

ROMANIAN ACADEMY

School of Advanced Studies of the Romanian Academy

„Simion Stoilow" Institute of Mathematics

PhD Thesis Summary

Understanding Visual Understanding: From Brain Signals to Explainable Multimodal Representations

THESIS ADVISOR:
Prof. Dr. Marius Leordeanu

PHD STUDENT:
Eng. Nicolae Cudlenco

Bucharest, 2026

Contents

Abstract **3**

1 Introduction **4**

2 Preliminary Work: Brain Signals and Visual Understanding **7**
2.1 Dataset 7
2.2 Method 8
2.3 Experiments and Results 8
2.4 Discussion and Implications 9

3 The Graph of Events in Space and Time **10**
3.1 The GEST Model 10
3.2 The Vision: GEST as a Multimodal Hub 10
3.3 Text to GEST 11
3.4 Initial Validation 12
3.4.1 Graph Matching for Text Similarity 12
3.4.2 Human Evaluation of Generated Videos 13
3.5 Discussion 13

4 The GEST-Engine: From Event Graphs to Synthetic Video **14**
4.1 Overview and Evolution 14
4.2 The World Editor 15
4.3 Architecture Overview 17
4.4 Multi-Modal Artifact Collection 18
4.5 Procedural GEST Generation 19
4.6 Text Generation Pipeline 19
4.7 Production Orchestration 19

5 Agentic Text-to-GEST Generation **20**
5.1 Introduction 20
5.2 The Staged LLM Approach 20
5.3 Agentic System with Function Calling 21

5.4 Evaluation 22
5.5 The Agentic System as the Full Realization of GEST 23

6 Experimental Evaluation **24**
6.1 The GTASA Dataset 24
6.2 Human Evaluation of Video Generation 25
6.3 Video Captioning 26
6.4 Spatiotemporal Probing 26
6.5 Case Study: VLM Weak Points 28
6.6 Discussion 28

7 Conclusions and Future Work **29**
7.1 Summary of Contributions 29
7.2 Future Work 30
7.3 Closing Remarks 30

Bibliography **32**

Abstract

This thesis investigates visual understanding from two complementary perspectives — how the human brain processes visual information, and how structured representations can bridge the gap between vision and language for machines — culminating in a system that generates controllable multi-actor narrative videos with perfect spatiotemporal annotations.

An empirical study of brain signals during visual recognition reveals that frontal electrodes, associated with higher-level cognitive processing, contribute more to classification accuracy than occipital electrodes over the primary visual cortex, and that EEG features complement deep visual features (improving recognition from 91% to over 97%). Motivated by this finding, we develop the Graph of Events in Space and Time (GEST), a formal representation that decomposes visual scenes into events connected by temporal, spatial, and semantic relationships, validated as both a semantic similarity measure and a video generation specification.

We develop the GEST-Engine, a pipeline that executes GESTs deterministically in a 3D game engine, producing multi-actor videos with dense frame-level annotations — RGB, instance segmentation, per-frame pairwise spatial relation graphs, event-to-frame temporal mappings, and natural language descriptions — at zero marginal cost. To bridge narrative coherence with simulator validity, we develop an agentic text-to-GEST system where an LLM plans stories through natural language while a programmatic state backend enforces constraints through validated tool calls, achieving 80% end-to-end success versus 0% for a staged LLM pipeline.

Using the engine, we produce the GTASA corpus: 938 multi-actor videos with 892 million pairwise spatial relations across 1.45 million frames. A human study (1,020 annotations, 16 evaluators) shows 69% physical validity and 4.09/5 semantic alignment versus 18%/2.50 (VEO 3.1) and 13%/1.75 (WAN 2.2). Fine-tuning video captioning models on GEST data outperforms both VEO-generated and real-only training. Probing four frozen video encoders reveals that self-supervised encoders encode spatial structure significantly better than VLM visual encoders, and that frontier VLMs perform near chance (56%) at detecting physically invalid videos.

Keywords – *visual understanding, event graphs, video generation, spatiotemporal annotations, multi-actor simulation, game engines, agentic AI, synthetic data*

Chapter 1

Introduction

This thesis investigates visual understanding from two complementary perspectives — how the human brain processes visual information, and how structured representations can bridge the gap between vision and language for machines.

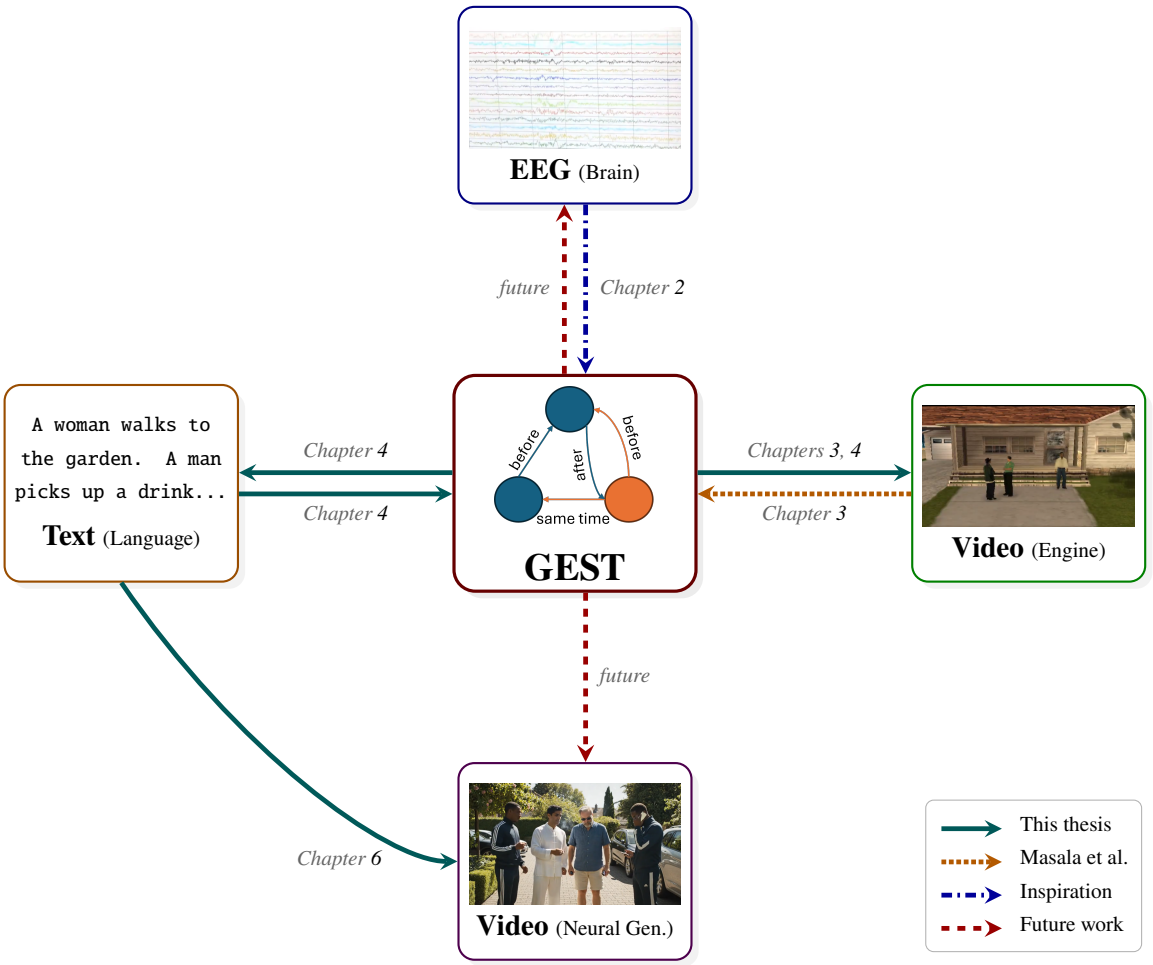


Figure 1.1: **Thesis overview.** GEST serves as the central representation connecting brain signals, language, and video. Solid arrows: contributions of this thesis. Dash-dot arrow: EEG-based inspiration (Chapter 2). Dotted: complementary work by Masala et al. [22]. Dashed: future research directions.

The central argument is that visual understanding benefits from explicit, structured representations that make the relationships between entities, events, and modalities transparent and computable. Rather than relying solely on implicit statistical patterns learned from raw data, we propose that grounding visual content in a formal graph-based structure — where events are nodes, and temporal, spatial, and semantic relationships are edges — enables more faithful video generation, more effective model training, and deeper insight into what existing systems actually learn about the visual world.

The thesis is organized around five research questions:

RQ1: What can brain signals reveal about the nature of visual understanding?

RQ2: Can an explicit graph-based representation bridge the gap between visual and linguistic modalities?

RQ3: Can a game-engine-based system produce semantically faithful multi-actor videos with perfect spatiotemporal annotations?

RQ4: Can LLM-based agents generate GEST graphs that are both narratively coherent and valid for the simulator?

RQ5: Can structured synthetic data improve video understanding models and reveal what they learn about space and time?

Table 1.1 maps each contribution to its publication, research question, and chapter. We summarize each below.

Table 1.1: Summary of contributions, their associated publications, research questions, and chapters.

