WALK AND IT APPEARS: INTERFACE-CONSTRAINED CONSTRUCTION OF TRAVERSABLE 3D WORLDS

Anonymous authors
Paper under double-blind review



Figure 1: We present a system that grows an explorable 3D world from a single image or text prompt. A world program defines global structure and appearance; local constructors generate terrain, buildings, and furnished interiors under shared spatial constraints. Generation proceeds ahead of the user as they move, while persistent storage preserves previously built regions for revisits. The same pipeline supports urban, natural, and indoor environments.

ABSTRACT

Turning a prompt or visual reference into a world one can traverse requires new geometry to connect to previous construction and arrive before the user reaches it. We present *Walk and It Appears*, a system that authors a persistent world program, constructs local geometry through canonical interfaces, and streams completed records under stable identities. These interfaces carry terrain elevations and road crossings between chunks, parcel frames into buildings, and entrances into interiors. The reported evaluation covers 233 runs across 14 task–seed configurations and six language-model backbones. Equal-backbone mean scores favor our system over direct expandable-code generation in physical validity (0.80 versus 0.31), route success (0.80 versus 0.37), and online chunk availability (0.95 versus 0.62) under the reported runtime configurations. A $1.5\,\mathrm{km}$ outward-and-return diagnostic probes expansion and revisits; image-conditioned and indoor cases are curated demonstrations. The evaluation lacks per-task score breakdowns and component ablations, so cross-task variability and individual module effects remain unresolved.

1 INTRODUCTION

An explorable world cannot end at the edge of its first image. Applications in interactive simulation, games, and embodied-agent evaluation require a scene that remains traversable as a user moves from distant terrain to streets, buildings, and rooms. Constructing such a scene from a single image or short prompt is difficult for three coupled reasons: unseen space must be specified, newly generated geometry must agree with previously visited space, and usable content must arrive before the user reaches it. The challenge is therefore both generative and operational: visual plausibility at one

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.

camera pose is insufficient if roads break at chunk boundaries, entrances do not meet floors, or expansion stalls.

Existing methods often synthesize a bounded 3D scene and extend it with image, latent-block, or geometric inpainting (Höllein et al., 2023; Wu et al., 2024; Li et al., 2025b). Interactive video world models instead generate the next observation, sometimes retrieving earlier frames to improve consistency (Sun et al., 2026; Wang et al., 2026b). Procedural systems provide reusable geometry rules, but a high-level reference must still be translated into compatible landscape, settlement, building, and interior decisions (Raistrick et al., 2023; Wang et al., 2025). These approaches expose a common difficulty: local content can look coherent while its spatial interfaces, physical access, and long-term identity remain underconstrained. The comparison in this paper addresses the complete-system alternative of asking the same language model to write either a finite scene or an expandable tile constructor; it does not isolate any single component of prior published systems.

Our central design is a *persistent construction program* derived from the reference. It assigns terrain, settlement, building, and interior decisions at the scales where they apply. Canonical interfaces fix the geometry that neighboring regions must share, while stable identities let exploration construct nearby content and retrieve committed content on revisit. We call the resulting system *Walk and It Appears*. Separating the authored world definition from resident geometry lets local construction extend the world without rewriting earlier space.

Three components implement this idea (Figure 2). *Reference-conditioned program authoring* extracts and checks spatial and visual constraints, then produces reusable terrain, layout, building, and interior programs. *Interface-constrained local construction* evaluates those programs at stable spatial identities: shared edge profiles connect terrain and infrastructure, parcels constrain buildings, and shells and openings constrain interiors. The *persistent-streaming component* schedules construction ahead of the user, keeps only nearby geometry resident, and reloads committed records when an area is revisited. Each component serves the same goal: new content should be conditioned by the original reference and by geometry already committed to the world.

The quantitative evaluation spans 233 runs across 14 task–seed configurations and six language-model backbones, comparing the system with finite and expandable direct-code generation under matched model-call ceilings. Against the expandable baseline, the reported equal-backbone means are 0.80 versus 0.31 for physical validity and 0.80 versus 0.37 for route success; online chunk availability is 0.95 versus 0.62 under the specified runtime configurations. A $1.5\,\mathrm{km}$ outward-and-return stress route probes expansion and revisits. Curated urban, natural, and indoor examples illustrate the range of produced content. The source does not provide per-task scores or module ablations, limiting generalization and causal attribution.

Our contributions are threefold:

- A reference-conditioned, persistent world definition that carries terrain, settlement, architecture, and interior decisions across exploration requests.
- A local construction interface that connects adjacent chunks, road and river crossings, building pads, entrances, and interior openings under stable identities.
- An integrated streaming system and paired evaluation of requirement fidelity, physical usability, visual preference, long-range behavior, and online availability, with explicit coverage and protocol limitations.

2 Related Work

2.1 Expandable 3D scene generation

Image-conditioned scene methods convert generated views into geometry, but the geometry must remain plausible when the camera leaves the initial view. Text2Room (Höllein et al., 2023) fuses generated images and aligned depth into textured room meshes; WonderWorld (Yu et al., 2025) uses guided depth diffusion to connect interactively generated scenes. BlockFusion (Wu et al., 2024) extrapolates overlapping latent tri-planes, WorldGrow (Li et al., 2025b) grows hierarchical 3D blocks, and SynCity (Engstler et al., 2025) synthesizes and fuses contextualized tiles. NuiWorld (Lee et al., 2026) represents variable collections of scene chunks, whereas MetaEarth3D (Cao et al.,

2026) addresses scaling from terrain to street blocks. Video-to-geometry pipelines offer another path: Lyra 2.0 (Shen et al., 2026) aligns past observations through geometry before reconstructing generated trajectories, and Streaming4D (Liu et al., 2026) couples block-wise video generation to incremental reconstruction. Urban generators narrow the domain to exploit stronger structure: InfiniCity (Lin et al., 2023) uses satellite maps, CityDreamer (Xie et al., 2024) separates building instances and backgrounds, GaussianCity (Xie et al., 2025) decodes compact point representations, and CityGen (Deng et al., 2025) expands semantic layouts. MajutsuCity (Huang et al., 2025) adds language-guided layouts, assets, and materials. These works establish that growth beyond one fixed volume is feasible, but their representations and evaluation targets vary substantially. Our method also constructs space incrementally. It differs in making shared terrain and infrastructure edges, building pads, exterior–interior openings, and committed scene identities explicit interfaces of a persistent 3D construction program; our paired tests compare against direct-code baselines rather than against the published systems listed here.

2.2 Interactive video worlds and memory

Video world models generate observations conditioned on camera or action, providing a direct route to interactive visual experience. Genie 3 (Ball et al., 2025) and Hunyuan-GameCraft (Li et al., 2025a) demonstrate controllable environment generation, while Yume-1.5 (Mao et al., 2025) and LingBot-World 2.0 (Gao et al., 2026) support text-driven events during interaction. A central challenge is returning to earlier places without losing their appearance or spatial identity. WorldPlay (Sun et al., 2026) reconstitutes historical context and uses memory-aware distillation; Matrix-Game 3.0 (Wang et al., 2026b) retrieves camera-relevant memories; and Infinite-World (Wu et al., 2026) compresses history through a pose-free hierarchy. SANA-WM (Zhu et al., 2026) studies efficient long-context modeling and camera control. External memory provides a complementary route: WorldMem (Xiao et al., 2025) stores frames and state, MultiGen (Po et al., 2026) maintains editable state shared across players, and Alaya-EVOKE (Yin et al., 2026) retrieves geometric context from a camera-indexed bank. The common thread is that a model needs access to what was generated earlier; however, a retrieved image or latent state does not by itself specify the physical interfaces of a road, doorway, or walkable floor. We address this different representation problem by retaining constructed geometry and its spatial provenance, rendering revisits from the stored scene, and using the same entity identity when new neighboring content is added. The resulting comparison concerns geometric usability and chunk availability as well as appearance; it does not claim a head-to-head advantage over these video models.

2.3 Procedural and program-guided construction

Programs can express reusable scene structure more compactly than a list of meshes. Infinigen (Raistrick et al., 2023) provides procedural natural environments, and Infinigen Indoors (Raistrick et al., 2024) adds indoor generators and constraint-based arrangement. Language-guided methods translate intent into executable structure: Holodeck (Yang et al., 2024) derives layouts and object relations from text, and SceneCraft (Hu et al., 2024) writes and refines Blender programs. WorldGen (Wang et al., 2025) combines layout reasoning, procedural construction, and generative assets for traversable worlds. WorldClaw (Guo et al., 2026) establishes global terrain before progressively constructing terrain-conditioned regions, while Code as Worlds (Wang et al., 2026a) and Recursive Code World Models (Li et al., 2026) use executable representations for physical reasoning and hierarchical scene construction. Magpie (Zhan et al., 2026) separates engine execution from generative rendering, and Code World Model (Chen et al., 2026) evolves persistent state with a coding agent and compiles it into proxy videos. Our system shares the premise that programs and persistent state can carry decisions beyond one generation call. Its specific construction contract binds reference-derived programs to coordinate-stable identities and canonical interfaces, then materializes terrain, settlement, buildings, and interiors only when requested. This contract matters because a district rule can constrain a parcel before a building is assembled, and a building shell can constrain an interior before furniture is placed. The streaming runtime stores the result so that subsequent requests extend an existing world. We evaluate this integrated system against direct finite and expandable code generation; the contribution of each individual module still requires a controlled ablation.

3 METHOD

Given a reference u (text or one or more images) and user poses $p_0, p_1, \ldots$, the task is to provide renderable, traversable geometry around each p_t. Newly exposed terrain, roads, buildings, and rooms must meet what has already been built, and usable content must arrive before the user reaches it. Exploration may expand the world indefinitely in coordinate space, while the renderer keeps a bounded working set.

