[TINY PAPER] GEST-ENGINE: CONTROLLABLE MULTI-ACTOR VIDEO SYNTHESIS WITH PERFECT SPATIOTEMPORAL ANNOTATIONS

Nicolae Cudlenco[1,3], **Mihai Masala**[2] **& Marius Leordeanu**[1,2]
[1]Institute of Mathematics of the Romanian Academy, Bucharest, Romania
[2]National University of Science and Technology Politehnica Bucharest, Romania
[3]Büchi Labortechnik AG, Flawil, Switzerland
{nicolae.cudlenco,mihaimasala,leordeanu}@gmail.com
cudlenco.n@buchi.com

ABSTRACT

The world is a complex and dynamic place with multiple concurrent events that happen constantly between entities such as people and objects. While large-scale datasets with annotations obtained manually or through automatic post-processing of videos exist and facilitate the training of world models, few of them capture this complexity with ground-truth annotations. We introduce GEST-Engine, a system that given a Graph of Events in Space and Time (GEST) — a specification that encompasses entities, events, and temporal constraints — makes use of game world simulation to generate in a controlled manner, complex multi-actor and multi-object videos with pixel-level ground-truth annotations and frame-synchronized temporal segments, cross actor temporal relations, and cross entity spatial relations, reverse-mapped to the initial specification. We describe our complete end-to-end workflow that encompasses random GEST generation, a scalable pipeline for artifact generation and collection, and a sample corpus of 398 multi-actor videos spanning 37 action types with dense annotations at zero marginal cost.

1 INTRODUCTION

There have been impressive advances in neural world models and video generation lately, all made possible by large-scale data — whether self-generated through environment interaction (Ha & Schmidhuber, 2018; Hafner et al., 2023; Alonso et al., 2024), scraped from the internet (Bruce et al., 2024; Assran et al., 2025), or drawn from undisclosed proprietary corpora (Brooks et al., 2024; Ball et al., 2025; Wan et al., 2025; HaCohen et al., 2024; 2026). However, while world state perception has advanced rapidly, studies such as Chen et al. (2026) observe that robust data is still limited in the realm of action understanding. Their Action100M addresses this with 147 million temporally localized segments from 1.2 million instructional videos, but annotates individual actions performed by single actors in sequential steps, providing only temporal segments and captions.

Other synthetic video generation experiments have been done (Richter et al., 2016; Black et al., 2023), demonstrating the potential of synthetic data in real-world applications. Although other simulation engines exist (Masala et al., 2023; Ji et al., 2020; Dosovitskiy et al., 2017; Puig et al., 2018), the generated videos contain basic sequential execution and lacks temporal orchestration for complex multi-actor coordination (e.g., before, after) and are limited in the produced dense annotations.

In this work, we present GEST-Engine, a system that orchestrates multiple actors executing parallel actions with explicit temporal coordination governed by Allen interval algebra (Allen, 1983), alongside a procedural GEST generator and batch production orchestrator. Beyond temporal segments, we log detailed spatial relations between all entities at every frame, produce per-frame texture segmentation masks, and record complete scene graphs with causal and logical edges — alongside camera parameters for each frame, and dense textual descriptions — all as deterministic ground truth at zero marginal annotation cost.

2 THE GEST SPECIFICATION

A GEST is a directed graph $G = (\mathcal{V}, \mathcal{E})$ where nodes $\mathcal{V}$ represent events and edges encode relationships between events (such as, but not limited to temporal, spatial, logical or semantic) (Masala et al., 2023). Each event node $v_i = (a_i, p_i, O_i, \ell_i, \pi_i)$ specifies an action a_i, a performer p_i, participating entities O_i, a location ℓ_i, and properties π_i. Special `Exists` nodes declare actors and objects, establishing the entity inventory.