Contribution	**Publication**	**RQ**	**Chapter**
Brain signals and visual understanding	Cudlenco et al., Neurocomputing 2020 [10]	RQ1	Ch. 2
The GEST representation	Masala, Cudlenco et al., ICCV 2023 [21]	RQ2	Ch. 3
The GEST-Engine	Cudlenco et al., published at the ICLR 2026 Workshop on World Models [7]; submitted to ECCV 2026 (arXiv:2604.10385) [9]	RQ3	Ch. 4
Agentic text-to-GEST generation	Cudlenco et al., submitted to CVPR Workshop 2026 (arXiv:2604.10383) [8]	RQ4	Ch. 5
Experimental validation	Cudlenco et al., submitted to ECCV 2026 (arXiv:2604.10385) [9]	RQ5	Ch. 6

1. **Brain signals and visual understanding** [10] (Chapter 2): an EEG study revealing that frontal electrodes associated with higher cognitive processing contribute more to visual classification than occipital electrodes over the primary visual cortex, and that EEG features complement deep visual features (91% → 97%).

2. **The GEST representation** [21] (Chapter 3): the Graph of Events in Space and Time, validated as a semantic similarity measure (outperforming BLEU, METEOR, ROUGE, SPICE) and as a video generation specification (87% human preference against neural baselines).

3. **The GEST-Engine** [7, 9] (Chapter 4): a complete system for generating multi-actor videos with dense frame-level annotations — RGB, instance segmentation, per-frame spatial relation graphs, event-to-frame temporal mappings, and natural language descriptions — at zero marginal cost. The system comprises approximately 100,000 lines of code across Lua, C++, Python, and JavaScript.

4. **Agentic text-to-GEST generation** [8] (Chapter 5): an LLM-based system achieving 80% end-to-end success versus 0% for staged LLM pipelines, with agentic narratives winning 79% of text and 74% of video comparisons against procedural baselines.

5. **Experimental validation** [9] (Chapter 6): the GTASA corpus (938 videos, 892M spatial relations, 1.45M frames) with three evaluation lines — human evaluation (69% validity vs 18%/13% for VEO/WAN), video captioning pretraining (GEST data outperforms VEO and real-only), and spatial probing of video encoders (self-supervised > VLM visual encoders).

Chapter 2

Preliminary Work: Brain Signals and Visual Understanding

Evidence from cognitive neuroscience shows that visual understanding is not confined to the primary visual cortex but engages higher cognitive regions distributed across the brain [16, 24]. Before pursuing this direction computationally — which leads to the Graph of Events in Space and Time (GEST) in Chapter 3 — we first sought to explore this question empirically, using the most direct means available: recording brain signals from people as they look at images.

2.1 Dataset

We collected a novel EEG dataset specifically designed for this study. The dataset comprises recordings from six volunteers (four male, two female), all university graduates aged 25–35. Each subject participated in two recording sessions: one for training data and one for testing data, separated by a break of several hours during which the subjects removed the headset and went about their normal activities. This separation is the defining feature of our experimental protocol — it was designed to ensure that any classifier trained on the data could not rely on session-specific noise patterns, temporal autocorrelations, or the subject's momentary cognitive state to achieve high accuracy.

The stimulus set consists of six visual categories, deliberately chosen to include both objects and natural scenes: flowers, airplanes, cars, park, seaside, and old town. Each class contains 60 images, split into 40 for training and 20 for testing, yielding a total of 360 images. Images were presented sequentially at a rate of one image every 7 seconds. During the training session, subjects were instructed to focus on each image and to actively think about the concept rather than passively viewing the stimulus.

[0]This chapter is based on: **Cudlenco**, Popescu, Leordeanu — Neurocomputing 2020 [10].

Table 2.1: Electrode pairs ranked by classification accuracy. Table from [10].

Rank	Channels	Acc (%)	Brain areas
1	FC5/FC6	66.21	Frontal-central lobe
2	F7/F8	64.14	Frontal lobe
3	AF3/AF4	63.02	Prefrontal lobe
4	P7/P8	62.94	Parietal lobe
5	T7/T8	61.69	Temporal lobe
6	F3/F4	59.66	Frontal lobe
7	O1/O2	59.25	Occipital — Visual cortex

2.2 Method

Raw EEG signals were transformed into frequency-domain features via Gabor wavelet filtering across nine scales applied to all 14 channels. We evaluated three classifiers: Ridge Regression, a CNN, and stacked LSTMs. For the fusion experiment, EEG features (from a two-layer LSTM) were concatenated with visual features from an Inception-v3 network trained on the stimulus images and passed through a joint classifier.

2.3 Experiments and Results

Electrode ablation. A systematic evaluation of each electrode pair (Table 2.1) revealed that the pair with the highest standalone classification accuracy was FC5/FC6, located over the frontal-central lobe — a region associated with higher cognitive processing, not visual perception (Figure 2.1). The top three pairs (FC5/FC6, F7/F8, AF3/AF4) are all frontal or prefrontal. The occipital pair O1/O2, positioned directly over the primary visual cortex, ranked last. Using only the top-3 frontal pairs yielded 69.39% accuracy versus 66.76% when using all channels. This pattern was remarkably consistent across all six subjects, strengthening the conclusion that the effect is not an artifact of individual variation but reflects a genuine property of how the brain processes visual categories.

Model comparison. Ridge Regression with Gabor filtering (29.2%) outperformed the CNN (26.5%) and LSTM (25.67%) on the multiclass task, consistent with the limited data regime where simpler models generalize more reliably. All models exceeded the 16.67% random baseline.

EEG–visual fusion. Combining EEG features with deep visual features improved classification from 91% to over 97% averaged across subjects. Every subject showed improvement (4–6 percentage points), confirming that brain signals carry complementary information not captured by image-based classifiers alone.

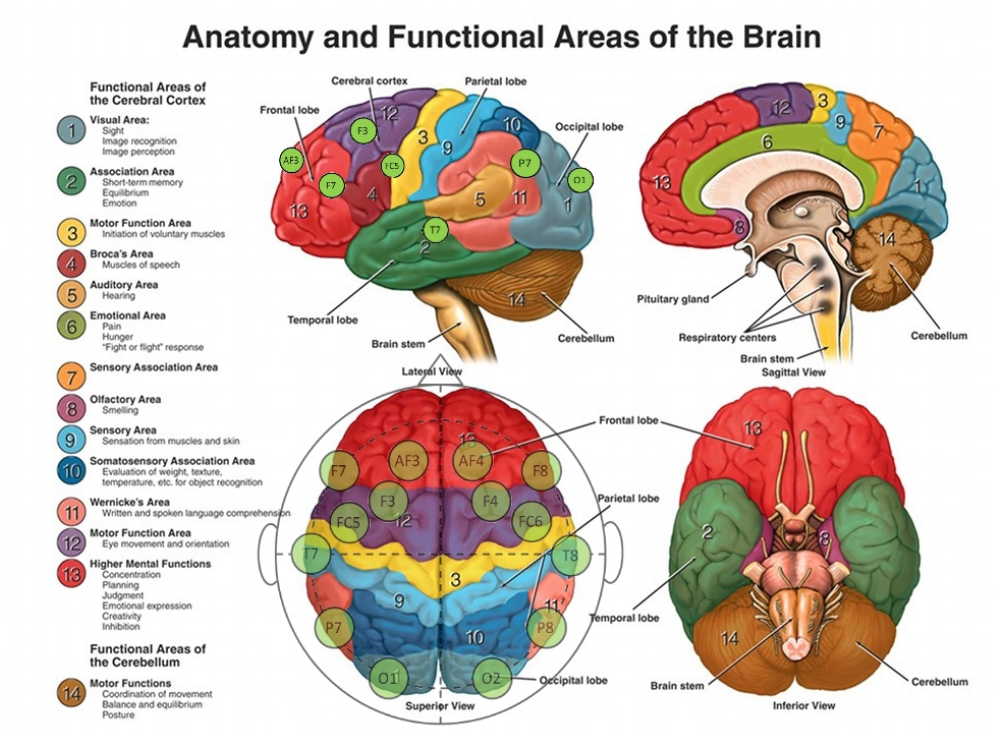


Figure 2.1: **Electrode locations relative to brain functional areas.** The most informative electrodes for visual classification (FC5/FC6, F7/F8, AF3/AF4) are located over frontal areas, not over the occipital visual cortex. Figure from [10].

2.4 Discussion and Implications

These results have a direct bearing on the central theme of this thesis. If visual understanding in the brain engages distributed cortical networks — including areas responsible for language, planning, and abstract reasoning — then computational approaches to visual understanding should aim to capture this level of structured, conceptual processing. A representation that operates only at the level of pixels and learned statistical features, no matter how powerful, would miss the kind of relational and semantic information that the brain evidently brings to bear on even a simple image classification task. This insight motivated our development of the Graph of Events in Space and Time (GEST), an explicit graph-based representation that encodes events, actors, and their temporal and spatial relationships at a conceptual level. GEST is the subject of the next chapter.

Chapter 3

The Graph of Events in Space and Time

3.1 The GEST Model

A GEST is a directed graph $G = (\mathcal{V}, \mathcal{E})$ where nodes represent events and edges encode relationships between them. Each event node E_i specifies an action, the entities involved, a location, a timeframe, and additional properties. A distinctive feature is that physical objects are also represented as events (e.g., "John's car exists"), unifying actions and entities under the same representational machinery. Special `Exists` nodes declare actors and objects, establishing the entity inventory.

Edges relate two events and can encode temporal (before, after, same_time, concurrent), spatial (near, behind, left, right), causal (because, causes), or semantic (same, co-reference) relationships. The temporal relations draw on Allen's interval algebra [1].

Perhaps the most powerful property of GEST is its ability to operate at multiple levels of abstraction through two operations: *graph-to-node collapse* (any subgraph can be compressed into a single event at a higher level) and *node-to-graph expansion* (any event can be expanded into a detailed subgraph). This hierarchical property is essential for bridging vision and language, where descriptions of the same scene vary in granularity. Figure 3.1 illustrates a GEST graph for a multi-event narrative.