Our pipeline has three components (Figure 2). *Reference-conditioned program authoring* turns u into a checked world definition. *Interface-constrained local construction* evaluates that definition at stable spatial identities, carrying terrain edges, parcels, building shells, and openings into neighboring content. The *persistent-streaming component* requests geometry near p_t, commits completed records, and reloads them under the same identities. Deterministic seeds, shared assets, and level-of-detail rendering operate within this pipeline.

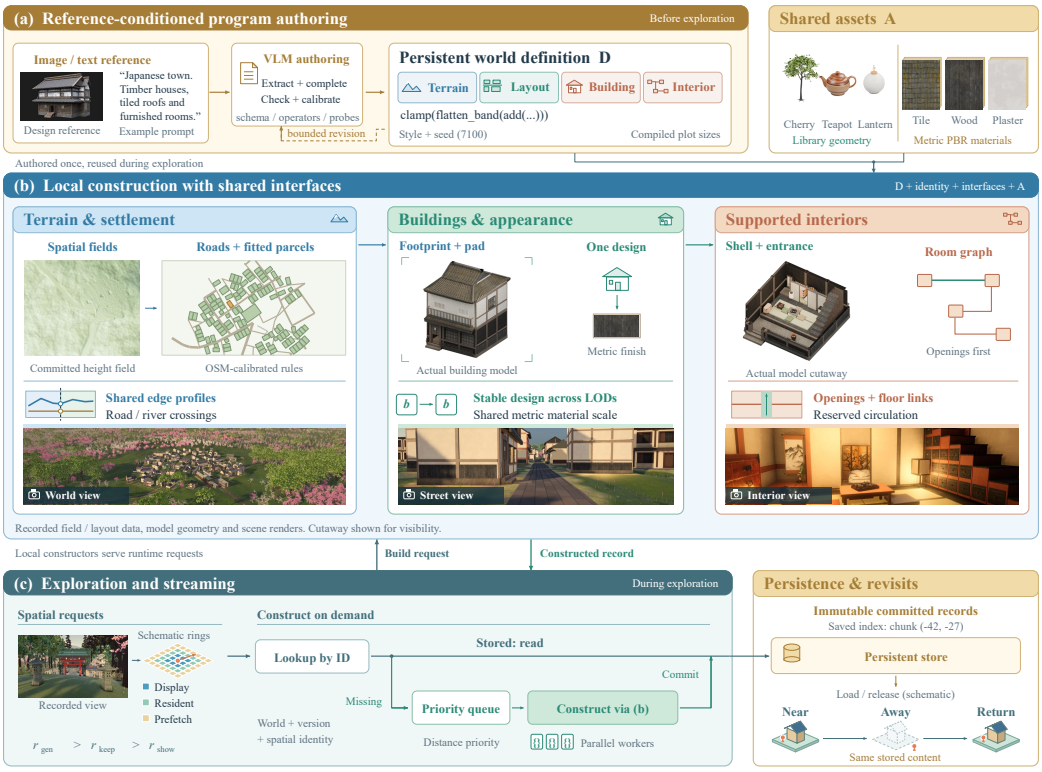

Figure 2: System overview. (a) Reference-conditioned authoring produces a persistent world definition. (b) Local constructors use shared interfaces to connect terrain and settlements, buildings, and interiors. (c) Exploration schedules construction, while committed records and stable identities support revisits.

3.1 REFERENCE-CONDITIONED PROGRAM AUTHORING

Let s be a world seed. The reusable definition $\mathcal{D}$ and the materialized state $\mathcal{S}_t$ after exploration step t are distinct:

$$\mathcal{D} = (\Sigma, G, \mathcal{L}, \mathcal{K}, s), \qquad \mathcal{S}_t = \{(\iota, C_\iota) : \iota \in I_t\}. \tag{1}$$

Here Σ holds semantic and appearance attributes; G defines terrain and infrastructure fields; $\mathcal{L}$ holds settlement plans; and $\mathcal{K}$ holds building and interior programs. The set I_t indexes committed spatial or entity identities. Record C_ι stores geometry or its construction specification, materials, placements, and traversal data for identity ι. Outdoor chunks have identities $i \in \mathbb{Z}^2$ and domains $\Omega_i = L(i + [0,1]^2)$, where L is chunk width. Buildings retain settlement and parcel identities; indoor regions additionally carry floor identities. Only a bounded subset of I_t resides in the renderer.

A multimodal language model extracts landforms, architecture, vegetation, materials, atmosphere, and spatial constraints from u, preserving which constraints are explicit in text and which are inferred from visual evidence. It completes unspecified attributes, then authors typed terrain operators, district and parcel rules, architectural templates, and room programs. Schema checks, operator signatures, cross-field constraints, and terrain and settlement probes return field-specific feedback for bounded revision. Eligible numerical failures can be calibrated against measured terrain responses. Coarse plans may reserve water, green space, and public land before local streets and parcels are built; architectural templates are compiled into minimum plot and pad dimensions before allocation. During exploration, region and district choices constrain vegetation and building families, and each building reuses one design. Stable seeds introduce local variation without changing the authored world. Language-model inference stays outside per-chunk generation.

3.2 Interface-constrained local construction

A constructor is selected by content type $\tau(\iota)$ and receives the relevant shared interface $\mathcal{B}_\iota$:

$$C_\iota = \mathcal{F}_{\tau(\iota)}(\mathcal{D}, \iota, \mathcal{B}_\iota; \mathcal{A}), \tag{2}$$

where $\mathcal{A}$ is the reusable asset library. The interface carries terrain edge profiles and feature crossings for chunks, footprint and support for buildings, and openings and floor links for interiors. Constructors operate in metric coordinates and store local-to-world transforms. Thus, later requests inherit existing constraints rather than independently guessing adjoining geometry. Appendix B gives the construction order and the assumptions behind the shared-interface invariants.

Interface contract. Table 1 specifies the records passed between authoring, local construction, and streaming. The table makes each constructorś inherited spatial constraints explicit; the full construction order and the assumptions behind the shared-boundary and revisit invariants are in Appendix B.

Table 1: Inputs and outputs of the described construction interfaces. A record is committed by the runtime after construction.

Interface	Producer → consumer	Shared fields or identity
World definition	Authoring → constructors	Semantic and appearance attributes, terrain/infrastructure fields, settlement plans, building/interior programs, world seed
Chunk boundary	Shared fields → adjacent chunks	Canonical edge identity, sampled elevation profile, road/river crossing identities and geometry
Building frame	Parcel layout → building constructor	Parcel and design identity, footprint, orientation, supporting pad elevation, street frontage
Interior opening	Building shell → room constructor	Shell identity, local frame, floor elevation, doorway/opening geometry, available volume and floor links
Stored content	Constructor → runtime store/renderer	World and generator version, stable spatial/entity identity, geometry or specification, materials, placement, traversal data

For a world with several region types, the terrain program combines regional height fields g_k at world coordinate $\mathbf{x} \in \mathbb{R}^2$ with seed-dependent nonnegative weights w_k:

$$g(\mathbf{x}; s) = \sum_k w_k(\mathbf{x}; s) g_k(\mathbf{x}; s), \qquad w_k \geq 0, \quad \sum_k w_k = 1. \tag{3}$$

Region index k also selects appearance and vegetation; transition widths determine how the weights blend. Coarse samples guide hydrology, road routing, and settlement placement. Local road and river crossings have canonical IDs and carry position, width, elevation, tangent, and feature identity. Layout rules calibrated from OpenStreetMap road and footprint measurements relate streets, blocks, frontage, and parcel occupancy; Appendix Figure 3 shows generated and reference layouts at equal scale.

Both chunks incident to an edge derive the same elevation profile and crossings. After infrastructure adjustments, each chunk samples a shared height field h_0 on a boundary-inclusive grid and refines its

interior:

$$h_i^{(r+1)}(\mathbf{x}) = h_i^{(r)}(\mathbf{x}) + a_i(\mathbf{x})[1 - q(\mathbf{x})]\Delta h_i^{(r)}(\mathbf{x}), \qquad h_i^{(0)} = h_0|_{\Omega_i}. \tag{4}$$

Here r is the refinement step, $\Delta h_i^{(r)}$ the drainage or relaxation update, and $q \in [0, 1]$ protects channels, roads, and building pads. The envelope $a_i \in [0, 1]$ vanishes on the boundary and two neighboring sample rings; a shared-field halo provides seam-normal samples. Therefore neighboring chunks i and j at equal resolution retain $h_i = h_j = h_0$ at shared vertices, regardless of construction order. The same crossing records connect roads and rivers across the edge.

Street-facing parcel frames and pad elevations constrain building footprints, storeys, roofs, facade bays, and entrances. One design identity governs both coarse and detailed geometry, keeping massing, materials, and openings consistent across levels of detail. Assets and physically based materials are bound by semantic role and metric repeat length, so larger walls extend texture coverage without enlarging individual motifs; examples appear in Appendix Figure 4. Regional slope, moisture, and exclusion fields constrain vegetation and props. The building shell then supplies an interior frame, floor elevation, doorway, and available volume. Indoor programs reserve stair shafts and canonical horizontal openings before room subdivision; room adjacency supplies a connected door graph. Furniture is placed only after circulation is reserved: for usable room floor R, reserved passages Q, occupied area O, and candidate clearance footprint P, acceptance requires

$$P \subseteq R, \qquad P \cap (Q \cup O) = \varnothing. \tag{5}$$

Accepted footprints extend O. Rendering and collision use the same placement and opening records, including the exterior entrance and its interior floor connection.

