. *Journal of Personality and Social Psychology*, 82(3):379–398.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009.

Michael Igorevich Ivanitskiy, Alex F Spies, Tilman Räuker, Guillaume Corlouer, Chris Mathwin, Lucia Quirke, Can Rager, Rusheb Shah, Dan Valentine, Cecilia Diniz Behn, and 1 others. 2023. Structured

world representations in maze-solving transformers. *arXiv preprint arXiv:2312.02566*.

Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4480–4498, Online. Association for Computational Linguistics.

Kengo Kuma. 1986. *Ten Houses Theory (Juttakuron)*. Parco Publishing, Tokyo, Japan.

Sicong Leng, Yang Zhou, Mohammed Haroon Dupty, Wee Sun Lee, Sam Joyce, and Wei Lu. 2023. Tell2design: A dataset for language-guided floor plan generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14680–14697.

Jinmin Li, YILU Luo, SHUAI Lu, JINGYUN Zhang, J Wang, RIZEN Guo, and SHAOMING Wang. 2024a. Chatdesign: Bootstrapping generative floor plan design with pre-trained large language models. In *Proceedings of the 29th International Conference of the Association for Computer Aided Architectural Design Research in Asia (CAADRIA), Hongkong*, pages 23–25.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2024b. Self-alignment with instruction backtranslation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

You-Dong Liang and Brian A Barsky. 1984. A new concept and method for line clipping. *ACM Transactions on Graphics (TOG)*, 3(1):1–22.

Zhi Hao Luo, Luis Lara, Ge Ya Luo, Florian Golemo, Christopher Beckham, and Christopher Pal. 2024. Dstruct2design: Data and benchmarks for data structure driven generative floor plan design. *arXiv preprint arXiv:2407.15723*.

Nelson Nauata, Kai-Hung Chang, Chin-Yi Cheng, Greg Mori, and Yasutaka Furukawa. 2020. House-gan: Relational generative adversarial networks for graph-constrained house layout generation. In *European Conference on Computer Vision*, pages 162–177. Springer.

Nelson Nauata, Sepidehsadat Hosseini, Kai-Hung Chang, Hang Chu, Chin-Yi Cheng, and Yasutaka Furukawa. 2021. House-gan++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13632–13641.

OpenAI. 2025. Introducing GPT-5.2. https://openai.com/index/introducing-gpt-5-2/. Accessed: 2025-12-25.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544.

Zhenting Qi, Fan Nie, Alexandre Alahi, James Zou, Himabindu Lakkaraju, Yilun Du, Eric Xing, Sham Kakade, and Hanlin Zhang. 2025. Evolm: In search of lost language model training dynamics. *arXiv preprint arXiv:2506.16029*.

Sizhong Qin, Chengyu He, Qiaoyun Chen, Sen Yang, Wenjie Liao, Yi Gu, and Xinzheng Lu. 2024. Chathousediffusion: Prompt-guided generation and editing of floor plans. *Preprint*, arXiv:2410.11908.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, and 1 others. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

William Rudman, Michal Golovanevsky, Amir Bar, Vedant Palit, Yann LeCun, Carsten Eickhoff, and Ritambhara Singh. 2025. Forgotten polygons: Multimodal large language models are shape-blind. *arXiv preprint arXiv:2502.15969*.

Ayan Sengupta, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. Persona-aware generative model for code-mixed language. In *arXiv preprint arXiv:2309.02915*.

Mohammad Amin Shabani, Sepidehsadat Hosseini, and Yasutaka Furukawa. 2023. Housediffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5466–5475.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Scott J. South, Ying Huang, Amy Spring, and Kyle Crowder. 2016. Neighborhood attainment over the adult life course. *American Sociological Review*, 81(6):1276–1304.

Peiyang Su, Weisheng Lu, Junjie Chen, and Shibo Hong. 2024. Floor plan graph learning for generative design of residential buildings: a discrete denoising diffusion model. *Building Research & Information*, 52(6):627–643.

Fan-Yun Sun, Weiyu Liu, Siyi Gu, Dylan Lim, Goutam Bhat, Federico Tombari, Manling Li, Nick Haber, and Jiajun Wu. 2025. Layoutvlm: Differentiable optimization of 3d layout via vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29469–29478.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.