3.2 The Vision: GEST as a Multimodal Hub

GEST was designed as a hub connecting multiple modalities (Figure 3.2) through four functional pathways: (A) text to GEST — constructing a graph from natural language; (B) GEST to text — generating language from a graph; (C) GEST to video — generating video from a specification; and (D) video to GEST — extracting a graph from video. Any two pathways can be chained for cross-modal conversion (e.g., text to video via GEST), with the graph serving as an explicit, inspectable intermediate stage. Unlike end-to-end

[0]This chapter is based on: Masala, **Cudlenco**, Rebedea, Leordeanu — ICCV 2023 [21].

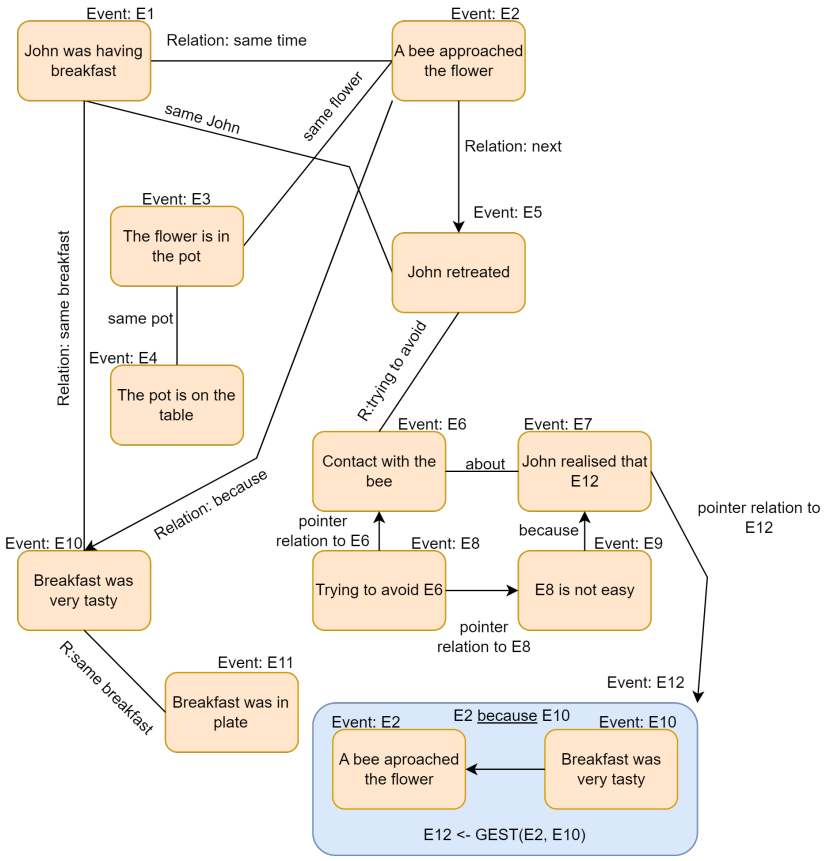


Figure 3.1: **GEST graph example.** A narrative decomposed into twelve event nodes connected by temporal, causal, spatial, and co-reference edges. Bottom: graph-to-node collapse aggregates events into a higher-level event. Figure from [21].

neural models, a GEST-mediated pipeline allows each conversion step to be examined and corrected independently. A single GEST can produce multiple distinct videos (same narrative in different environments) and multiple texts (same events at different detail levels), while multiple texts or videos may map to the same GEST if they share the same event structure — enabling semantic comparison at the level of meaning rather than surface form.

3.3 Text to GEST

Three approaches were developed for converting natural language text into GESTs: (1) rule-based parsing with human correction, using dependency parsing and semantic role labeling to extract events and relations; (2) GPT-3 finetuning on manually annotated GEST examples, producing GESTs directly from text via few-shot learning; and (3) zero-shot generation with ChatGPT using structured prompts. All three produce GESTs at the semantic level but none guarantees that the resulting graph is executable by the GEST-Engine. Bridging this gap required coupling language models with procedural constraint enforcement, as described in Chapter 5.

GEST to Text. We developed a two-stage text generation pipeline for the GEST-Engine

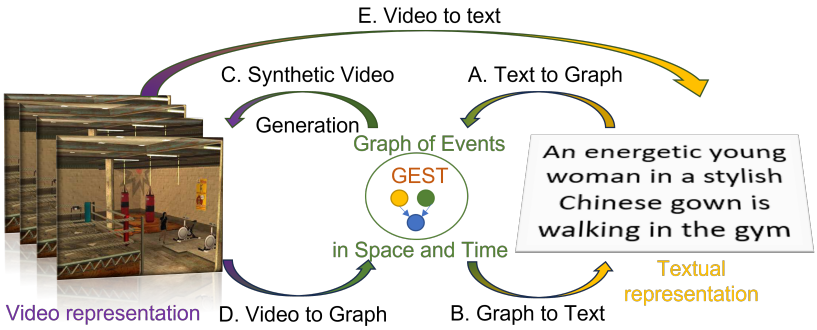


Figure 3.2: **Functional overview of the GEST framework.** GEST serves as the central component, enabling bidirectional conversion between visual and textual representations. Figure from [21].

(Chapter 4), adapting the proto-language approach of Masala et al. [22] — which algorithmically generates structured textual descriptions from GEST graphs by sorting and grouping events in space and time — to incorporate the rooms where actions occur and the actors' genders, and coupling it with GPT-4o refinement to produce natural, fluent descriptions from the engine's output.

Video to GEST. Complementary to our generation direction, Masala et al. [22] developed an approach to extract GEST graphs from real videos, combining semantic segmentation, action detection, and object tracking. This pathway closes the loop: videos can be converted back into structured event representations, enabling video-to-video comparison at the semantic level.

GEST to Video. We developed a video generation engine capable of taking a GEST graph as input and producing a corresponding multi-actor video in a 3D simulation environment. The full system is described in Chapter 4.

3.4 Initial Validation

3.4.1 Graph Matching for Text Similarity

We evaluated whether comparing texts in GEST space captures semantic similarity more effectively than standard metrics. Both texts were converted to GESTs and compared using Spectral Matching (SM) [20] and Neural Graph Matching (NGM) [30] with GloVe-based affinity matrices.

Table 3.1 shows that GEST-SM achieves the highest correlation (61.70%) and Fisher score (1.20), while GEST-NGM achieves the highest accuracy (86.31%). Combining BLEURT with GEST-SM yields the best performance across all metrics (Table 3.2), confirming that GEST captures a dimension of semantic similarity complementary to neural text metrics.

Table 3.1: GEST vs. text similarity metrics on Videos-to-Paragraphs. Table from [21].

Method	**Corr(%)**	**Acc(%)**	**F**	**AUC(%)**
BLEU@4	24.45	75.52	0.28	52.65
METEOR	58.48	84.23	1.12	73.90
ROUGE	51.11	83.40	0.72	68.92
SPICE	59.42	84.65	1.04	74.43
BERTScore	57.39	85.89	1.07	**77.93**
GEST-SM	**61.70**	84.65	**1.20**	75.47
GEST-NGM	60.93	**86.31**	0.98	76.75

Table 3.2: BLEURT combined with other metrics. GEST-SM provides the largest complementary gain. Table from [21].

Method	**Corr(%)**	**Acc(%)**	**F**	**AUC(%)**
BLEURT	70.93	90.04	2.03	88.02
+GEST-SM	**72.89**	**90.87**	**2.21**	**89.80**
+GEST-NGM	71.91	90.46	2.05	88.58
+SPICE	71.94	88.80	2.09	87.71
+METEOR	71.20	89.63	2.07	87.62

3.4.2 Human Evaluation of Generated Videos

We compared GEST-generated videos against CogVideo [15] and Text2VideoZero [19]. Six independent annotators evaluated 111 video–text pairs. In 87% of cases, annotators selected the GEST-generated video as the best semantic match (average score 7.93/10 vs 5.55 for Text2VideoZero and 4.03 for CogVideo). An automatic evaluation using VALOR [6] confirmed the same ranking: our videos achieved the best score on six of seven standard captioning metrics.

3.5 Discussion

These results establish GEST as a viable representational framework for connecting vision and language at the level of semantic content. The representation proved expressive enough to capture complex multi-actor narratives with temporal ordering, spatial context, and causal relationships, and structured enough to enable both graph-based comparison and deterministic video generation.

The video generation engine used for the experiments in this chapter supported only sequential execution without formal temporal constraints and did not produce dense annotations. These limitations were addressed in the full GEST-Engine described in the next chapter.

Chapter 4

The GEST-Engine: From Event Graphs to Synthetic Video

The idea of using a graph of events in space and time as a specification for video generation originated with the observation that if we could programmatically control every entity in a 3D world — spawning actors, placing objects, triggering animations, and recording the results — we could produce training data at a scale and with an annotation density that no manual labeling effort could match. The graph structure described in Chapter 3 provided the formal framework; what remained was to build a system that could take such a graph as input and realize it as a video.

4.1 Overview and Evolution

The GEST-Engine operates on Grand Theft Auto: San Andreas (2004) via Multi Theft Auto (MTA), an open-source multiplayer modification framework that exposes the game engine through a Lua scripting interface. We chose this platform for its open-source nature (GPLv3), minimal hardware requirements (enabling parallel execution across multiple VMs on commodity hardware), and rich pre-existing assets: 733 object models across 32 categories, over 2,500 animations across 101 categories, 312 actor skins, and more than 70 distinct environments.