3.3 Persistent streaming and revisits

Nested generation, retention, and display neighborhoods have radii $r_{\text{gen}} \geq r_{\text{keep}} > r_{\text{show}}$ around the user. Missing chunks are prioritized by proximity and built asynchronously by workers; completed records are published to the renderer. Coarse distant terrain and detailed nearby buildings share the same definitions and identities. Interior requests retain neighboring floor regions and the stair owner during vertical travel.

If traversal speed is v, effective generation lead distance is d, request-to-usable latency is ℓ, and λ_{build} and λ_{enter} are completed and newly required chunk rates, sustained traversal requires

$$d/v \geq \ell, \qquad \lambda_{\text{build}} \geq \lambda_{\text{enter}}. \tag{6}$$

Prefetching provides lead time, while worker concurrency supplies capacity. The runtime stores committed content under the world, generator version, and spatial identity, with hashes for immutable specifications and records. Eviction releases resident geometry and materials but preserves committed construction choices. A revisit loads the stored record, and subsequent generation sees the same identity and interfaces.

4 Experiments

We evaluate requirement satisfaction, physical validity, walkability, visual preference, long-range expansion, and online chunk availability as distinct outcomes. Scores lie in $[0, 1]$ with higher values indicating better outcomes. The latter two scores are system diagnostics under the specific runtime configurations below.

4.1 Experimental setup

Tasks and data. The quantitative evaluation comprises 233 runs across 14 task–seed configurations and six backbones: GLM 5.3 Flash, Gemini 3.8 Flash, GPT-6 Astra, GPT-6 Sol, Claude Opus 5.5, and Claude Fable 5.1. Quantitative generations use text commissions specifying buildings, roads, terrain, water, vegetation, and facilities; visual-reference and interior scenes are curated demonstrations. Each commission is decomposed into atomic requirements with a source phrase, capability category, check rule, evidence source, and threshold. The fixed check library and category-level task-profile definitions appear in Appendix A.1. The source does not provide per-configuration scores or metric-specific run counts, so these total counts should not be treated as the denominators of the table means.

Implementation and comparisons. Our pipeline authors a world program and settlement layout before deterministic local construction; there is no gradient training or batch size. We compare it with *Direct-finite*, which writes a self-contained Three.js scene, and *Direct-expandable*, which writes a coordinate-indexed terrain function and tile constructor. Each backbone receives the same commission, hard-requirement text, and role vocabulary under ceilings of 12 model calls, six world-authoring rounds, two layout-authoring rounds where applicable, and a 3600 s model-facing budget; direct generation may repair its output within that budget. The initial window is a 3×3 grid of $256\,\mathrm{m}$ chunks. Captures use 1280×720 resolution with time and weather fixed; each method retains its renderer. Our runtime uses three workers and display/retention/generation radii of 2/3/3 chunks; Direct-expandable constructs tiles synchronously in the browser. Within a backbone, compared methods use the same eligible task–seed conditions. Scores average equally over available runs within each backbone, then equally over available backbone means; unavailable metrics are omitted. Generation failures score zero for supported outcomes, and visual comparisons award one to a delivered arm when its counterpart fails.

Table 2 summarizes the reported overall scores. These comparisons evaluate complete systems: the direct-code baselines do not use our planners, solvers, asset library, or materializer. Per-backbone rows are in Appendix C. The score tables alone cannot estimate cross-task uncertainty or the effect of an individual component.

Table 2: Reported overall scores, ours/direct baseline, formed by equally averaging available backbone means. Req. is verified requirement satisfaction, Walk is route success, and Visual is pairwise preference. ↑ indicates higher is better. The source does not provide metric-specific run counts or per-task breakdowns. Bold marks the higher value in each pair.

Comparator	Req. ↑	Physical ↑	Walk ↑	Visual ↑
Direct-finite	$\mathbf{0.50}/0.36$	$\mathbf{0.88}/0.39$	$\mathbf{1.00}/0.70$	$\mathbf{0.59}/0.41$
Direct-expandable	$\mathbf{0.42}/0.31$	$\mathbf{0.80}/0.31$	$\mathbf{0.80}/0.37$	$\mathbf{0.81}/0.19$

4.2 Instruction fidelity and local usability

A frozen requirement receives a verified pass (P), verified failure (F), or unresolved verdict (U). Requirement satisfaction is $P/(P + F + U)$, so unsupported requests and missing evidence remain in the denominator. Building instances pass a common physical gate only when rendered footprint and height are at least $12\,\mathrm{m}^2$ and $2.5\,\mathrm{m}$, at least 90% of contact samples lie within $0.10\,\mathrm{m}$ of support, no certified collision occurs, and the footprint lies within 15 m of a walkable road. The physical score is the valid fraction of evaluated building instances. Storey and spatial-relation checks use measured geometry rather than role labels alone.

Walkability uses one offline capsule-walker proxy for both methods over rasterized terrain, water, and inflated building footprints. A common policy chooses up to three 150–$300\,\mathrm{m}$ routes on each world's road network, with at most 50% shared path length. The walker takes 0.25 m steps with a 0.30 m radius and maximum upward/downward steps of 0.60/2.0 m; water deeper than 0.60 m blocks travel outside road corridors. Routes are chosen separately from each world, so endpoint difficulty is not matched.

Against Direct-finite, the reported backbone-mean physical score is 0.88 versus 0.39, and walkability is 1.00 versus 0.70 (Table 7). Physical validity favors ours in all six backbone rows. Requirement satisfaction is 0.50 versus 0.36 and favors ours for five of six backbones; Claude Opus 5.5 favors Direct-finite. These observations support the integrated system's local usability on the reported evaluation, without identifying which component produces the gain.

4.3 Expandable worlds and online availability

Direct-expandable tests coordinate-indexed generation without our authored construction stack. Our system scores 0.42 versus 0.31 for requirements, 0.80 versus 0.31 for physical validity, and 0.80 versus 0.37 for walkability (Table 8). Physical validity favors ours for five of six backbone rows; Claude Opus 5.5 favors Direct-expandable. The differing construction resources and separately selected routes limit causal interpretation.

The long-range policy uses a $1.5\,\mathrm{km}$ outward route and return, four revisit points, nominal speed 3 m/s, and a 2700 s capture budget. The stress camera can bypass collisions, float over water, and skip missing ground; collision-aware walking is scored separately above. Fixed world coordinates and camera poses define each revisit observation, with noon and clear weather. The four-component diagnostic averages occupancy IoU, same-view SSIM, the far/near physical-validity ratio capped at one, and one minus seam-connection error. A missing component scores zero unless all components are unobserved. Our score is 0.82 versus 0.55 (Table 3). Direct-expandable declares no road/river boundary connections, so its connection component is unmeasured and scored zero, capping its composite at 0.75. The gap combines consistency and observation coverage rather than isolating revisit fidelity.

Online availability is one minus the chunk deadline-miss rate, counting never-delivered chunks as misses. Its reported backbone mean is 0.95 versus 0.62; the rounded complements correspond to approximate mean miss scores of 0.05 and 0.38. These are deadline outcomes under different runtime configurations and shared-machine contention, not measured frame rate, chunk latency, or throughput. Appendix A documents the collection conditions.

Table 3: System-level expansion diagnostics, ours/Direct-expandable. ↑ indicates higher is better. The long-range score assigns zero to missing components, including the baseline's undeclared connection term; the two runtimes use different routes, initial states, and execution modes.

Metric	Ours	Baseline
Long-range composite ↑	0.82	0.55
Online availability ↑	0.95	0.62

4.4 Component-aligned interpretation

Direct-finite and Direct-expandable replace the full authored construction stack, with the latter also implementing coordinate indexing. The persistent physical and route gaps show that coordinate indexing in that baseline does not reproduce the complete system's measured local usability. They do not isolate program authoring, shared interfaces, or streaming because the methods also differ in assets, construction primitives, rendering, and runtime. Appendix B separates observed system scores from conditional construction invariants and the absent one-component-at-a-time comparisons.

4.5 Perceptual quality

We compare six matched views of each generated scene: three eye-level, two oblique, and one overhead view near the densest building group. A single GPT-5.6 Sol judge rates naturalness, requirement match, content richness, and layout in both anonymous left–right orders; inconsistent decisions become ties. Preference is $(W + 0.5T)/(W + L + T)$. The aggregate Visual score averages the four dimensions and repeated views within each run, then eligible runs within each backbone and available backbone means with equal weight. The judge prompt and rubric appear in Appendix A. This is a model-based diagnostic without human calibration.

The reported aggregate visual preference is 0.59 against Direct-finite and 0.81 against Direct-expandable, including generation-failure handling. Table 4 separates dimensions for available rendered views; the source does not report their effective view counts or weighting across backbones. Naturalness, richness, and layout favor ours, whereas image-based requirement match favors both baselines. Visual plausibility and physical usability therefore need separate assessment from semantic compliance.

4.6 Qualitative results

The Japanese-town prompt requests a human-scale timber settlement organized around a public square, with lanes leading to a stream, bridge, and woodland, under coherent daylight. It also requests consistency across viewpoints and during outward exploration. The full prompt is given in Appendix F. Our result uses Claude Opus 5.5 as the backbone; the displayed outputs use text input without reference images. Figure 5 shows labeled outputs in four view categories, exposing

Table 4: Reported dimension-level visual preference (%) from available rendered views, shown as ours/baseline. ↑ indicates higher is better. Ties count as one half; bold marks the preferred side. Generation failures enter the overall Visual score in Table 2.