In this work we focus on temporal and spatial relations, with temporal edges such as $\{\texttt{before}, \texttt{after}, \texttt{same\_time}, \texttt{concurrent}\}$, constraining and coordinating execution order across actors and actions. Spatial edges such as $\{\texttt{on}, \texttt{near}, \texttt{left}, \texttt{right}, \texttt{in\_front}, \texttt{behind}\}$, describe entity arrangement during execution.

This structure provides three guarantees by construction: **(i) object permanence** — entities declared via `Exists` nodes persist throughout the story; **(ii) causal consistency** — the engine enforces graph-specified event ordering; and **(iii) multi-actor coordination** — concurrent events across actors are synchronized through shared temporal constraints, with execution ordering derived via transitive closure (Section 3).

3 SYSTEM ARCHITECTURE

GEST-Engine transforms a GEST specification into annotated video through four stages.

Stage 1: Graph parsing and validation. The engine extracts actors from `Exists` nodes, identifies required locations, objects, and actions, and validates that available 3D environments (Episodes) can satisfy all requirements through a set-cover algorithm with backtracking. For multi-location stories, a MetaEpisode wrapper aggregates environments and automatically generates cross-episode movement actions. A valid group of episodes is randomly chosen (ensuring multiple possible simulations for the same GEST).

Stage 2: Object grounding. Abstract graph entities are mapped to concrete 3D objects via a chain-ID system that ensures consistent object identity across actions while preventing multi-actor conflicts on shared resources.

Stage 3: Temporal orchestration. The ActionsOrchestrator coordinates multi-actor execution. An EventPlanner constructs a happens-before graph from temporal edges and applies *Floyd-Warshall transitive closure* to derive a total ordering, then partitions events into temporal segments. Events within a segment execute concurrently; segments execute sequentially. This supports three synchronization patterns: sequential ($e_1 \rightarrow e_2$), parallel ($e_1 \leftrightarrow e_2$ via `same_time`), and interleaved ($e_1 \parallel e_2$ via `concurrent`).

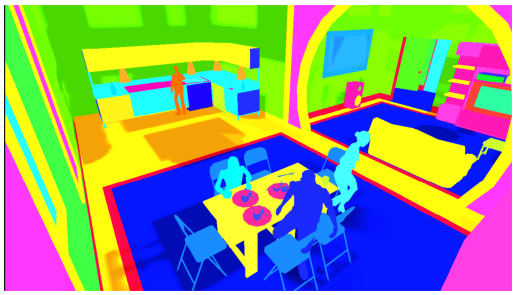
Stage 4: Execution and capture. Actions are dispatched to actor handlers in the 3D environment. A camera manager tracks the actions and automatically switches focus between actors. Simultaneously, the artifact collection system (Section 4) captures frame-aligned multi-modal annotations.

4 MULTI-MODAL ARTIFACT COLLECTION

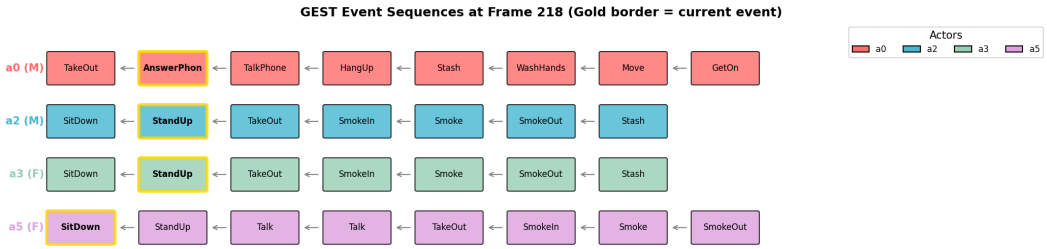
GEST-Engine's artifact collection system captured frames are aligned to the structured event graph that generated them, linking visual observations to their causal and temporal context. An *ArtifactCollectionManager* implements a freeze/collect/unfreeze state machine ensuring frame-consistent multi-modal capture. Beyond standard modalities (RGB via Desktop Duplication API, texture segmentation via HLSL shader with FNV-1a texture hashing), the system produces two additional annotation types: **frame-level spatial relations** and **event temporal alignment**. The system is open for extension to collect additional modalities like depth, actor pose, and optical flow.