Xilin Wang, Jia Zheng, Yuanchao Hu, Hao Zhu, Qian Yu, and Zihan Zhou. 2025. From 2d cad drawings to 3d parametric models: A vision-language approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7961–7969.

Yinyi Wei and Xiao Li. 2024. Graph-augmented text-based floorplan generation. In *2024 International Conference on Automation in Manufacturing, Transportation and Logistics (ICaMaL)*, pages 1–9. IEEE.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yixuan Yang, Zhen Luo, Tongsheng Ding, Junru Lu, Mingqi Gao, Jinyu Yang, Victor Sanchez, and Feng Zheng. 2025b. Optiscene: Llm-driven indoor scene layout generation via scaled human-aligned data synthesis and multi-stage preference optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, and 1 others. 2024. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16227–16237.

Jun Yin, Pengyu Zeng, Jing Zhong, Peilin Li, Miao Zhang, Ran Luo, and Shuai Lu. 2025a. Floorplan-deepseek (fpds): A multimodal approach to floorplan generation using vector-based next room prediction. *arXiv preprint arXiv:2506.21562*.

Jun Yin, Jing Zhong, Pengyu Zeng, Peilin Li, Miao Zhang, Ran Luo, and Shuai Lu. 2025b. Floorplan-mae: A self-supervised framework for complete floorplan generation from partial inputs. *arXiv preprint arXiv:2506.08363*.

Pengyu Zeng, Wen Gao, Jun Yin, Pengjian Xu, and Shuai Lu. 2024. Residential floor plans: Multi-conditional automatic generation using diffusion models. *Automation in Construction*, 162:105374.

Hang Zhang, Anton Savov, and Benjamin Dillenburger. 2024a. Maskplan: Masked generative layout planning from partial input. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8964–8973.

Zhanwei Zhang, Shizhao Sun, Wenxiao Wang, Deng Cai, and Jiang Bian. 2024b. Flexcad: Unified and versatile controllable cad generation with fine-tuned large language models. *arXiv preprint arXiv:2411.05823*.

A Persona hub Details

Each persona specifies a compact set of demographic variables (e.g., occupants, children, elders, potential caregivers) and induces different demand priorities when combined with extracted layout facts C_{fact}.

Representative personas include:

(i) *couple without children/elders* (DINK);

(ii) *couple planning for children* (future nursery reserved);

(iii) *couple + 1 child*;

(iv) *couple + 2 children*;

(v) *solo female*;

(vi) *solo male*;

(vii) *two-generation co-residence* (adult child + parent);

(viii) *couple co-residing with parents* (three-generation pressure);

(ix) *elders staying temporarily for childcare*, and

(x) *with caregiver* (nanny/aid).

These personas are used to condition wishlist injection and demand prioritization during synthesis.

B Structure-consistent wall masking details

We construct self-supervised training pairs by corrupting a complete floorplan while enforcing that the corrupted input remains *structure-consistent* with our representation (e.g., valid wall endpoints, consistent wall–room incidence, and closed room loops when present). Following the main text, we represent a floorplan as $\mathcal{X} = \mathcal{S} \cup \mathcal{F}$, where $\mathcal{S}$ denotes immutable structural elements and $\mathcal{F}$ denotes designable functional units.

For implementation, we further decompose a plan into geometric entities

$$x = \{W^{\text{out}}, W^{\text{in}}, A, O, S, F\}, \tag{5}$$

where W^{out} and W^{in} denote outer/inner walls, A rooms (each room stores a wall-loop outline), O openings (doors/windows attached to walls), S structural fixtures (e.g., shafts, columns), and F

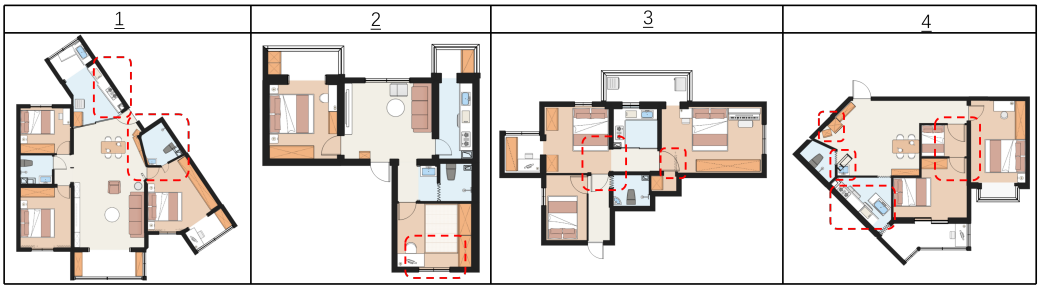

Figure 9: Visualization of failure cases

designable furniture. We sample a corruption configuration $\omega \sim \pi(\omega)$ and generate