Comparator	Naturalness ↑	Requirement match ↑	Content richness ↑	Layout ↑
Direct-finite	**84.3**/15.7	20.4/**79.6**	**75.9**/24.1	**54.6**/45.4
Direct-expandable	**93.8**/6.2	46.9/**53.1**	**88.5**/11.5	**76.0**/24.0

building massing, street-level detail, vegetation, and settlement layout. Construction protocols for the comparison rows are unspecified, so this figure is illustrative and does not support a controlled ranking. Appendix E contains four outdoor reference-conditioned settings (city, coast, desert, Japanese town) and three indoor settings (office, service, pool), presented as curated examples rather than quantitative task replications. Table 10 summarizes the resulting task coverage and evidence type.

5 Discussion and Limitations

The measured gains concern an integrated system, and several protocol features limit their interpretation. The reported campaign includes 233 runs across 14 task–seed configurations, but the source gives no per-task scores or metric-specific run counts, and some tasks were used during system development; visual-reference and indoor examples are curated demonstrations rather than a held-out benchmark. The two direct-code baselines differ from our system in planning, construction, assets, and rendering, so the paired comparison cannot assign gains to any one module. In the long-range stress test, the arms follow different hashed headings and motion policies and begin from different initial states; moreover, missing connection declarations lower the baseline's four-part composite score by construction. That score therefore mixes geometric consistency with observation coverage and is not a controlled estimate of revisit fidelity alone. Visual preferences come from one model judge without human calibration, and image-based requirement match favors the baselines despite stronger geometry-based validity for our system. Finally, the term *unbounded* describes coordinate-addressable construction and persistent storage, not proof of infinite successful traversal; online deadlines were measured over the specified $1.5\,\mathrm{km}$ route. The most useful next evidence is per-task scores and uncertainty on a held-out set of diverse commissions, matched route and renderer conditions, human visual assessment, and module-wise ablations of authoring, interfaces, and streaming.

6 Conclusion

Walk and It Appears treats a reference as a persistent world program whose local constructions share spatial interfaces and whose completed records survive exploration. The system joins terrain, settlements, buildings, and interiors with a streaming runtime, producing physically usable scenes in the reported quantitative evaluation and illustrating visual and indoor settings. Paired comparisons show gains in physical validity, route success, and measured chunk availability against direct-code generation, alongside mixed semantic and visual-requirement outcomes. Per-task analysis with metric-specific counts, held-out prompts, controlled module ablations, and matched long-range protocols are needed to establish how broadly these gains transfer and which mechanisms drive them.

AI use statement

We used an AI assistant to aid and polish the writing of this manuscript, including improving clarity, grammar, and organization, and assisting with reference checks. The assistant was not used to generate experimental measurements. The authors are responsible for the final content, including all claims, results, and references.

REFERENCES

Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A New Frontier for World Models. Google DeepMind technical report, 2025. URL https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/.

Jinqi Cao, Zhiping Yu, Baihong Lin, Chenyang Liu, Zhenwei Shi, and Zhengxia Zou. MetaEarth3D: Unlocking World-scale 3D Generation with Spatially Scalable Generative Modeling, 2026. URL https://arxiv.org/abs/2604.22828.

Yiwen Chen, Guosheng Lin, and Chi Zhang. Code World Model: Coding Agent as World Brain, 2026. URL https://arxiv.org/abs/2608.25927.

Jie Deng, Wenhao Chai, Jianshu Guo, Qixuan Huang, Junsheng Huang, Wenhao Hu, Shengyu Hao, Jenq-Neng Hwang, and Gaoang Wang. CityGen: Infinite and Controllable City Layout Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2020–2030, June 2025.

Paul Engstler, Aleksandar Shtedritski, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. SynCity: Training-Free Generation of 3D Worlds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27585–27595, October 2025.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, Ka Leong Cheng, Haojie Zhang, Jian Gao, Tianrui Feng, Yuzheng Liu, Yao Yao, Yinghao Xu, Xing Zhu, Yujun Shen, and Hao Ouyang. Infinite Worlds with Versatile Interactions, 2026. URL https://arxiv.org/abs/2607.07534.

Chunchao Guo, Jinpeng Li, Yang Li, and Zilong Huang. WorldClaw: Agentic 3D Open-World Generation at Scale, 2026. URL https://arxiv.org/abs/2608.05248.

Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2Room: Extracting Textured 3D Meshes from 2D Text-to-Image Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7909–7920, October 2023.

Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A Ross, Cordelia Schmid, and Alireza Fathi. SceneCraft: An LLM Agent for Synthesizing 3D Scenes as Blender Code. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19252–19282. PMLR, 21–27 Jul 2024.

Zilong Huang, Jun He, Xiaobin Huang, Ziyi Xiong, Yang Luo, Junyan Ye, Weijia Li, Yiping Chen, and Ting Han. MajutsuCity: Language-driven Aesthetic-adaptive City Generation with Controllable 3D Assets and Layouts, 2025. URL https://arxiv.org/abs/2511.20415.

Han-Hung Lee, Cheng-Yu Yang, Yu-Lun Liu, and Angel X. Chang. NuiWorld: Exploring a Scalable Framework for End-to-End Controllable World Generation, 2026. URL https://arxiv.org/abs/2601.19048.

Jiaqi Li, Junshu Tang, Zhiyong Xu, Longhuang Wu, Yuan Zhou, Shuai Shao, Tianbao Yu, Zhiguo Cao, and Qinglin Lu. Hunyuan-GameCraft: High-dynamic Interactive Game Video Generation with Hybrid History Condition, 2025a. URL https://arxiv.org/abs/2506.17201.

Sikuang Li, Chen Yang, Jiemin Fang, Taoran Yi, Jia Lu, Jiazhong Cen, Lingxi Xie, Wei Shen, and Qi Tian. WorldGrow: Generating Infinite 3D World, 2025b. URL https://arxiv.org/abs/2510.21682.

Zhiqi Li, Yuxuan Liao, and Bo Zhu. Recursive Code World Models: Building Complex Worlds through Recursive Scene Programs, 2026. URL https://arxiv.org/abs/2609.11499.

Chieh Hubert Lin, Hsin-Ying Lee, Willi Menapace, Menglei Chai, Aliaksandr Siarohin, Ming-Hsuan Yang, and Sergey Tulyakov. InfiniCity: Infinite-Scale City Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22808–22818, October 2023.

Xiaoyan Liu, Jiaxin Liu, Kangrui Li, and Sifan Zhou. Streaming4D: Accelerate 4D World Models via Block-wise Video Generation and Incremental Reconstruction, 2026. URL https://arxiv.org/abs/2609.00610.

Xiaofeng Mao, Zhen Li, Chuanhao Li, Xiaojie Xu, Kaining Ying, Tong He, Jiangmiao Pang, Yu Qiao, and Kaipeng Zhang. Yume-1.5: A Text-Controlled Interactive World Generation Model, 2025. URL https://arxiv.org/abs/2512.22096.

Ryan Po, David Junhao Zhang, Amir Hertz, Gordon Wetzstein, Neal Wadhwa, and Nataniel Ruiz. MultiGen: Level-Design for Editable Multiplayer Worlds in Diffusion Game Engines, 2026. URL https://arxiv.org/abs/2603.06679.

Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite Photorealistic Worlds Using Procedural Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12630–12641, June 2023.

Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, Zeyu Ma, and Jia Deng. Infinigen Indoors: Photorealistic Indoor Scenes using Procedural Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21783–21794, June 2024.

Tianchang Shen, Sherwin Bahmani, Kai He, Sangeetha Grama Srinivasan, Tianshi Cao, Jiawei Ren, Ruilong Li, Zian Wang, Nicholas Sharp, Zan Gojcic, Sanja Fidler, Jiahui Huang, Huan Ling, Jun Gao, and Xuanchi Ren. Lyra 2.0: Explorable Generative 3D Worlds, 2026. URL https://arxiv.org/abs/2604.13036.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. WorldPlay: Towards Long-Term Geometric Consistency for Real-Time Interactive World Modeling, 2026. URL https://arxiv.org/abs/2512.14614.

Dilin Wang, Hyunyoung Jung, Tom Monnier, Kihyuk Sohn, Chuhang Zou, Xiaoyu Xiang, Yu-Ying Yeh, Di Liu, Zixuan Huang, Thu Nguyen-Phuoc, Yuchen Fan, Sergiu Oprea, Ziyan Wang, Roman Shapovalov, Nikolaos Sarafianos, Thibault Groueix, Antoine Toisoul, Prithviraj Dhar, Xiao Chu, Minghao Chen, Geon Yeong Park, Mahima Gupta, Yassir Azziz, Rakesh Ranjan, and Andrea Vedaldi. WorldGen: From Text to Traversable and Interactive 3D Worlds, 2025. URL https://arxiv.org/abs/2511.16825.

Hanyang Wang, Yimo Cai, Weiliang Chen, Jiawei Chi, Haowen Sun, Qiyu Dai, Yi-Hsin Hung, Xingzhuo Guo, Jinshan Ren, Runmao Yao, Ziwei Liu, Mingsheng Long, Yueqi Duan, Jun Gao, Jiangran Lyu, Fangfu Liu, and Jialong Wu. Code as Worlds: Agentic Discovery of Executable World Representations for Physical Reasoning, 2026a. URL https://arxiv.org/abs/2608.27549.

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, Yidan Xietian, Jiangbo Pei, Liang Hu, Boyi Jiang, Hua Xue, Zidong Wang, Haofeng Sun, Wei Li, Wanli Ouyang, Xianglong He, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-Game 3.0: Real-Time and Streaming Interactive World Model with Long-Horizon Memory, 2026b. URL https://arxiv.org/abs/2604.08995.