The GEST-Engine represents a substantial engineering effort. The system comprises approximately 38,000 lines of Lua for the engine core, 8,500 lines of C++ for the native screenshot module and pathfinding, 53,000 lines of Python for the procedural generator and the agentic text-to-GEST system (Chapter 5), and 600 lines of JavaScript for the interactive GEST visualizer — totalling approximately 100,000 lines of code across all components. This entire codebase was developed by a single researcher over the course of the doctoral program. Our choice to build on an existing game rather than creating a virtual world

[0]This chapter is based on: **Cudlenco**, Masala, Leordeanu — published at the ICLR 2026 Workshop on World Models [7]; and **Cudlenco**, Masala, Leordeanu — submitted to ECCV 2026 (arXiv:2604.10385) [9].



Figure 4.1: **The World Editor in use.** A living room (left) and gym (right) episode showing POIs (cylinders), region vertices, camera positions, and action lists. Green POIs have actions defined; red POIs have none.

from scratch proved critical to making this feasible: by leveraging the assets, rendering, physics, and animation systems already present in GTA San Andreas, we could focus our engineering effort entirely on the graph orchestration, temporal coordination, and annotation collection layers. For comparison, VirtualHome [25] — the most closely related system — was developed by a team of seven researchers with nine contributors to the public codebase over a comparable timespan (2018–2024, based on public commit history), and despite this larger team, it remains limited to single-actor sequential programs in closed indoor environments without temporal orchestration or dense spatial annotations. The source code is publicly available[1,2].

4.2 The World Editor

One limitation of GTA San Andreas via MTA is that while we have programmatic access to the game's objects, animations, and characters, we do not have access to the geometry of the environment — there is no API to query the layout of rooms, the positions of furniture, or the walkable paths through a building. To address this, we developed a World Editor: an in-game tool that allows us to walk through any environment in the game and interactively define all the spatial and semantic information needed for simulation.

The World Editor defines the following elements, serialized to JSON files loaded at runtime: *Regions* (polygon boundaries representing spatial areas such as bedrooms, kitchens, hallways), *Points of Interest* (specific locations within regions where actions can be performed, each with defined action chains and object types), *Templates and Supertemplates* (reusable configurations of objects at POIs, with supertemplates providing random selection among template variants for visual diversity), *Cameras* (positioned within regions for automated cinematography), and *Pathfinding graphs* (navigation meshes enabling actors to walk between regions and POIs while avoiding obstacles).

[1] `https://github.com/ncudlenco/mta-sim`
[2] `https://github.com/ncudlenco/multiagent_story_system`

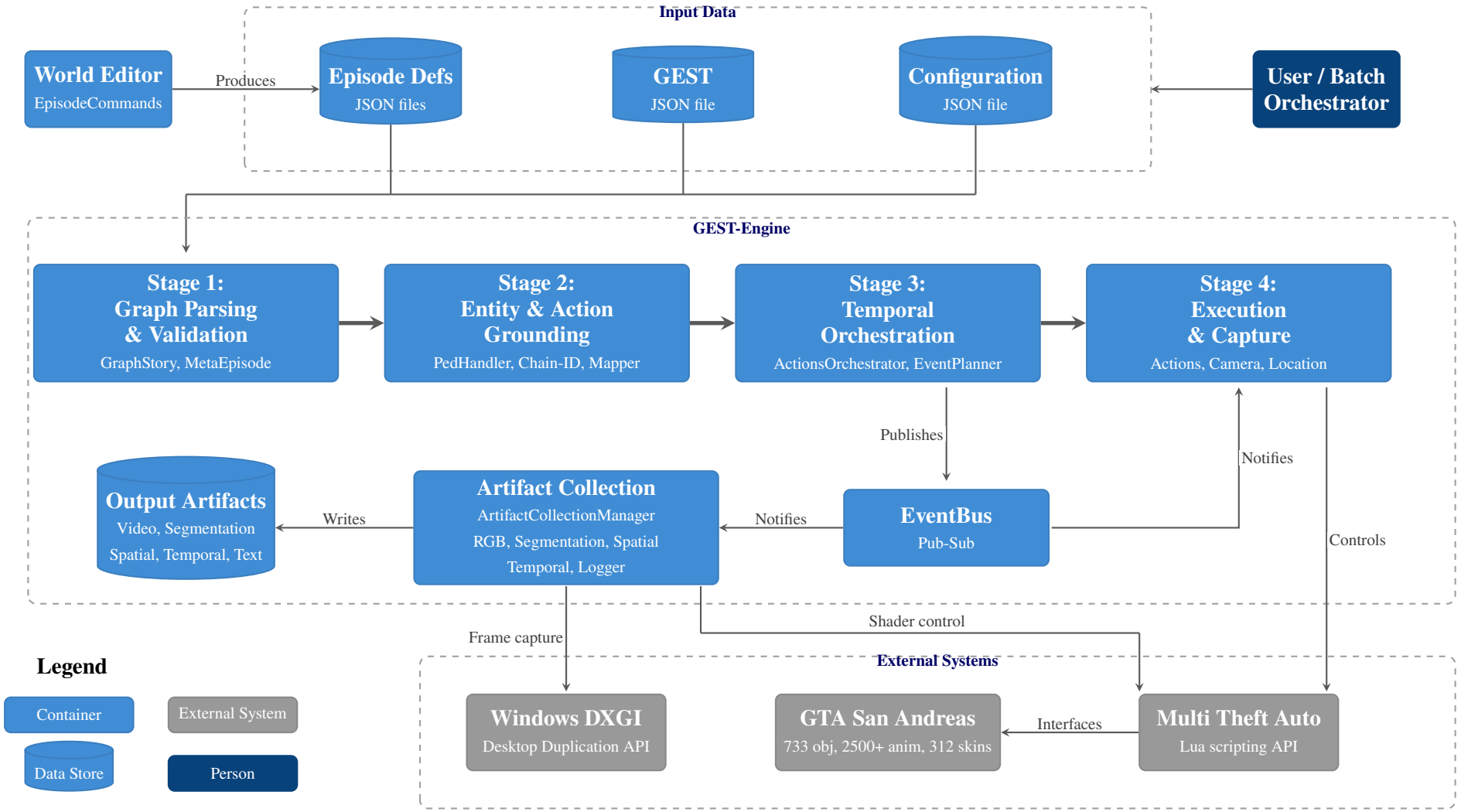


Figure 4.2: **System architecture of the GEST-Engine.** Input data feeds into the four-stage pipeline: **Stage 1** (Section 4.3) parses and validates the GEST, **Stage 2** (Section 4.3) grounds actors, objects, and actions to the 3D world, **Stage 3** (Section 4.3) orchestrates multi-actor execution and publishes events, **Stage 4** (Section 12) executes actions via Multi Theft Auto. The EventBus notifies both Stage 4 and the Artifact Collection system (Section 4.4), which captures dense multi-modal annotations.

Algorithm 1: Temporal segment computation via dependency-first scheduling.

Input: GEST events $\mathcal{V}$, temporal constraints C
Output: Temporal segments $\{S_1, S_2, \ldots, S_k\}$ partitioning $\mathcal{V}$

$H \leftarrow$ BuildHappensBeforeGraph(C);
$H^* \leftarrow$ FloydWarshallClosure(H);
satisfied $\leftarrow \emptyset$; $k \leftarrow 0$;
while $|satisfied| < |\mathcal{V}|$ **do**
    ready $\leftarrow \{e \in \mathcal{V} \setminus \text{satisfied} \mid \text{IsReady}(e, H^*, \text{satisfied})\}$;
    if *ready* $= \emptyset$ **then**
        error "Deadlock: circular dependency detected";
    groups $\leftarrow$ GroupByConcurrency(ready, H^*);
    foreach *group* $g \in$ *groups* **do**
        $k \leftarrow k + 1$; $S_k \leftarrow g$;
        satisfied $\leftarrow$ satisfied $\cup\, g$;
return $\{S_1, \ldots, S_k\}$;

4.3 Architecture Overview

The engine transforms a GEST specification into annotated video through four stages (Figure 4.2).

Stage 1: Graph Parsing and Validation. The input GEST is parsed, its requirements (actors, objects, actions, locations) are extracted, and a valid set of episodes is selected through a set-cover algorithm with backtracking. For multi-location stories, a MetaEpisode wrapper aggregates multiple episodes and generates cross-episode movement actions. A valid group of episodes is randomly chosen, ensuring multiple possible simulations for the same GEST.

Stage 2: Entity & Action 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. Every required object and action sequence is mapped to all valid POI chains in the selected episodes through bidirectional maps that enable deferred runtime selection among valid alternatives.