$$\tilde{x} = \mathcal{C}(x; \omega), \tag{6}$$

training the model to reconstruct the original plan from $\tilde{x}$ in a denoising fashion.(Vincent et al., 2008; Devlin et al., 2019; He et al., 2022; Pathak et al., 2016) We implement $\mathcal{C}$ with two masking modes, both followed by a deterministic repair step.

Deletable vs. protected walls. We only delete a subset of inner walls to avoid violating practical constraints. Specifically, we define

$$\mathcal{W}_{\text{del}}(x) = W^{\text{in}} \setminus \mathcal{W}_{\text{protected}}(x), \tag{7}$$

where $\mathcal{W}_{\text{protected}}(x)$ includes walls associated with protected functional/structural regions (e.g., columns, shafts, flues, and stairs), as well as walls filtered by conservative geometric heuristics (e.g., unusually large thickness/length or wide window openings).

Mode I: Random region wall masking. This mode removes inner walls inside a randomly sampled missing region while optionally clipping walls that intersect the region boundary.

- **Region sampling.** Let P be the polygon induced by the outer outline W^{out}. We sample a ratio $r \sim \mathcal{U}(0, 1)$ and then sample an axis-aligned rectangle R such that

$$\text{area}(P \cap R) \approx r \cdot \text{area}(P). \tag{8}$$

- **Wall deletion and clipping.** For each candidate wall $w \in \mathcal{W}_{\text{del}}(x)$, we compute the fraction of its length inside R, denoted $\rho(w, R)$. We categorize walls as:
  - **fully masked** if $\rho(w, R) \geq \tau_{\text{in}}$ (e.g., 0.9), deleting the whole wall;
  - **boundary-intersecting** if $\tau_{\text{int}} \leq \rho(w, R) < \tau_{\text{in}}$ (e.g., $\tau_{\text{int}} = 0.1$), clipping the wall segment at the rectangle boundary to keep only the outside portion.

  Clipping is computed by a standard line–rectangle intersection (e.g., Liang–Barsky(Liang and Barsky, 1984)), producing a new boundary endpoint p^* which replaces the endpoint that lies inside R. This partial deletion yields realistic missing segments and encourages the model to learn wall continuation from surrounding structure.

Mode II: Rule-guided non-loadbearing wall deletion. To better match practical demolition patterns, we optionally apply a rule-guided deletion based on $\mathcal{W}_{\text{protected}}(x)$ above. We then delete from $\mathcal{W}_{\text{del}}(x)$ by sampling an integer deletion level $k \sim \mathcal{U}\{0, 1, \ldots, 10\}$ and removing k randomly selected wall-chains (collinear and endpoint-connected segments within the same room) to avoid implausible fragmented removals.

Structure-consistency repair. After deletion/clipping, we apply a deterministic repair $\mathcal{R}$ to maintain consistency of the floorplan representation: (i) update wall endpoints and incident relations after deletion/clipping to avoid dangling references; (ii) merge collinear walls across removed junctions to reduce fragmentation; (iii) remove rooms whose wall-loop outlines are broken (i.e., cannot form a closed cycle); (iv) drop openings attached to deleted walls; (v) canonicalize wall identifiers to stabilize serialization. Overall,

$$\tilde{x} = \mathcal{R}\big(\mathcal{D}(x; \omega)\big), \tag{9}$$

where $\mathcal{D}$ instantiates either Mode I (region deletion with optional clipping) or Mode II (rule-guided wall-chain deletion), and $\mathcal{C} = \mathcal{R} \circ \mathcal{D}$.

Finally, we keep only structural fixtures in the input while predicting designable furniture in the output.

C Data Distribution

Data Source and Usage Constraints. The CPT and SFT data used in this work are derived from real-world residential floorplans. The dataset was collected under strict non-disclosure agreements and anonymized to remove all personally identifiable information and location-related metadata. Due to legal and copyright constraints, the training data cannot be released. Nevertheless, we provide a benchmark for evaluation.