Ruiqi Wu, Xuanhua He, Meng Cheng, Tianyu Yang, Yong Zhang, Zhuoliang Kang, Xunliang Cai, Xiaoming Wei, Chunle Guo, Chongyi Li, and Ming-Ming Cheng. Infinite-World: Scaling Interactive World Models to 1000-Frame Horizons via Pose-Free Hierarchical Memory, 2026. URL https://arxiv.org/abs/2602.02393.

Zhennan Wu, Yang Li, Han Yan, Taizhang Shang, Weixuan Sun, Senbo Wang, Ruikai Cui, Weizhe Liu, Hiroyuki Sato, Hongdong Li, and Pan Ji. BlockFusion: Expandable 3D Scene Generation using Latent Tri-plane Extrapolation. *ACM Transactions on Graphics*, 43(4):1–17, July 2024. doi: 10.1145/3658188. URL http://dx.doi.org/10.1145/3658188.

Zeqi Xiao, Yushi LAN, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. WorldMem: Long-term Consistent World Simulation with Memory. In *Advances in Neural Information Processing Systems*, volume 38, pp. 49632–49652. Curran Associates, Inc., 2025. doi: 10.52202/085713-1659.

Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. CityDreamer: Compositional Generative Model of Unbounded 3D Cities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9666–9675, June 2024.

Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. Generative Gaussian Splatting for Unbounded 3D City Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6111–6120, June 2025.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, Chris Callison-Burch, Mark Yatskar, Aniruddha Kembhavi, and Christopher Clark. Holodeck: Language Guided Generation of 3D Embodied AI Environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16227–16237, June 2024.

Yuanyang Yin, Gongxuan Wang, Yifan Zhan, Chuanhao Li, Kaipeng Zhang, and Feng Zhao. Alaya-EVOKE: From Linear-Scaling Supervision to Endless World, 2026. URL https://arxiv.org/abs/2608.13546.

Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T. Freeman, and Jiajun Wu. WonderWorld: Interactive 3D Scene Generation from a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5916–5926, June 2025.

Xiaoyu Zhan, Xinyu Wang, Xiaohong Zhang, Huanjie Zhu, Tengjiao Sun, Pengcheng Fang, Jiaxing Yu, Yanwen Guo, and Dongjie Fu. Magpie: Real-Time World Renderer for Interactive Games, 2026. URL https://arxiv.org/abs/2608.27168.

Haoyi Zhu, Haozhe Liu, Yuyang Zhao, Tian Ye, Junsong Chen, Jincheng Yu, Tong He, Song Han, and Enze Xie. SANA-WM: Efficient Minute-Scale World Modeling with Hybrid Linear Diffusion Transformer, 2026. URL https://arxiv.org/abs/2605.15178.

A ADDITIONAL BENCHMARK PROTOCOL DETAILS

The quantitative evaluation comprises 233 experimental runs across 14 task–seed configurations and six language-model backbones. This appendix records the scoring rules, runtime conditions, and model settings used for Table 2 and Appendix C.

Scoring and missing observations. Generation is successful when scene artifacts are available; physical validity and traversal are evaluated separately. Unsuccessful generation receives zero for supported scores. In a visual comparison, the method that produces an artifact receives one if the other method fails. For each metric, the compared methods use the same eligible task, seed, and backbone conditions with numerical observations. A wholly unavailable metric is excluded from its mean. Coverage can differ across comparisons, so the walkability average for our method can differ between the finite and expandable comparisons. Repeated views are averaged within a run before averaging over runs within each backbone. The Overall rows average the available backbone means with equal weight. Dimension-level visual preferences use available rendered views; generation failures are accounted for in the aggregate Visual scores.

Within the four-component long-range score, missing components contribute zero; the score is unavailable if all four components are missing. Repeated records of the same terrain seam are combined before evaluation, retaining all samples. Direct-expandable does not declare road or river boundary connections, so its connection component is unmeasured and contributes zero. Its attainable composite score is therefore 0.75. The long-range score measures both consistency and observation coverage.

Long-range collection conditions. Both methods are evaluated on a 1.5 km outward route followed by a return, with 50 m waypoint spacing and four revisit locations at 250, 600, 1000, and 1400 m. Each location is captured on both legs. Chunk width is 256 m, display/retention/generation radii are 2/3/3 chunks, and nominal speed is 3 m/s. The scene clock is stopped at noon with clear weather. The two methods use different route headings. Our real-time stress camera bypasses building collisions and travels over water, whereas the baseline's in-page walker skips collisions in 3 m steps. Our runtime begins with nine delivered chunks and expands with three background workers; Direct-expandable begins empty and constructs tiles synchronously in the browser. Each method retains its own renderer. Thus, the shared distance and speed define a system-level stress test with different trajectories, motion policies, and starting states.

Runtime environment. Collection uses an AMD Ryzen 9 9950X CPU (16 cores, 32 threads), 64 GB of RAM, an NVIDIA GeForce RTX 5090D GPU, and headless Chrome. Online expansion uses three generation processes. Experiments run on shared hardware, so available resources vary across runs.

Models and visual judge. Generation uses GLM 5.3 Flash, Gemini 3.8 Flash, GPT-6 Astra, GPT-6 Sol, Claude Opus 5.5, and Claude Fable 5.1. Compared methods use the same backbone. GLM 5.3 Flash uses constrained JSON decoding for structured authoring. The visual judge is GPT-5.6 Sol with medium reasoning effort. It is separate from the evaluated backbones but shares a model family with the GPT entrants; human calibration has not been performed. Runs interrupted by infrastructure failures were repeated. Some evaluation tasks were also used during system development.

Visual judge prompt and rubric. The judge receives the original commission and two anonymized images. Each image comparison is evaluated in both presentation orders; a side wins only if both judgments select it, and all other outcomes are ties. The system prompt is:

> You are an expert evaluator of generated 3D virtual worlds. You will see two images of two DIFFERENT generated worlds made for the same commission, each rendered from the same camera placement relative to its settlement centre. Judge only what is visible in the images. Do not reward or penalise image resolution or compression. The order of the images is random and carries no information.

The four evaluation dimensions are:

Naturalness. Plausibility as a real place: terrain, water and vegetation look natural; buildings and objects sit on the ground at believable scale; materials and lighting are coherent. Floating, buried, intersecting or broken geometry counts against.

Requirement fit. How well the visible world matches the commission text: the kind, size and character of the settlement and landscape it asks for, and the absence of things it rules out.

Detail. Richness and variety of meaningful visual detail (building forms, roofs, streets, vegetation types, props). Repetitive copies or visual noise are not detail.

Layout. Organisation of the settlement: streets and paths connect, buildings face and line streets sensibly, public spaces and districts read coherently. If no settlement is visible, judge how coherent the visible arrangement is.

For each dimension, the judge selects image 1, image 2, or a tie and gives a one-sentence reason. The prompt instructs the judge to choose a tie only when the images are genuinely equal on that dimension.

A.1 Requirement checks and task profiles

A fixed library maps commission fragments to observable checks (Table 5). Each atomic requirement retains its source phrase, assigned capability category, check rule, evidence source, and threshold. A verified pass (P) requires supporting evidence; a verified failure (F) requires contrary evidence; missing or insufficient evidence gives an unresolved verdict (U). Unsupported requests also receive U and remain in the denominator of requirement satisfaction. A generation-time role label alone is not evidence.

Table 5: Requirement-check categories and evidence sources. Each atomic requirement records its source phrase, capability category, check type, required evidence, and threshold. Checks use observable properties of the delivered world; unsupported requirements remain in the scoring denominator. This table defines the check library; per-task category counts are not available.

Capability	Observable checks	Required evidence
Existence and attributes		
Settlement presence and scale	Settlement presence; physically valid building count within the requested range	Exported layout
Building functions	Presence and count of requested buildings or facilities	Exported layout
Land cover	Fields, woodland, and other requested land-cover types	Layout, chunk contents, terrain rasters
Building attributes	Storey count and material configuration	Layout, effective world configuration
Street scale	Length of the main street	Exported layout
Water presence	Presence of a river	Chunk specifications, runtime manifest
Relations and topology		
Road topology	Through routes, roads meeting at junctions, and radial density falloff	Chunk specifications, layout, runtime manifest
Road–water relations	Number and location of road–river crossings	Chunk specifications, runtime manifest
Functional placement	Proximity to a feature; frontage on a specified street	Layout, terrain and water rasters
Building–terrain relations	Elevation band, contour following, steep-slope connectors, and ground drop across a footprint	Layout, terrain and slope rasters
Building adjacency	Shared walls between adjacent buildings	Exported layout
Composite requirements	All constituent conditions must hold	Combined evidence above
Unsupported requirements	Requested properties outside the fixed check library	Retained as unresolved

For category c, let P_c, F_c, and U_c count verdicts across the evaluated worlds, and let $N_c = P_c + F_c + U_c$. Its pass and unresolved rates are P_c/N_c and U_c/N_c when $N_c > 0$. With $\mathcal{C} = \{c : N_c > 0\}$, the

micro pass rate is $\sum_{c\in\mathcal{C}} P_c / \sum_{c\in\mathcal{C}} N_c$; the macro pass rate is $|\mathcal{C}|^{-1} \sum_{c\in\mathcal{C}} P_c/N_c$. For task t, let $N_{t,c}$ be the number of its atomic requirements assigned to category c; the task profile $q_{t,c} = N_{t,c} / \sum_j N_{t,j}$ describes its mix of requested capabilities. These definitions permit category-aware reporting when per-task verdict records are available. The current score tables do not include those counts.

B COMPONENT EVIDENCE AND ABLATION IDENTIFIABILITY

The available comparisons replace the complete authored construction stack. They do not contain a run that disables one of the three proposed components while retaining the same assets, renderer, and remaining components. Table 6 distinguishes the directly specified construction property from the observed system-level metric and the missing causal comparison.