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 [13] to derive a total ordering, then partitions events into temporal segments through dependency-first scheduling (Algorithm 1). 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. The PedHandler manages actor lifecycle, 55 action classes implement the full action taxonomy, the location system manages spatial semantics and POI selection, and the

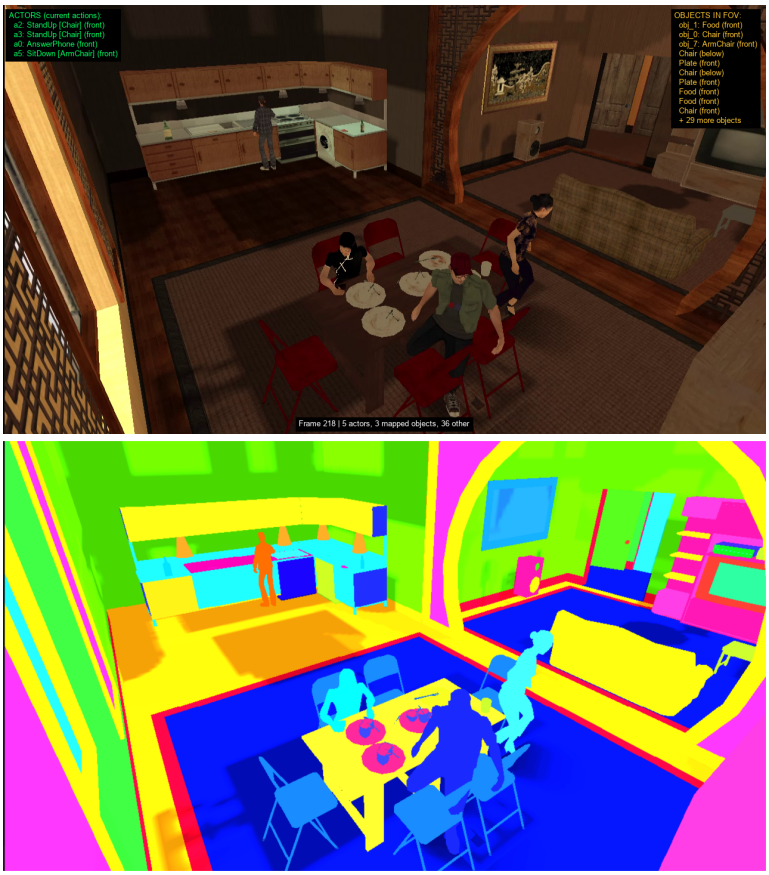


Figure 4.3: **Multi-modal outputs.** Top: RGB frame with active events and camera-relative spatial annotations. Bottom: instance segmentation via HLSL texture hashing — each unique texture maps to a distinct color. Figure from [7].

camera system provides automated cinematography with focus-based actor tracking. An EventBus decouples the orchestration from the artifact collection system through a publish-subscribe mechanism.

4.4 Multi-Modal Artifact Collection

The artifact collection system captures dense, frame-aligned annotations during simulation at zero marginal cost. An ArtifactCollectionManager coordinates capture through a freeze/collect/unfreeze state machine ensuring frame-consistent multi-modal capture. The system produces three primary artifact types (Figure 4.3):

RGB capture uses a native C++ module built on the Windows Desktop Duplication API (DXGI), acquiring frames directly from the GPU's front buffer with minimal latency. Frames are encoded to H.264 via Windows Media Foundation directly on the GPU.

Instance segmentation is produced by an HLSL pixel shader that assigns each unique texture a distinct color via FNV-1a hashing, providing texture-level segmentation finer than traditional object-level masks.

Per-frame spatial relation graphs are computed by querying the 3D position and rotation

of every entity at each frame, computing pairwise relations between all entity pairs — not just entity-to-camera — producing a complete per-frame spatial relation graph with story-level entity identifiers maintained throughout.

Event-frame temporal alignment is captured by subscribing to `graph_event_start` and `graph_event_end` signals, recording the exact frame at which each GEST event begins and ends.

4.5 Procedural GEST Generation

The procedural generator creates random but valid GESTs from the engine's capability registry. It operates through round-based action chain generation: for each region in the story's spatial sequence, every actor receives one action chain per round, built by following valid next-action sequences at available POIs. Chains are constructed in temporary buffers and only committed to the GEST when fully valid, guaranteeing executability by construction. Interactions between co-located actors (handshakes, conversations, object exchanges) are created with `same_time` constraints for synchronized execution. Temporal relations between rounds and region transitions enforce clean sequential ordering.

Beyond standalone generation, the procedural generator serves as the state-management backend for the agentic system described in Chapter 5, exposing its internal state (actors, events, constraints, object tracking, POI capacity) through a delegation interface that allows an external LLM agent to invoke the same validated operations.

4.6 Text Generation Pipeline

The engine produces natural language descriptions of the simulated stories through a two-stage pipeline. First, a proto-language generator mechanically concatenates action names with entity descriptions and temporal connectors, producing intentionally ungrammatical but informationally complete output (e.g., `the first woman sitdowns a chair. Meanwhile, the man pickups a food.`). Second, GPT-4o rewrites the proto-language into natural, grammatically correct text while preserving all actions and temporal relations.

4.7 Production Orchestration

The production pipeline manages batch execution across multiple parallel virtual machines. A centralized Python orchestrator manages job distribution, health monitoring, automatic crash detection and recovery, artifact validation, and cloud upload. The GTASA corpus of 938 videos was produced on a single workstation with 128 GB of RAM across 25 parallel VMs.

Chapter 5

Agentic Text-to-GEST Generation

5.1 Introduction

The text-to-GEST approaches from Chapter 3 produce GESTs at the semantic level but none guarantees that the resulting graph is executable by the GEST-Engine. A GEST that references actions, objects, or locations absent from the simulation environment, or violates temporal constraints and action chain rules, will fail during execution. The challenge is generating GESTs that are both *narratively coherent* (telling a meaningful story) and *simulator-valid* (respecting the engine's constraints). LLMs excel at the former but cannot maintain the precise state-tracking needed for the latter across the 50–200 events of a multi-actor story.

5.2 The Staged LLM Approach

We first built a staged LLM refinement pipeline: a LangGraph workflow chaining six specialized stages — *Concept* (abstract hierarchical GEST), *Casting* (skin assignment), *Episode Placement* (mapping scenes to episodes), *Setup* (off-camera preparation), *Screenplay* (translating narrative to action sequences), and *Scene Detail* (complete executable GEST) — each with structured output validation via Pydantic. The pipeline produced narratively coherent stories but across 50 generation attempts, **zero** produced an executable specification. Failure modes included hallucinated actions, invalid action chains, object lifecycle violations, temporal cycles, and context overflow. The fundamental problem: LLMs cannot maintain state coherence across hundreds of events, and post-hoc correction requires backtracking through dependent state at superlinear cost. This motivated a separation-of-concerns architecture.

[0]This chapter is based on: **Cudlenco**, Masala, Leordeanu — submitted to CVPR Workshop 2026; arXiv:2604.10383 [8].

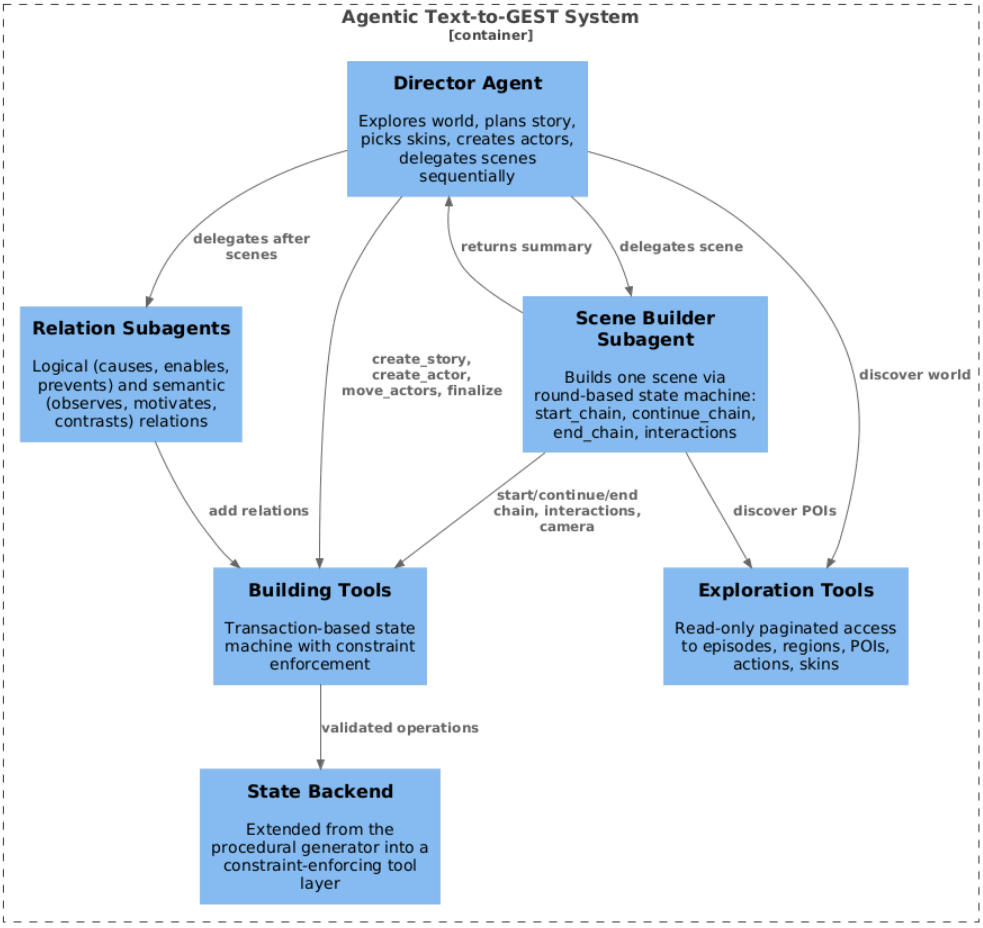


Figure 5.1: **Agentic system architecture.** The Director Agent plans the story and delegates individual scenes to the Scene Builder Subagent. Both agents interact through Exploration Tools (read-only) and Building Tools (constraint-enforcing), operating on the State Backend. Figure from [8].