Per-frame spatial relation graphs. At each captured frame, the collector queries the 3D positions and rotations of every entity (actors and objects) along with the camera state (position, lookAt, FOV, roll). For each entity, it computes camera-relative spatial data: Euclidean distance, horizontal and vertical angles, a coarse direction bucket (front/back/left/right/above/below), and an in-FOV flag (with the option to only collect objects in the FOV). Crucially, it also computes *pairwise* relations

(a) RGB with spatial annotations



(b) Instance segmentation



(c) GEST event sequences (gold border = current event)

Figure 1: Multi-modal outputs: (a) RGB with active events and camera-relative directions; (b) instance segmentation via HLSL texture hashing; (c) GEST event sequences across 4 actors.

between all entity pairs—not just entity-to-camera—producing a complete per-frame spatial relation graph. Each entity is tagged with its story-level ID from the input GEST, maintaining the link between the symbolic specification and the visual observation across the entire video. Content-hash deduplication avoids redundant writes for static scenes.

Event-frame temporal alignment. An EventBus-driven collector subscribes to `graph_event_start` and `graph_event_end` signals emitted by the orchestration engine, recording the exact frame at which each GEST event begins and ends: $\{e_i \mapsto [\text{startFrame}_i, \text{endFrame}_i]\}$. This produces a complete temporal bridge between the input graph and the output video—every frame can be mapped back to which graph events are active, and every event can be located to its precise frame range.

5 Generated Corpus and Applications

A procedural Python generator creates GEST specifications with configurable complexity (actor count, actions per location, constraint density, object reuse). The generator queries the engine's capability registry—episode definitions, available POIs, action chains, and object capacities—to ensure every generated graph is executable without conflicts. An automated batch pipeline orchestrates multiple virtual machines running independent engine instances, with centralized health monitoring, automatic failure recovery, artifact validation, and cloud upload—enabling unattended corpus-scale production. Table 1 summarizes the current corpus; the batch pipeline supports arbitrary scaling beyond this initial sample.

Initial human evaluation. Following Masala et al. (2023), we collected annotations from independent human evaluators comparing engine-generated videos against videos from VEO 3.1 (Google, 2025) and WAN 2.2 (Wan et al., 2025) prompted with the engine's textual descriptions. GEST-Engine reaches 75.9% physical validity and 4.09/5 semantic alignment, versus 26.0%/2.55 (VEO) and 20.1%/1.80 (WAN).

The generated corpus enables several downstream applications for world model research:

Temporal reasoning evaluation. The event-frame alignment preserves the full GEST constraint graph—co-occurrence (`same_time`) and causal ordering (`before`/`after`)—enabling evaluation that tests *genuine* temporal reasoning rather than single-frame recognition (Cai et al., 2024;

Table 1: Corpus statistics from 398 procedurally generated multi-actor stories.

Metric	Value
Videos generated	398
Total events	11,627
Total temporal relations	4,603 (43.5% before, 43.5% after, 13% same_time)
Unique action types	37 (social, manipulation, locomotion, exercise)
Object types	15 (furniture, devices, consumables, equipment)
Environments	11 (house×3, garden, classroom, gym×3, office x 2, common)
Actors per video	2–6 (mean 3.38)
Events per video	10–65 (mean 29.21)

Mangalam et al., 2023). In contrast to Action100M's (Chen et al., 2026) LLM-generated temporal segments (which potentially approximate boundaries), GEST-Engine's mappings are exact by construction.

Dynamic scene graph generation. The per-frame spatial relation graphs constitute ground-truth dynamic scene graphs—a notoriously expensive annotation to obtain from real video. Action Genome (Ji et al., 2020) provides sparse annotations (5 sampled frames per action interval) with documented noise; GEST-Engine produces *dense*, per-frame spatial relations with consistent entity identifiers at zero cost.