Table 6: Component-aligned evidence from the reported overall scores. Pairs are ours/Direct-expandable; ↑ indicates higher is better. These are system comparisons, not module ablations.

Component	Constructed property	Available observation	Needed ablation
Program authoring	Checked terrain, layout, building, and room specification	Requirement ↑ $0.42/0.31$; visual requirement match favors baseline	Same constructors with unchecked authoring
Shared interfaces	Equal-resolution shared terrain vertices and canonical crossings	Physical ↑ $0.80/0.31$; walk ↑ $0.80/0.37$	Same program and assets with local-only interfaces
Persistent streaming	Stable committed records and bounded resident set	Online ↑ $0.95/0.62$; long-range diagnostic ↑ $0.82/0.55$	Same constructors without prefetch or record retrieval

Construction order and shared records. An outdoor edge is named by its axis and integer grid coordinates. Both incident chunks query that identity for the same elevation profile, sampled normal derivative, and road or river crossings; a crossing records position, width, elevation, tangent, and feature identity, with flow direction and discharge for rivers. The shared height field is sampled on a boundary-inclusive grid before local drainage or thermal updates. The update envelope is zero on the boundary and two neighboring sample rings and rises through a quintic transition, while a halo from the shared field supplies seam-normal samples. For a building, the parcel and prepared pad establish the local frame before the chosen design is assembled; the shell then supplies the interior frame, floor elevation, doorway, and available volume. For multiple floors, the stair shaft is reserved before room subdivision, and the lower of its two endpoint floors owns the stair geometry. Room adjacency then defines candidate door connections; the room graph selects a spanning tree and optional extra links. Horizontal openings have canonical edge identities and divide the boundary wall between incident regions. Door approaches and circulation strips are reserved before furniture placement. These steps describe the implemented dependency order in the supplied method; their isolated effects have not been measured.

Shared-boundary invariant. For neighboring equal-resolution chunks i and j, let $\mathbf{x}$ be a shared grid vertex. Both sample the same $h_0(\mathbf{x})$ by canonical edge identity. Equation 4 sets $a_i(\mathbf{x}) = a_j(\mathbf{x}) = 0$ on the edge; induction over refinement steps r gives $h_i^{(r)}(\mathbf{x}) = h_j^{(r)}(\mathbf{x}) = h_0(\mathbf{x})$. Canonical feature IDs likewise select the same road and river crossing records on both sides. This is a construction invariant under the specified equal-resolution sampling and shared-record assumptions; it is not a measured seam-error ablation.

Revisit identity invariant. If (ι, C_ι) has been committed to $\mathcal{S}_t$, renderer eviction releases only the resident copy. A revisit under the same world and generator version retrieves C_ι, so geometry and collision records retain the committed identity. This statement depends on successful storage and retrieval and does not imply a measured latency, frame-rate, or photometric-consistency guarantee.

C PER-BACKBONE RESULTS AND SUPPORTING FIGURES

Table 7: Comparison with finite direct generation across six backbones. Scores include generation failures. ↑ indicates higher is better; bold marks the larger value within each backbone and in the overall summary.

Backbone	Method	Req. ↑	Physical ↑	Walk ↑	Visual ↑
GLM 5.3 Flash	Ours	**0.53**	**0.95**	**1.00**	**0.60**
	Direct-finite	0.41	0.59	**1.00**	0.40
Gemini 3.8 Flash	Ours	**0.50**	**0.90**	**1.00**	**0.70**
	Direct-finite	0.26	0.05	0.50	0.30
GPT-6 Astra	Ours	**0.53**	**0.90**	**1.00**	**0.52**
	Direct-finite	0.09	0.00	0.00	0.48
GPT-6 Sol	Ours	**0.53**	**0.96**	**1.00**	**0.68**
	Direct-finite	0.41	0.45	0.67	0.32
Claude Opus 5.5	Ours	0.41	**0.75**	**1.00**	0.42
	Direct-finite	**0.76**	0.74	**1.00**	**0.58**
Claude Fable 5.1	Ours	**0.47**	**0.80**	**1.00**	**0.60**
	Direct-finite	0.24	0.52	**1.00**	0.40
Overall	Ours	**0.50**	**0.88**	**1.00**	**0.59**
	Direct-finite	0.36	0.39	0.70	0.41

Table 8: Comparison with expandable direct generation across six backbones. Scores include generation failures. ↑ indicates higher is better; bold marks the larger value within each backbone and Overall row, with both values bold for a displayed tie; – denotes an unavailable measurement.

Backbone	Method	Req. ↑	Physical ↑	Walk ↑	Visual ↑	Online ↑
GLM 5.3 Flash	Ours	**0.26**	**0.47**	**0.00**	**0.88**	**1.00**
	Direct-expandable	**0.26**	0.30	**0.00**	0.12	0.41
Gemini 3.8 Flash	Ours	**0.50**	**0.90**	–	**0.80**	**1.00**
	Direct-expandable	0.38	0.31	–	0.20	**1.00**
GPT-6 Astra	Ours	**0.53**	**0.90**	**1.00**	**0.70**	**0.89**
	Direct-expandable	0.26	0.16	0.83	0.30	0.52
GPT-6 Sol	Ours	**0.53**	**0.96**	**1.00**	**0.75**	0.78
	Direct-expandable	0.18	0.09	0.00	0.25	**1.00**
Claude Opus 5.5	Ours	0.41	0.75	**1.00**	**0.73**	**1.00**
	Direct-expandable	**0.76**	**0.97**	**1.00**	0.27	0.76
Claude Fable 5.1	Ours	**0.29**	**0.80**	**1.00**	**1.00**	**1.00**
	Direct-expandable	0.00	0.00	0.00	0.00	0.00
Overall	Ours	**0.42**	**0.80**	**0.80**	**0.81**	**0.95**
	Direct-expandable	0.31	0.31	0.37	0.19	0.62

Backbone-level directional check. Table 9 exposes the signs hidden by equal-backbone overall means. Each entry counts backbone rows where our displayed score wins, ties, or loses against the corresponding direct baseline. Counts use rounded values; displayed ties may conceal differences at higher precision. Backbones are not independent task replicates; the table does not supply cross-task uncertainty.

Table 9: Wins/ties/losses for our system across available backbone rows, computed from the displayed per-backbone values. ↑ denotes the higher-is-better direction of the underlying score, not of the count triplet. The expandable walk metric has five available rows; all other listed metrics have six.

Comparator	Req. ↑	Physical ↑	Walk ↑	Visual ↑	Online ↑
Direct-finite	$5/0/1$	$6/0/0$	$3/3/0$	$5/0/1$	–
Direct-expandable	$4/1/1$	$5/0/1$	$3/2/0$	$6/0/0$	$4/1/1$



Figure 3: Layout examples from real and generated settlements. Road networks and building footprints from OpenStreetMap (Tokyo and Seoul) are shown alongside two independently generated districts from our system. All panels cover a $1\,\mathrm{km} \times 1\,\mathrm{km}$ window at the same scale. Gray polygons indicate building footprints, and lines indicate roads. OSM data: © OpenStreetMap contributors.



Figure 4: Examples of reusable assets and materials. Top: selected 3D assets from the scene library. Bottom: material base-color maps used in the generated Japanese settlement. Asset previews are rendered from the existing library models and individually normalized for display.

D JAPANESE-TOWN QUALITATIVE COMPARISON

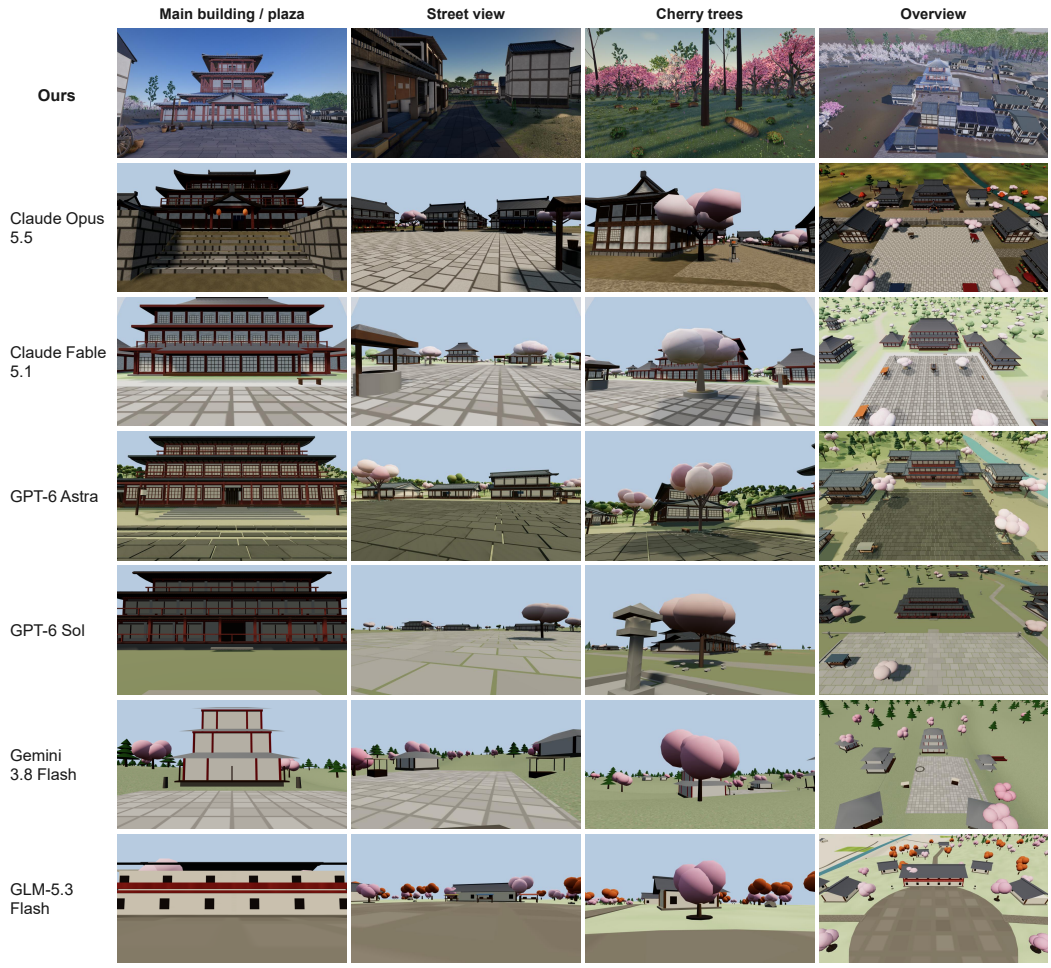

Figure 5: Japanese-town illustrative outputs for the prompt in Appendix F. Our system uses Claude Opus 5.5; displayed outputs use text input without reference images. Rows identify our system and the direct-code baselines. Columns show the plaza, street, cherry trees, and overview. No quantitative claim is drawn from these rows.