5.3 Agentic System with Function Calling

The agentic system separates narrative planning from constraint enforcement. The LLM handles narrative decisions through natural language reasoning, while a programmatic state backend — the procedural generator from Chapter 4 repurposed as a constraint-enforcing tool layer — handles all state-tracking through validated tool calls. The system is built on LangGraph for workflow orchestration and deepagents for hierarchical agent delegation. Both the Director and Scene Builder use Claude Haiku[1] as the LLM backend.

The Director Agent receives a seed text and proceeds in four phases: exploration (browsing episodes, regions, POIs, and skins through paginated read-only tools), casting (creating actors with names, genders, skins, and starting regions), scene building (delegating each scene to the Scene Builder with a narrative task description), and finalization (linking scene boundaries with cross-scene temporal constraints).

The Scene Builder Subagent receives an isolated context (scene, episode, region, ac-

[1] `https://www.anthropic.com/claude/haiku`

Table 5.1: End-to-end generation success rates. Table from [8].

Approach	Attempted	Successful	Rate
Staged LLM	50	0	0%
Procedural random	34	25	74%
Hybrid agentic (seeded)	25	20	80%
Hybrid agentic (autonomous)	~20	~10	~50%

tors, narrative description) and constructs events through a round-based state machine. Each round: `start_round` returns current actor states; for each actor, `start_chain` returns valid next actions, followed by `continue_chain` calls, and `end_chain` to commit; optional `do_interaction` creates synchronized two-actor events; `end_round` adds cross-actor temporal ordering. The Scene Builder cannot create actors, move actors between regions, or finalize the GEST — enforcing a clean separation between story-level and scene-level concerns.

Relation Subagents optionally enrich the graph with logical relations (causes, enables, prevents, requires) and semantic relations (observes, interrupts, motivates). These populate the edge types defined in the GEST formalism that procedural generation never produces.

Tool-Based Constraint Enforcement implements a transaction-based state machine with four states (`IDLE`, `STORY_CREATED`, `IN_SCENE`, `IN_ROUND`). Key mechanisms: actor state tracking (posture, held objects, location), transactional chains (temporary buffers, commit or discard), POI capacity tracking, temporal cycle detection, and interaction rules. Every GEST produced is executable by construction because invalid operations are rejected at the tool level.

5.4 Evaluation

Generation success (Table 5.1). The staged pipeline produces 0% executable output. The agentic system achieves 80% with seeded generation and approximately 50% in fully autonomous mode. Every valid GEST is structurally correct by construction — failures are LLM-side (repetitive planning loops) or engine-side, never constraint violations.

Narrative quality. A 3-model LLM jury (GPT-5.2 Pro, Gemini 3.1 Pro, Claude Opus 4.6) compared agentic and procedural stories. When comparing the Director's authored narrative against procedural Video Description from Graph (VDG) descriptions, agentic narratives win 79% of text and 74% of video comparisons, confirming genuine story structure absent from procedural generation.

Narrative fidelity. Event coverage is 1.89/5 in autonomous mode and 3.12/5 when seeded. The gap reflects platform constraints (limited cinematography, temporal nuance, atmospheric expression, and action blending) rather than architectural limitations — the agentic system transfers directly to any engine that exposes these channels.

5.5 The Agentic System as the Full Realization of GEST

Over the course of this thesis, we built each direction of the GEST framework incrementally. In Chapter 3, we developed text-to-GEST conversion through rule-based parsing, GPT-3 finetuning, and zero-shot ChatGPT generation [21] — producing narratively valid graphs, but without any guarantee that they could be executed in a simulation. We adapted the proto-language pipeline from Masala et al. [22] for the engine's text generation, adding location and gender context to enable GEST-to-text conversion for engine output. In Chapter 4, we built the GEST-Engine [7, 9] that executes GESTs deterministically in a 3D environment, and the procedural generator that produces executable graphs at scale — but only with temporal relations and without narrative coherence. Each of these components exercises part of the GEST formalism, but none exercises all of it simultaneously.

The agentic system brings them together. The Director uses the hierarchical expansion property of GEST to decompose a natural language narrative into scenes and then into granular events — the text-to-GEST direction, now with executability guarantees. The Scene Builder grounds these events in the engine's capabilities through the exploration tools and the capability registry. The state backend enforces validity through the same procedural generator that produces the GTASA corpus. The Relation Subagents populate the logical and semantic edge types defined in the GEST formalism — the same types of relations that our earlier text-to-GEST approaches captured at the narrative level but that the procedural generator never produces. The GEST-Engine then executes the fully populated graph, producing video with dense annotations. And the text generation pipeline closes the loop back to natural language.

The complementary direction is also active. The video-to-GEST extraction approach from Masala et al. [22] can recover structured event graphs from the generated videos, while the text generation pipeline converts the executed GEST into a VDG description of what visually happened. From that VDG, the reconstituted narrative approach used in our evaluation demonstrates that an LLM can infer a story-level intent from the granular execution description — closing a second loop from video back to natural language at the narrative level, not just the action level.

This is where the GEST framework comes full circle. We conceived it as a bridge between vision and language operating at the level of events in space and time. The agentic system is the orchestration layer that realizes this bridge in full: from a natural language story, through a formally specified and fully annotated event graph with temporal, logical, and semantic relations, to a deterministic multi-actor video with dense annotations, and back to natural language at multiple levels of abstraction — with every component of the GEST formalism populated and every direction active.

The next chapter investigates the practical value of the dense annotations produced at corpus scale.

Chapter 6

Experimental Evaluation

This chapter evaluates whether the resulting system and data are useful — not just as an engineering artifact, but as a research tool for the broader video understanding community. The central question is whether we can leverage the explicit world models of 3D game engines to generate semantically meaningful videos, for video understanding, and benchmarking.

6.1 The GTASA Dataset

We produced GTASA (Ground Truth Annotations for Spatiotemporal Analysis): 938 procedurally generated multi-actor videos comprising 28,200 action clips across 37 action types and 14.3 hours of footage, with 892 million pairwise spatial relations across 1.45 million frames and 29,579 exact event-to-frame temporal mappings. Each story has 7–65 events (mean 29.4) across 2–6 actors (mean 3.43), spanning 10 episodes from 4 environment categories. The corpus is publicly available[1], along with the GEST-Engine source code[2] and the orchestration system[3].

Table 6.1 compares the GTASA corpus with existing video datasets and synthetic platforms. Per clip, the GEST-Engine provides two orders of magnitude more spatial relations than Action Genome (31.6K vs 170 per clip). Using the same textual descriptions as prompts, we generated corresponding videos with VEO 3.1 [14] and WAN 2.2 [28] as neural baselines for the human evaluation and captioning experiments.

[0]This chapter is based on: **Cudlenco**, Masala, Leordeanu — submitted to ECCV 2026; arXiv:2604.10385 [9].

[1]`https://huggingface.co/datasets/nnc-001/gtasa-01`

[2]`https://github.com/ncudlenco/mta-sim`

[3]`https://github.com/ncudlenco/multiagent_story_system`

Table 6.1: Comparison with video datasets and synthetic platforms. Per clip, the GEST-Engine provides two orders of magnitude more spatial relations than Action Genome (31.6K vs 170 per clip). Table from [9].

Dataset	Clips	Hours	Spatial Rels	Temporal Ann.	Interact.
VideoMix22M [2]	22M	1M+	–	–	no
Kinetics [18]	650K	1,800	–	clip-level	no
Action100M [5]	147M	128K	–	approx.	no
CARLA [11]	–	–	–	–	no
BEDLAM [4]	30K	–	–	–	no
VirtualHome [25]	2.8K	–	–	action-level	no
VH Act. Genome [26]	5.8K	–	scene graphs	action-level	no
PVSG [31]	400	–	150K frames	annotated	yes
GEST [21]	25	–	–	sequential	no
Action Genome [17]	10K	82	1.7M	5 fr/action	yes
GEST-Engine (ours)	28.2K	14.3	**892M**	29.6K concurrent	yes

6.2 Human Evaluation of Video Generation

We collected 1,020 annotations from 16 independent evaluators across 766 unique videos, evaluating physical validity (binary) and semantic alignment (1–5 Likert scale). The GEST-Engine achieves 69% physical validity versus 18% (VEO) and 13% (WAN), and 4.09/5 semantic alignment versus 2.50 and 1.75 (Table 6.2). The 31% of engine videos judged invalid are visual artifacts of the underlying game engine (animation glitches, pathfinding jitter), not failures of the world model. Inter-annotator agreement was moderate for validity ($\alpha = 0.488$) and substantial for semantic alignment ($\alpha = 0.636$, with 85.4% of disagreements within one Likert point).

Reverse matching. As an independent validation, a separate group of annotators wrote textual descriptions based solely on each video, without seeing the original prompt; judges were then asked how well each description matched the original prompt anchor. GEST-Engine videos achieve the highest mean match score (56.64/100) and best average rank (1.42/3.00), followed by VEO 3.1 (42.94, 1.86) and WAN 2.2 (23.44, 2.72) — engine-generated videos are described in ways that most faithfully recover the specification.

Table 6.2: Human evaluation of video generation quality. Table from [9].

Category	#Videos	Validity (↑)	Semantic (↑)	Rank (↓)
GEST (ours)	281	**0.69**	**4.09**	**1.08**
VEO 3.1	255	0.18	2.50	2.04
WAN 2.2	230	0.13	1.75	2.88

6.3 Video Captioning

We fine-tuned two video captioning models — VideoLLaMA3 2B [32] and Qwen3-VL 2B [3] — under three conditions: fine-tuning on GEST data then fine-tuning on real video (G), fine-tuning on VEO data then fine-tuning on real video (V), and real-only training (R).