The cornerstone of our dataset is the serialization of floorplan vectors into executable Python code. Unlike static formats (e.g., JSON), this approach leverages the pre-trained model's proficiency in modeling variable dependencies. By representing spatial layouts as structured code, the model is forced to internalize geometric constraints, such as room adjacency and wall closure. During the CPT stage, we use approximately 40B tokens of training data. To ensure the model's understanding of both general textual knowledge and user demand, the corpus is partitioned into several subsets. Table 5 summarizes the components of the mixed training data and their respective roles in the training process.

To maintain the model's foundational reasoning while specializing in architectural geometry, we adopt a specific serialization format and data mixture strategy.

Data Mixing Strategy To mitigate catastrophic forgetting (Roziere et al., 2023), we mix some general-purpose data. This mixture serves three distinct cognitive functions:

D Metrics

Table 6 presents a comprehensive overview of our evaluation framework. These metrics impose hard constraints to ensure the Geometry Habitability of the floorplans. Key metrics include Generation Success (Pass@k) and Vector Validity ($\mathcal{P}_{\text{vec}}$), which verify the syntactic correctness of the vector representations. Furthermore, we assess spatial feasibility through Furniture Overlap ($\mathcal{P}_{\text{furn}}$), which penalizes physical collisions, and topological metrics such as Connectivity ($\mathcal{P}_{\text{topo}}$) and Reachability ($\mathcal{P}_{\text{reach}}$), which guarantee that all functional areas and furniture items are reachable via valid circulation.

E Failure Case Analysis

We visualize typical failure cases in Figure 9. The primary errors occur when dealing with **complex boundaries**. As seen in Case 1 and Case 4, when the input outline is highly irregular, the model struggles to partition the space effectively, leading to disconnected rooms or invalid overlaps. Another common failure arises in **extremely tight spaces**. In Case 2 and Case 3, the limited area forces the model to generate layouts where furniture obstructs doors or hallways become too narrow. These examples indicate that while our method performs well on standard layouts, it still faces challenges when handling extreme geometric irregularities or severe spatial constraints.

F AI Assistant Usage Disclosure

Research and Data Generation: We utilized Large Language Models (specifically Doubao, GPT, and Gemini) to facilitate the "Persona-Guided Specification Inversion" process. These models were used to synthesize diverse user personas and corresponding architectural requirements based on seed data from the RPLAN dataset. The AI-generated prompts were manually reviewed by the authors to ensure logical consistency.

Coding: AI coding assistants (e.g., GitHub Copilot) were used to assist in writing the infrastructure code for the evaluation pipeline. All AI-suggested code was thoroughly tested and verified by the research team.

Writing and Polishing: We used ChatGPT/Gemini for grammar correction, stylistic polishing, and rephrasing to improve the clarity and flow of the manuscript. The core arguments, technical descriptions, and data interpretations were entirely conceived and written by the human authors.

Component	Ratio	Rationale
Vector Code	90%	Core *Geometry-as-Code* generation tasks for spatial generation and interpretation.
General Python (Fujii et al., 2025)	5%	Prevents syntax degradation and maintains code-structure fluency.
Mathematical Data (Fujii et al., 2025)	4%	Enhances logical reasoning and numerical precision for coordinate arithmetic.
Domain NL (from the internet)	1%	Ensures robust alignment between user instructions and geometric execution.

Table 5: Composition of the Training Corpus.

Dimension	Metric Name	Symbol	Description & Key Indicators
Physical Plausibility	Generation Success	Pass@1	Evaluate the success rate of generating valid floorplans.
	Furniture Non-overlap	$\mathcal{P}_{\text{furn}}$	Quantifies the ratio of physically non-overlapping furniture areas (Physical Failure).
	Connectivity	$\mathcal{P}_{\text{topo}}$	Evaluates the graph topological connectivity of the generated room layout.
	Object Reachability	$\mathcal{P}_{\text{reach}}$	Measures the reachability of furniture within the floorplan.
	Vector Validity	$\mathcal{P}_{\text{vec}}$	Checks for geometric validity (e.g., syntactic correctness).

Table 6: Overview of the Evaluation Framework: Geometry Habitability.