E ADDITIONAL QUALITATIVE RESULTS

Figures 6–10 present curated demonstrations across five environments. Each outdoor showcase pairs its visual references with terrain and layout views, followed by rendered views at several scales. The indoor showcase pairs its references with floor plans and room views.

Table 10: Campaign size and illustrated task coverage. Counts of illustrated settings are descriptive and do not constitute independent quantitative test cases.

Task group	Input	Count	Available evidence
Quantitative campaign	Text	233 runs; 14 task–seed configurations	No per-configuration score breakdown available
Japanese-town comparison	Text	1 prompt	Selected four-view figure; no controlled metric
Outdoor showcases	Visual references	4 settings	Curated terrain, layout, and scene views
Indoor showcase	Visual references	3 room settings in one figure	Curated floor plans and room views

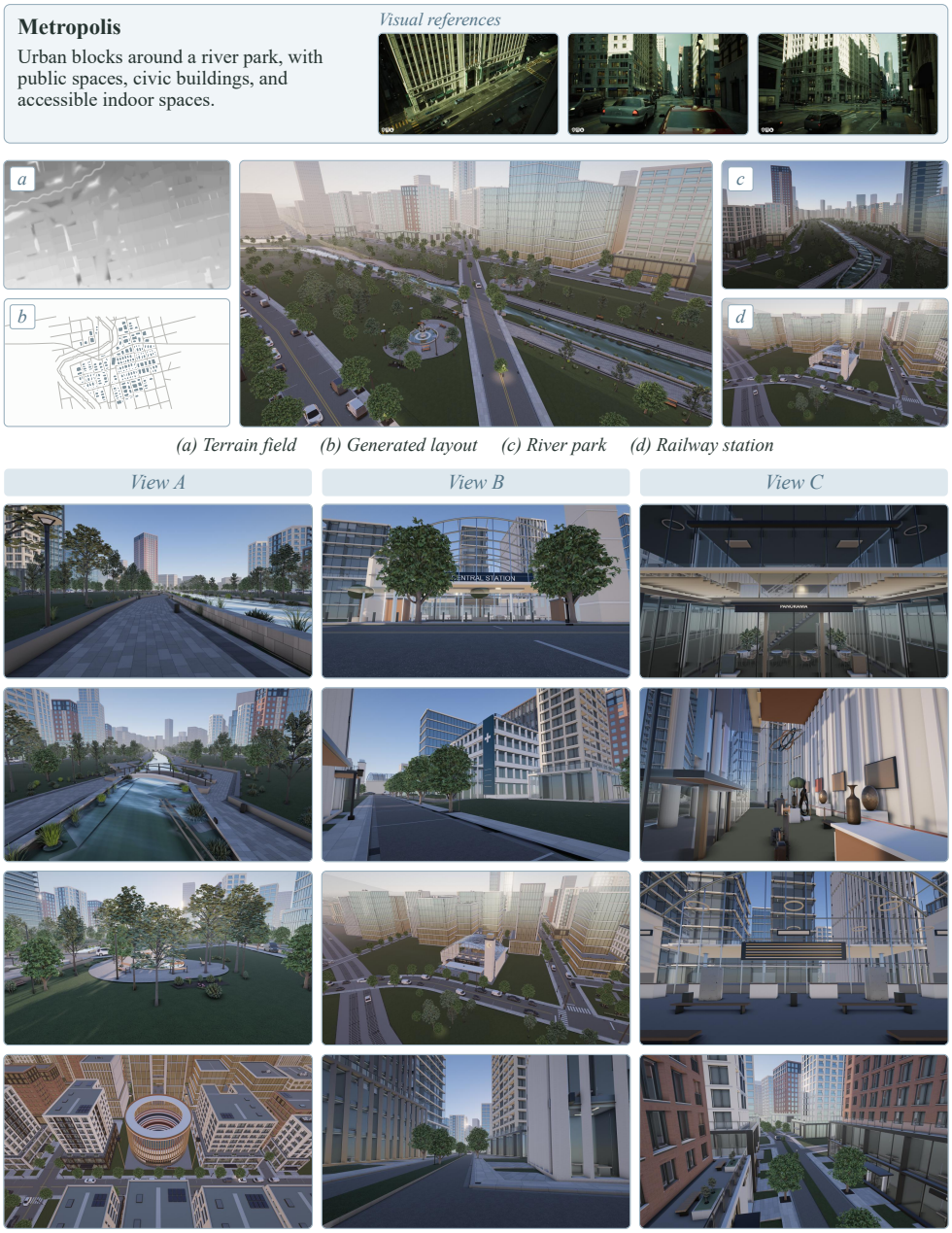

Figure 6: Modern city. Visual references and terrain/layout views precede renders of the river park, railway station, streets, public spaces, and building interiors. The views connect the settlement plan with details visible during ground-level exploration.

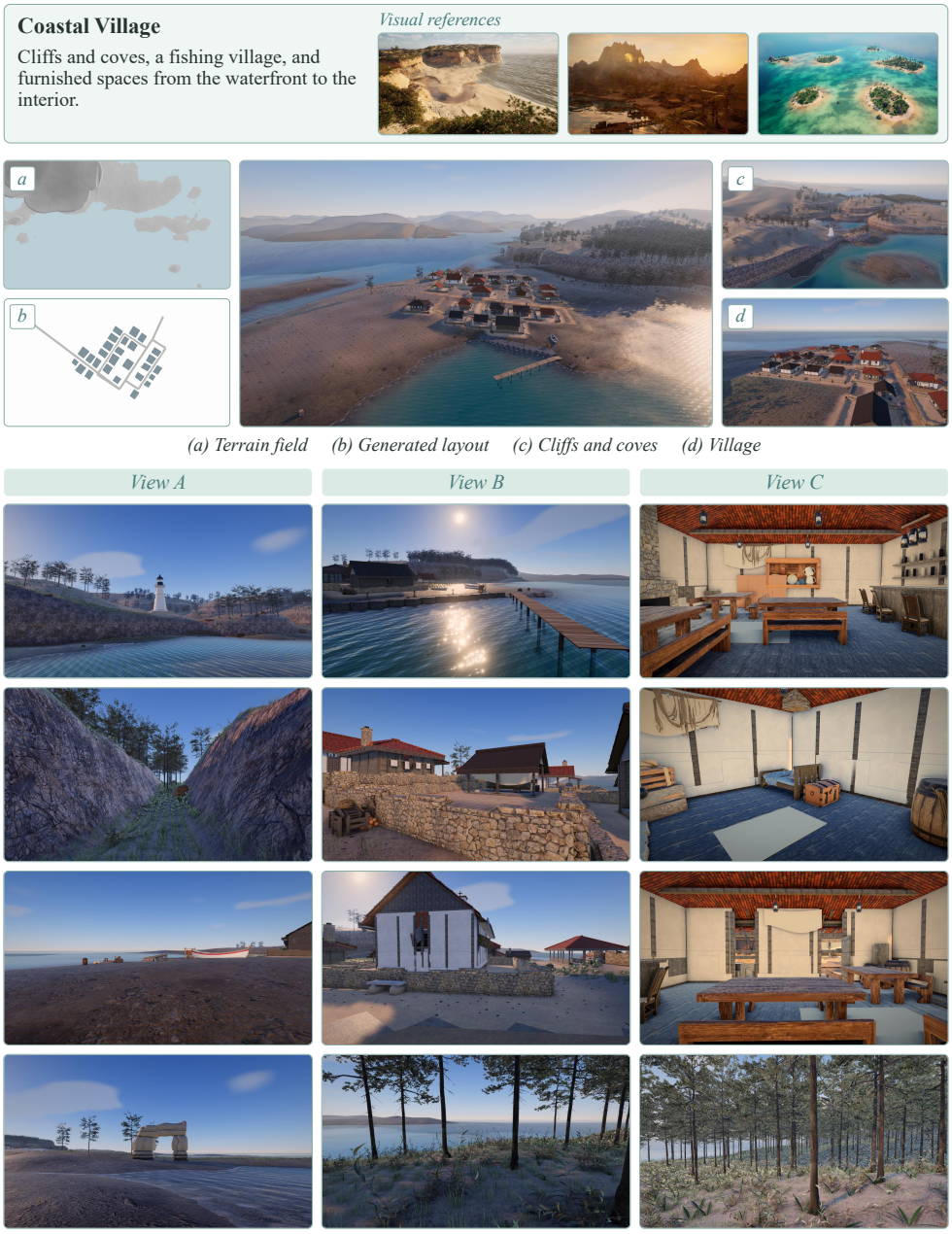

Figure 7: Coastal village. Reference images guide a settlement among cliffs and coves. Terrain and layout views locate the village along the waterfront; the rendered views show coastal paths, houses, furnished interiors, and nearby woodland.