A distinctive property of the GTASA corpus is that every event has an exact frame range. We exploited this to develop a hybrid frame sampling strategy: for each action event in the GEST, we selected the mid-event frame (the frame at the temporal center of the event's duration), then filled remaining slots (up to 64 frames total) with uniformly sampled frames at 1 fps. This ensures that every action is represented by at least one frame, even for short-duration events that uniform sampling might miss entirely. Table 6.3 shows the effect: the hybrid approach improves 5 of 7 metrics, with the most notable improvement on CIDEr (+6.47).

Table 6.3: Frame sampling ablation on VideoLLaMA3 2B (G condition). Table from [9].

Method	BLEU-4	METEOR	CIDEr	ROUGE-L	SPICE	BERTSc.	BLEURT
Hybrid	**13.62**	**36.36**	**30.25**	**36.95**	**29.72**	90.64	54.13
Uniform	12.73	35.08	23.78	36.64	29.43	**90.65**	**54.59**

Table 6.4 presents the full captioning results. Models fine-tuned on GEST data first consistently outperformed the other conditions, achieving the best scores on six out of seven automatic metrics, despite the substantial visual domain gap. VEO first-stage fine-tuning consistently hurts performance relative to real-only training (V ranks last on VideoLLaMA3 with 4.00), while GEST first-stage fine-tuning helps. This suggests that visually realistic but semantically unreliable synthetic data is worse than no synthetic data at all, while semantically reliable data from a visually dated engine provides genuine benefit. An independent LLM jury (GPT-5.2 Pro, Gemini 3.1 Pro, Claude Opus 4.6) confirmed the same ranking.

6.4 Spatiotemporal Probing

We probed four frozen video encoders — V-JEPA 2 [2], VideoMAE V2 [29] (self-supervised), Qwen3-VL [3], VideoLLaMA3 [32] (VLM visual encoders) — across 11 spatiotemporal reasoning tasks using exact 3D ground truth from the corpus. Table 6.5 presents the results.

Self-supervised encoders encode spatial structure better than VLM visual encoders. V-JEPA 2 and VideoMAE V2 consistently outperform Qwen3-VL across nearly all tasks (average 60.6 and 59.9 vs 52.4), despite Qwen3-VL being trained on orders of magnitude more data with language supervision. We extend the observation of [12, 27] from image to frozen video encoders.

The largest encoder gap is on dynamic tasks. Motion Presence shows a 25.5-point

Table 6.4: Video captioning results. G = GEST then Real, R = Real-only, V = VEO then Real, H = Human reference. Metrics Rank averages over the 7 metrics (H assigned rank 1). Jury Rank averages over three frontier LLMs. All Avg averages metrics and jury ranks. Table from [9].

	VideoLLaMA3 2B				Qwen3-VL 2B			
Metric	H	G	R	V	H	G	R	V
BLEU-4	–	**13.62**	13.49	11.48	–	10.70	**11.66**	8.51
METEOR	–	**36.36**	34.02	32.71	–	**29.68**	29.40	27.59
CIDEr	–	**30.25**	29.32	21.89	–	21.46	**25.21**	17.31
ROUGE-L	–	**36.95**	35.28	34.33	–	**30.96**	29.63	28.93
SPICE	–	**29.72**	28.05	26.44	–	**25.06**	23.98	23.63
BERTScore	–	90.64	**90.67**	90.31	–	**89.99**	89.82	89.80
BLEURT	–	**54.13**	53.20	52.99	–	49.20	49.17	**51.15**
Metrics Rank (↓)	**1.00**	**2.14**	2.86	4.00	**1.00**	**2.43**	2.86	3.71
Jury Rank (↓)	**2.02**	2.51	2.52	2.94	**1.45**	2.78	2.92	2.86
All Avg (↓)	**1.77**	**2.42**	2.61	3.21	**1.34**	**2.69**	2.90	3.07

Table 6.5: Balanced accuracy (%) on spatiotemporal probing tasks. VL3* = partial (8/11). Table from [9].

Category	Task	Rnd	V-JEPA 2	VMAE V2	Qwen3	VL3*
Scene	Actor Count	20.0	37.3	33.0	32.2	**61.0**
	Motion Pres.	50.0	**88.6**	87.6	63.1	–
Entity	Entity Pres.	50.0	**66.2**	63.0	56.2	–
	Camera Dist.	33.3	57.9	**58.2**	57.4	57.8
	Angle Change	50.0	**75.5**	75.3	62.4	60.1
	Appr./Recede	50.0	**70.9**	68.9	57.1	54.0
Pair	Depth Order	50.0	61.3	**63.0**	59.4	–
	Pair Direct.	12.5	**17.4**	16.4	15.1	15.7
	Pair Dist.	33.3	40.0	**41.9**	36.1	40.1
	Rel. Motion	50.0	67.8	**69.4**	61.5	62.8
Temporal	Event Bound.	50.0	**84.1**	81.7	75.6	78.7
Average (all 11)			**60.6**	59.9	52.4	–
Average (8 shared)			**56.4**	55.6	49.7	53.8

gap between V-JEPA 2 (88.6%) and Qwen3-VL (63.1%); Approach/Recede shows a 13.8-point gap. Tasks requiring integration of spatial change over time are where self-supervised encoders most clearly separate from VLM encoders.

VideoLLaMA3 excels at actor counting. On the 8 shared tasks, VideoLLaMA3 falls between the self-supervised encoders and Qwen3-VL, but dramatically outperforms all others on Actor Count (61.0% vs 37.3% for V-JEPA 2). Counting is a known failure mode of language-supervised vision encoders [23]; VideoLLaMA3's SigLIP encoder with DiffFP temporal compression may preserve distinct entity tokens better.

Pairwise compass direction remains difficult. Pairwise Direction scores are only marginally above the 12.5% random baseline across all encoders (15–17%), indicating that fine-grained directional reasoning between entities is poorly represented in current video encoders regardless of training objective.

6.5 Case Study: VLM Weak Points

We assessed whether frontier VLMs can detect physically invalid videos using 94 videos with full annotator agreement (45 valid, 49 invalid; 49 GEST, 25 VEO, 20 WAN). GPT-5.2 Pro, Gemini 3.1 Pro, and Claude Opus 4.6 were prompted with the same validity instructions used in the human study (Table 6.6).

Table 6.6: Frontier VLM performance on video validity detection. Table from [9].

Model	Accuracy	Precision	Recall	F1
gpt-5.2-pro	56.38	**68.94**	54.54	44.84
gemini-3.1-pro	56.38	59.76	57.44	54.09
claude-opus-4.6	55.32	60.21	56.60	51.62
Jury (maj.)	**58.51**	60.23	**59.21**	**57.70**

All three models hover around 56% accuracy — barely above the 47.9% a trivial always-valid classifier would achieve. The failure is systematic rather than model-specific: majority-voting across the three models gives only 58.51%. Most telling is the qualitative behavior: when asked to caption these invalid videos, the models ignore the inconsistencies entirely and narrate events smoothly even when objects teleport or actors change identity mid-scene. They are not merely failing to detect inconsistencies — they are not representing the object permanence and identity tracking that would make detection possible.

6.6 Discussion

A recurring theme is that semantic consistency outweighs visual realism. Our engine-generated videos — despite 2004-era graphics — outperform state-of-the-art neural generators on semantic faithfulness, provide more effective pretraining data than photorealistic VEO-generated videos, and enable diagnostic probing tasks that no real-world dataset can support. This demonstrates that explicit world models with structured representations provide complementary value to the visual realism of neural generation.

Chapter 7

Conclusions and Future Work

7.1 Summary of Contributions

This thesis investigated visual understanding from biological and computational perspectives, producing five contributions:

Brain signals and visual understanding [10]. EEG recordings during visual recognition revealed that frontal electrodes associated with higher cognitive processing contribute more to classification accuracy than occipital electrodes over the visual cortex. Combining EEG with deep visual features improved recognition from 91% to over 97%, confirming that brain signals carry complementary information. This motivated the search for explicit, conceptual representations of visual content.

The GEST representation [21]. The Graph of Events in Space and Time decomposes visual scenes and textual descriptions into events connected by temporal, spatial, and semantic relationships. GEST outperformed established text similarity metrics through graph matching, and GEST-generated videos achieved 87% human preference against neural baselines.

The GEST-Engine [7, 9]. A complete system for generating multi-actor videos with dense frame-level annotations at zero marginal cost. The engine implements temporal orchestration via Allen's interval algebra with Floyd-Warshall constraint satisfaction, a chain-ID system for object identity, and multi-modal artifact collection (RGB, instance segmentation, spatial relation graphs, event-frame mappings, natural language descriptions). The system comprises approximately 100,000 lines of code developed by a single researcher over six years.

Agentic text-to-GEST generation [8]. An LLM-based system that separates narrative planning from constraint enforcement, achieving 80% end-to-end success versus 0% for staged LLM pipelines. Agentic narratives win 79% of text and 74% of video comparisons against procedural baselines. With Relation Subagents populating logical and semantic edge types, this is the first approach to exercise the full expressive capacity of the GEST formalism.

Experimental validation [9]. The GTASA corpus (938 videos, 892M spatial relations) demonstrated three findings: engine-generated videos achieve 69% physical validity versus 18%/13% for VEO/WAN; GEST pretraining outperforms VEO and real-only training for video captioning; and self-supervised encoders encode spatial structure significantly better than VLM visual encoders, while frontier VLMs perform near chance (56%) at detecting physically invalid videos.