Video generation evaluation and controlled variation. GEST specifications serve as structured test cases for neural video generators, with per-frame annotations as automated metrics. A single GEST can be re-executed across different episodes sharing the same semantic locations (e.g., different kitchens, living rooms), producing visually distinct videos with identical event structure for controlled robustness studies.

6 Limitations and Future Work

Visual domain gap. The current backend (GTA San Andreas, 2004) introduces a visual gap with modern expectations, acceptable for structural reasoning where spatial layout matters more than texture quality. GTA V via FiveM is the immediate next target for photorealistic output (minimal estimated effort), with broader platform support (Unreal, Unity) enabled by the same adapter pattern.

Action vocabulary. The 37 action types and 10 environments cover household, gym, outdoor, and social activities; In the future, we plan to map more environments and actions and to further increase the video narrative complexity with LLM-powered GEST generation techniques similar to Lin et al. (2023).

Technical challenges. Outdoor scenes contain artifacts in the texture segmentations due to overlapping lightning textures. This will be fixed as development continues.

Benchmarks and evaluation. Leveraging artifacts produced with our system, we plan to create a benchmark for aberrations appearing in videos generated with the latest neural models (objects appearing, morphing into other objects), and to pretrain video understanding models on this data for real-world transfer evaluation.

7 Conclusion

We presented GEST-Engine, a system for generating multi-actor videos with perfect spatiotemporal annotations through game engine simulation. By executing formal GEST specifications, the system guarantees object permanence, causal ordering, and multi-actor coordination while providing frame-aligned segmentation, spatial relations, and event-frame mappings at zero marginal cost. Explicit world models (graph-based specifications) and implicit world models (neural video generators) are complementary (Ding et al., 2025; LeCun, 2022): GEST-Engine generates the structured training data that implicit models require.

REPRODUCIBILITY STATEMENT

The GEST-Engine source code is publicly available at `https://github.com/ncudlenco/mta-sim/releases/tag/v1.0-iclr2026`. The procedural GEST generator and batch production orchestrator are available at `https://github.com/ncudlenco/multiagent_story_system/releases/tag/v1.0-iclr2026`. The sample corpus of 398 multi-actor videos described in this paper is available at `https://huggingface.co/datasets/nnc-001/gtasa-01`. Both code repositories are tagged at the exact state used to generate the results reported in this paper.

ACKNOWLEDGMENTS

This work was supported by Büchi Labortechnik AG and by the project "Romanian Hub for Artificial Intelligence — HRIA", Smart Growth, Digitization and Financial Instruments Program, 2021–2027, MySMIS no. 351416.

REFERENCES

James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26 (11):832–843, 1983.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

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. 2025.

Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8726–8737, 2023.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL `https://openai.com/research/video-generation-models-as-world-simulators`.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, et al. Temporalbench: Benchmarking fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024.

Delong Chen, Tejaswi Kasarla, Yejin Bang, Mustafa Shukor, Willy Chung, Jade Yu, Allen Bolourchi, Theo Moutakanni, and Pascale Fung. Action100m: A large-scale video action dataset. *arXiv preprint arXiv:2601.10592*, 2026.

Jingtao Ding, Yunke Zhang, Yu Shang, Yuheng Zhang, Zefang Zong, Jie Feng, Yuan Yuan, Hongyuan Su, Nian Li, Nicholas Sukiennik, et al. Understanding world or predicting future? a comprehensive survey of world models. *ACM Computing Surveys*, 58(3):1–38, 2025.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017.

Google. Veo 3 model card. 2025. URL `https://storage.googleapis.com/deepmind-media/Model-Cards/Veo-3-Model-Card.pdf`. Accessed: March 04, 2026.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3), 2018.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10236–10247, 2020.

Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning. *arXiv preprint arXiv:2309.15091*, 2023.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Mihai Masala, Nicolae Cudlenco, Traian Rebedea, and Marius Leordeanu. Explaining vision and language through graphs of events in space and time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2826–2831, 2023.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8494–8502, 2018.

Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pp. 102–118. Springer, 2016.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

LLM USAGE

Claude Opus 4.5 was used for research, summarization, and paraphrasing. Everything was reviewed by the authors and complies with the ICLR LLM usage policy. Claude Code was used to assist with code generation.

APPENDIX A

Table 2 lists the complete set of 46 granular actions available to actors in the simulation environment, organized by semantic category.

Table 2: Complete action vocabulary organized by semantic category.

Category	Action	Description
Movement	`Move`	Navigate to a target location
	`GetOn`	Mount an object (e.g., treadmill, bike, bench)
	`GetOff`	Dismount from an object
	`SitDown`	Sit on a chair or surface
	`StandUp`	Stand up from a seated position
Social	`Talk`	Engage in conversation with another agent
	`TalkPhone`	Talk on a mobile phone
	`AnswerPhone`	Answer an incoming phone call
	`HangUp`	End a phone call
	`HandShake`	Shake hands with another agent
	`Hug`	Hug another agent
	`Kiss`	Kiss another agent
	`Laugh`	Laugh during interaction
	`Wave`	Wave at another agent
	`LookAt`	Direct gaze toward a target
Object Interaction	`PickUp`	Pick up an object from a surface or floor
	`PutDown`	Place a held object down
	`TakeOut`	Take an object out of a container
	`Give`	Hand an object to another agent
	`Receive`	Receive an object from another agent
	`Stash`	Store an object in inventory
	`TurnOn`	Turn on a device
	`TurnOff`	Turn off a device
	`PunchDesk`	Punch a desk
	`LayOnElbow`	Lay down resting on elbow while seated at a table
Consumption	`Eat`	Consume food
	`Drink`	Consume a beverage
	`Cook`	Prepare food
	`Smoke`	Smoke a cigarette
	`SmokeIn`	Retrieve a cigarette and start smoking
	`SmokeOut`	Throw away the cigarette and stop smoking
Physical Activity	`BarbellWorkOut`	Exercise with a barbell
	`BenchpressWorkOut`	Perform bench press exercise
	`DumbbellsWorkOut`	Exercise with dumbbells
	`JogTreadmill`	Jog on a treadmill
	`PedalGymBike`	Pedal a stationary gym bike
	`TaiChi`	Perform tai chi exercises
	`Punch`	Punch a punching bag
	`Dance`	Perform a dance animation
Device	`OpenLaptop`	Open a laptop / starts a computer
	`CloseLaptop`	Close a laptop / shuts down computer
	`TypeOnKeyboard`	Type on a keyboard
Personal	`WashHands`	Wash hands at a sink
	`Sleep`	Sleep on a bed
	`LookAtTheWatch`	Check the time on a wristwatch
	`Wait`	Idle in place

APPENDIX B

Table 3 summarizes the six distinct environment types used in the simulation. Each environment type corresponds to a different interior with different objects, rooms, and locations.

Table 3: Environment types and their characteristics.

Environment Type	**Setting**	**Variations**	**Available Objects**
House	Interior	3	Living room (sofas, music player, table), kitchen (counter, sink, table), bedroom (bed), bathroom (sink), hallway. Supports sitting, eating, drinking, dancing, appliance interaction, sleeping, smoking, phone use, laptop use, and social interactions.
Classroom	Interior	1	Desks with chairs arranged in rows, whiteboard area. Supports sitting, reading, typing, drinking, eating, smoking, phone use, and social interactions.
Office	Interior	2	Supports smoking, phone use, and social interactions.
Gym	Interior	3	Treadmills, stationary bikes, bench presses, punching bags, dumbbells, yoga/tai chi area. Supports all physical activity actions, social interactions, smoking, and phone use.
Garden	Exterior	1	Porch, garden area, driveway, street. Supports tai chi, smoking, phone use, and social interactions.
Common Area	Interior	1	Shared living room across interiors. Supports smoking, phone use, and social interactions.