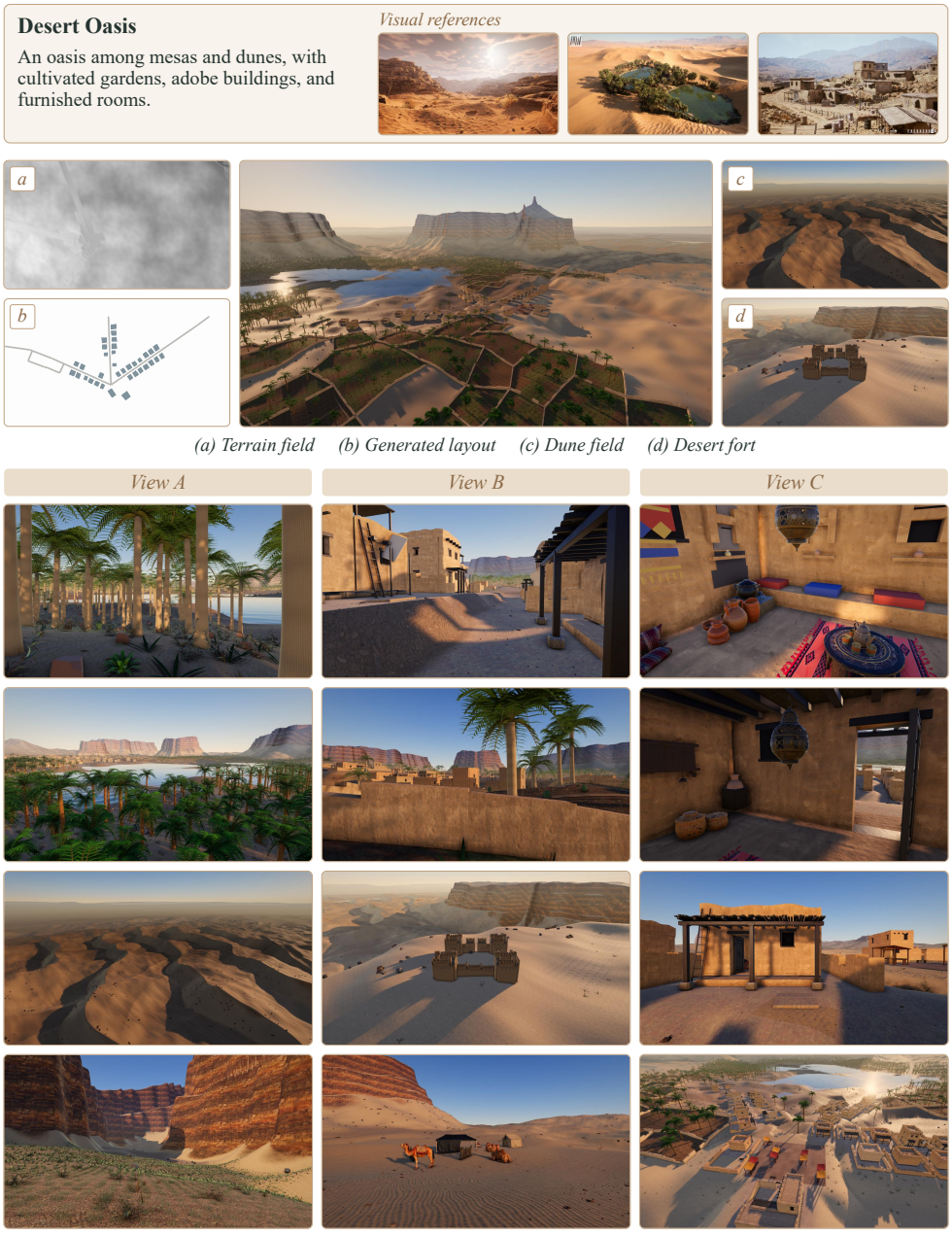

Figure 8: Desert oasis. Terrain and layout views show the settlement among mesas and dunes. Ground-level and overview renders include cultivated vegetation, adobe buildings, furnished interiors, and the desert fort.



Figure 9: Japanese mountain village. Reference images, terrain, and layout views establish the setting beneath Mount Fuji. The rendered views show timber streets, a cherry grove, a forest shrine, public buildings, and furnished interiors.

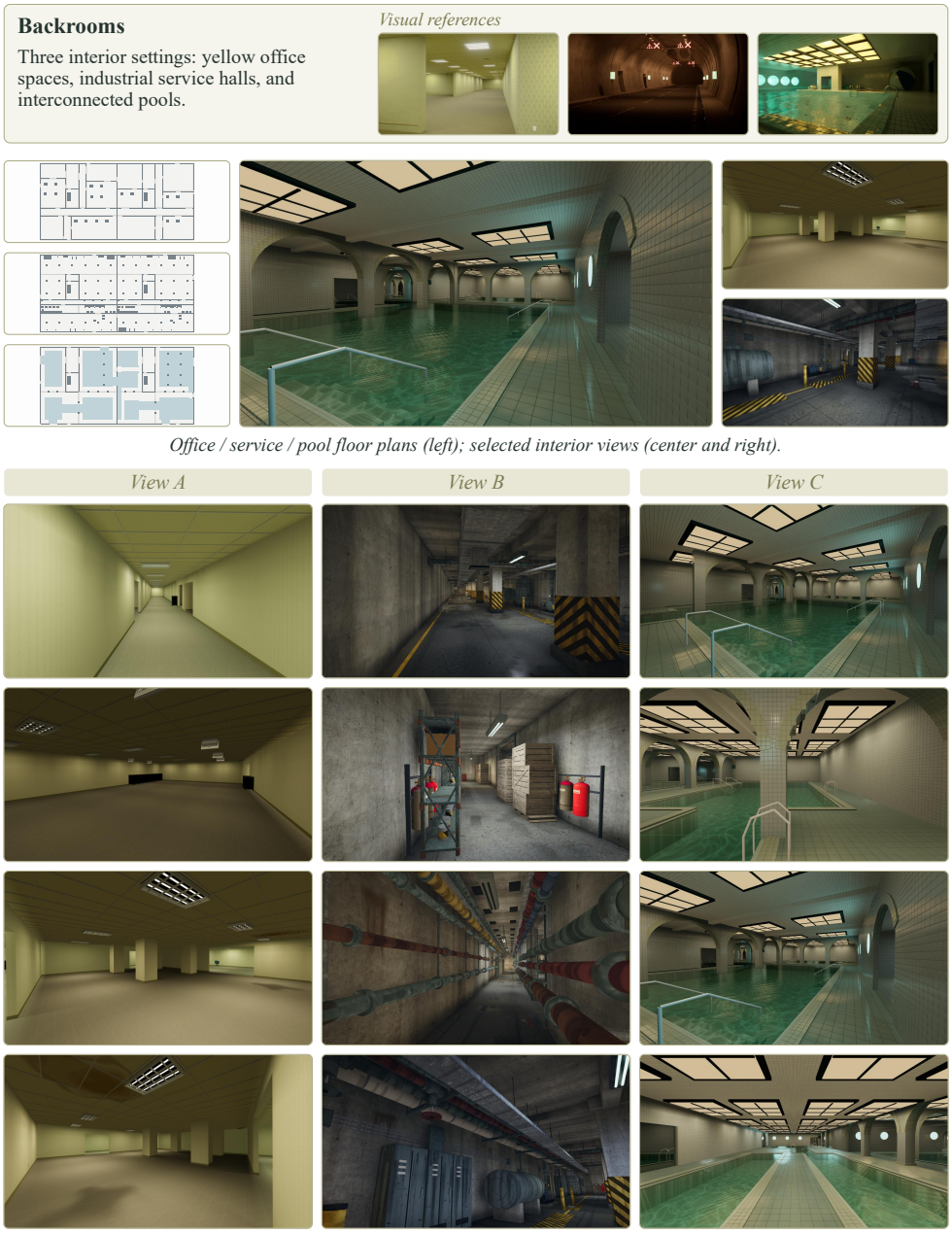

Figure 10: Indoor worlds. Visual references accompany floor plans for office, service, and pool settings. The three view columns show yellow office spaces, industrial service halls, and interconnected pools, respectively.

F INPUT FOR THE JAPANESE-TOWN COMPARISON

All methods in Figure 5 use text input without reference images. Our result uses Claude Opus 5.5 (`claude-opus-5-5`) as the backbone. The shared text prompt is reproduced verbatim below.

> Create an explorable, human-scale Japanese village in a natural landscape, with the visual atmosphere of Ghost of Tsushima. A stone-paved public square is enclosed by a layered group of timber buildings: a multi-storey main building, lower secondary houses, annexes, porches, balconies, thick curved grey tiled roofs, timber eaves, posts and beams. Use restrained red timber for the main building, brown timber for secondary buildings, grey-white walls and grey stone. Include stone steps connecting changes in elevation, narrow lived-in lanes, tree pits, small stalls and useful clustered props. Connect the square and lanes to a stream crossed by a bridge, then to waterside paths and woodland. Cherry trees and other vegetation frame the architecture; distant wooded hills and a Fuji-like mountain continue the landscape. Use coherent directional daylight, convincing materials, natural vegetation and clear foreground, middle-ground and background. The place should remain coherent when viewed from several directions and while moving outward from its centre at ground level. Build a spatial world, not just a single composed view.