7.2 Future Work

Several directions extend this work. Porting to GTA V via FiveM would substantially narrow the visual domain gap while preserving semantic guarantees. The cinematic camera mode, implemented at the infrastructure level, requires end-to-end validation. The agentic system's approximately 50% autonomous success rate can be improved through better variation strategies and loop detection. Using GEST-generated videos as stimuli for EEG data collection would close the loop back to the biological foundations of visual understanding explored in Chapter 2.

A particularly promising direction is using GESTs as a conditioning modality for neural video generators. Current text-to-video models take a single text prompt as input and must infer all spatial layout, temporal ordering, and actor coordination from the text alone — which, as our experiments show, they frequently fail to do. A GEST provides exactly the structured information that text lacks: which actors exist, what they do, in what order, and with what temporal constraints. We envision a two-stage pipeline: the GEST-Engine generates a semantically correct but visually dated video, and a neural model (e.g., a video-to-video translation network or a GEST-conditioned diffusion model) transforms it into photorealistic footage while preserving the semantic content. The GTASA corpus, with its paired GEST-video-text triplets, provides a natural training set for this approach. This direction would combine the semantic reliability of explicit world models with the visual realism of neural generation, and would contribute to the explainability of AI-generated video: because the GEST specification is human-readable and every frame is traceable to a specific event in the graph, the generated content would be interpretable and auditable in ways that purely neural generation is not.

7.3 Closing Remarks

This thesis began with the question of how visual understanding works — both in biological and computational systems — and arrived at a system for generating, annotating, and evaluating visual content through structured event representations. The journey from reading brain signals in Chapter 2 to generating multi-actor narrative videos in Chapter 4 may appear

circuitous, but it follows a single thread: the conviction that visual understanding, at its core, is about events — who did what, where, when, and why — not about pixels.

The GEST representation embodies this conviction. By placing events and their relationships at the center, rather than frames and their features, GEST provides a level of description where vision and language naturally meet, where generation and understanding share the same vocabulary, and where the gap between what a machine can produce and what it can explain narrows to zero. Every video the GEST-Engine produces is fully explained by the graph that generated it; every annotation is exact because it comes from the world model, not from a post-hoc labeling effort.

We believe this approach — explicit world models with structured event specifications — will become increasingly important as the field moves from generating short clips to producing long, coherent, multi-actor narratives. Neural generators will continue to improve in visual realism, but semantic reliability will require the kind of formal structure that representations like GEST provide. The two paradigms are not competitors but complements: explicit models generate the structured data and semantic guarantees, implicit models generate the visual realism and perceptual richness.

The tools we have built — the representation, the engine, the agentic generator, the corpus, and the experimental methodology — are publicly available. We hope they will serve not only as research infrastructure for the video understanding community, but also as a foundation for the longer-term vision of bridging perception, language, and thought through structured visual representations.

Bibliography

[1] J. F. Allen. "Maintaining knowledge about temporal intervals". In: *Communications of the ACM* 26.11 (1983), pp. 832–843.

[2] M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, M. Muckley, A. Rizvi, C. Roberts, K. Sinha, A. Zholus, et al. "V-jepa 2: Self-supervised video models enable understanding, prediction and planning". In: *arXiv preprint arXiv:2506.09985* (2025).

[3] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, et al. "Qwen3-vl technical report". In: *arXiv preprint arXiv:2511.21631* (2025).

[4] M. J. Black, P. Patel, J. Tesch, and J. 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*. 2023, pp. 8726–8737.

[5] D. Chen, T. Kasarla, Y. Bang, M. Shukor, W. Chung, J. Yu, A. Bolourchi, T. Moutakanni, and P. Fung. "Action100M: A Large-scale Video Action Dataset". In: *arXiv preprint arXiv:2601.10592* (2026).

[6] S. Chen, X. He, L. Guo, X. Zhu, W. Wang, J. Tang, and J. Liu. "Valor: Vision-audio-language omni-perception pretraining model and dataset". In: *arXiv preprint arXiv:2304.08345* (2023).

[7] N. Cudlenco, M. Masala, and M. Leordeanu. "[Tiny Paper] GEST-Engine: Controllable Multi-Actor Video Synthesis with Perfect Spatiotemporal Annotations". In: *ICLR 2026 the 2nd Workshop on World Models: Understanding, Modelling and Scaling*. 2026. URL: https://openreview.net/forum?id=uUofPYVMZH.

[8] N. Cudlenco, M. Masala, and M. Leordeanu. *Agentic Video Generation: From Text to Executable Event Graphs via Tool-Constrained LLM Planning*. Submitted to the CVPR 2026 Workshop on Agentic AI for Visual Media. 2026. arXiv: 2604.10383 [cs.CV]. URL: https://arxiv.org/abs/2604.10383.

[9] N. Cudlenco, M. Masala, and M. Leordeanu. *GTASA: Ground Truth Annotations for Spatiotemporal Analysis, Evaluation and Training of Video Models*. Submitted to the European Conference on Computer Vision (ECCV 2026). 2026. arXiv: 2604.10385 [cs.CV]. URL: https://arxiv.org/abs/2604.10385.

[10] N. Cudlenco, N. Popescu, and M. Leordeanu. “Reading into the mind’s eye: Boosting automatic visual recognition with EEG signals”. In: *Neurocomputing* 386 (2020), pp. 281–292.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. “CARLA: An open urban driving simulator”. In: *Conference on robot learning*. PMLR. 2017, pp. 1–16.

[12] M. El Banani, A. Raj, K.-K. Maninis, A. Kar, Y. Li, M. Rubinstein, D. Sun, L. Guibas, J. Johnson, and V. Jampani. “Probing the 3d awareness of visual foundation models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 21795–21806.

[13] R. W. Floyd. “Algorithm 97: shortest path”. In: *Communications of the ACM* 5.6 (1962), pp. 345–345.

[14] Google. “Veo 3 Model Card”. In: (2025). Accessed: March 04, 2026. URL: https://storage.googleapis.com/deepmind-media/Model-Cards/Veo-3-Model-Card.pdf.

[15] W. Hong, M. Ding, W. Zheng, X. Liu, and J. Tang. “Cogvideo: Large-scale pretraining for text-to-video generation via transformers”. In: *arXiv preprint arXiv:2205.15868* (2022).

[16] A. G. Huth, W. A. De Heer, T. L. Griffiths, F. E. Theunissen, and J. L. Gallant. “Natural speech reveals the semantic maps that tile human cerebral cortex”. In: *Nature* 532.7600 (2016), pp. 453–458.

[17] J. Ji, R. Krishna, L. Fei-Fei, and J. C. Niebles. “Action genome: Actions as compositions of spatio-temporal scene graphs”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10236–10247.

[18] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. “The kinetics human action video dataset”. In: *arXiv preprint arXiv:1705.06950* (2017).

[19] L. Khachatryan, A. Movsisyan, V. Tadevosyan, R. Henschel, Z. Wang, S. Navasardyan, and H. Shi. “Text2video-zero: Text-to-image diffusion models are zero-shot video generators”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 15954–15964.

[20] M. Leordeanu and M. Hebert. “A spectral technique for correspondence problems using pairwise constraints”. In: *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*. Vol. 2. 2005, 1482–1489 Vol. 2. DOI: 10.1109/ICCV.2005.20.

[21] M. Masala, N. Cudlenco, T. Rebedea, and M. Leordeanu. “Explaining vision and language through graphs of events in space and time”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 2826–2831.

[22] M. Masala and M. Leordeanu. “From Vision To Language through Graph of Events in Space and Time: An Explainable Self-supervised Approach”. In: *arXiv preprint arXiv:2507.04815* (2025).

[23] R. Paiss, A. Ephrat, O. Tov, S. Zada, I. Mosseri, M. Irani, and T. Dekel. “Teaching clip to count to ten”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 3170–3180.

[24] S. F. Popham, A. G. Huth, N. Y. Bilenko, F. Deniz, J. S. Gao, A. O. Nunez-Elizalde, and J. L. Gallant. “Visual and linguistic semantic representations are aligned at the border of human visual cortex”. In: *Nature neuroscience* 24.11 (2021), pp. 1628–1636.

[25] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba. “Virtualhome: Simulating household activities via programs”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8494–8502.

[26] Y. Qiu, Y. Nagasaki, K. Hara, H. Kataoka, R. Suzuki, K. Iwata, and Y. Satoh. “Virtualhome action genome: A simulated spatio-temporal scene graph dataset with consistent relationship labels”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 3351–3360.

[27] P. Tong, E. Brown, P. Wu, S. Woo, A. J. V. Iyer, S. C. Akula, S. Yang, J. Yang, M. Middepogu, Z. Wang, et al. “Cambrian-1: A fully open, vision-centric exploration of multimodal llms”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 87310–87356.

[28] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, et al. “Wan: Open and advanced large-scale video generative models”. In: *arXiv preprint arXiv:2503.20314* (2025).

[29] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, and Y. Qiao. “Videomae v2: Scaling video masked autoencoders with dual masking”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 14549–14560.

[30] R. Wang, J. Yan, and X. Yang. “Neural graph matching network: Learning lawler’s quadratic assignment problem with extension to hypergraph and multiple-graph matching”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[31] J. Yang, W. Peng, X. Li, Z. Guo, L. Chen, B. Li, Z. Ma, K. Zhou, W. Zhang, C. C. Loy, et al. “Panoptic video scene graph generation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 18675–18685.

[32] B. Zhang, K. Li, Z. Cheng, Z. Hu, Y. Yuan, G. Chen, S. Leng, Y. Jiang, H. Zhang, X. Li, et al. “Videollama 3: Frontier multimodal foundation models for image and video understanding”. In: *arXiv preprint arXiv:2501.13106* (2025).