Evaluating Simulation Platforms for Visual Affordance Understanding in Computer Vision

Maria Gabriela Lustosa Oliveira
Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas (Unicamp)
Campinas, São Paulo
Email: m188504@dac.unicamp.br

Paula Dornhofer Paro Costa
Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas (Unicamp)
Campinas, São Paulo
Email: paulad@unicamp.br

***Abstract*—Affordance understanding in computer vision goes beyond object recognition — it involves interpreting scenes in terms of potential agent-object interactions. This work examines the role of simulated environments in supporting such reasoning in visual scenes. Three simulation tools (Gymnasium, SUMO, and CARLA) are evaluated regarding their suitability for affordance research, with special attention to video-based data generation, multimodal perception, and interaction modeling. Within the Gymnasium library, selected environments are compared based on criteria such as element countability, scene diversity, and logging capabilities. The analysis identifies key limitations in existing tools and underscores the need for scalable, user-configurable platforms designed to support perception, learning, and generalization in affordance-centric applications.**

I. Introduction

Fundamental elements for the survival and adaptation of organisms include the ability to perceive their environment and infer actions based on the dispositional capabilities of these elements [1]. In artificial systems, this perceptual ability has been primarily developed through computer vision, which plays a central role in enabling machines to interpret visual input. Advances in this field are essential not only for basic perception but also for supporting more complex reasoning processes. Such capabilities are especially valuable in domains like engineering, robotics, and artificial intelligence (AI), where systems must react appropriately to their surroundings.

Historically, computer vision has focused on tasks such as segmentation, classification, and labeling of objects in images. Systems like CLIP [2] and Segment Anything [3] showcase advancements in visual element identification and categorization. However, these applications typically did not require models that could think about possible interactions between elements in a scene. The success in descriptive tasks has led to new research focused on the contextual interpretation of scenes, which is crucial for applications such as autonomous driving and robotic manipulation. In these contexts, simply recognizing objects is insufficient. It is essential to understand the relationships between scene components, predict actions, and plan pathways that adhere to the relevant goals and rules.

This brings us to the concept of affordances, originally introduced by Gibson [1]. He suggested that perception extends beyond merely identifying objects. It encompasses the immediate recognition of actions that can be performed with or initiated by those objects. Affordances enable us to describe how humans and other animals perceive their environment without requiring an intermediary classification process. An example is an open door: both a person and a cat can recognize the potential to pass through it, even though they may have different internal representations of the world.

Based on this comprehension of affordances, a key challenge is how to develop computational models of affordance learning and understanding capable of reproducing mechanisms similar to human intelligence. This requires visual environments that present a great diversity of objects, agents, and changing dynamics over time. Also, following the traditional machine learning paradigm, we need to train these models and have the means to annotate the elements and actions in the observed scenes.

Although real-world video annotated datasets exist, they are far from being adequate for this type of research. Typical limitations include scale, annotation quality, and variability. Even the most prominent datasets, like Ego-Exo4D [4], EPICKITCHENS [5], and HVU [6], are restricted to dozens of scenario categories or many different examples of the same type of environment. Furthermore, while their annotations may be high-quality and their examples naturalistic, they lack the ability to represent fine-grained variations in object arrangements within the same scene.

An envisioned approach is the adoption of simulated environments, which offer viable alternatives for generating data when suitable real-world datasets are unavailable. Simulations allow us full control over the environment, enabling us to define the number and the type of agents and objects, track element positions, monitor when objects will appear or disappear, and capture interactions. This control allows for the automatic generation of labeled datasets with high-quality annotations specifically tailored to computer vision tasks.

These simulated environments are valuable for affordance modeling because effective scene understanding and reasoning benefit from minimal variation in background scenes while allowing controlled variation in object configurations. Besides, video data is essential not only for representing the temporality of actions, but also for enabling us to predict future actions and perform temporal reasoning. Despite the abundance of image-based annotated datasets, video datasets that capture contin-

uous agent-object interactions with subtle scene variation are still rare, limiting experiment design.

In this context, this paper analyzes three simulation tools from the perspective of visual world understanding research, recognizing the demand for video-annotated and task-specific datasets in the field. The agents operating within these environments can be trained via reinforcement learning methodologies. Also, we explore two traffic simulators, which have been widely used in autonomous driving tests and routine planning.

It is important to note that the selected environments are well-established within their respective domains. Gymnasium [7] was chosen for being renowned for its reinforcement learning capabilities, while SUMO [8] and CARLA [9] are commonly used in the realm of autonomous driving.

II. Method

Following the previous motivation, our objective was to identify simulation tools capable of supporting affordance research. Our focus relies particularly on tasks that involve visual world understanding, multimodal perception, and interaction-based reasoning. Since affordances depend on perceiving not only the presence of objects but also their potential for action, the environments must offer more than visual recognition. They must allow controlled access to relationships and dynamics that emerge during agent-object interaction.

The environments analyzed in this work were selected based on their prominence within their specific research areas related to visual world understanding and action planning. Gymnasium [7] library is widely recognized in the reinforcement learning community for offering a broad selection of environments designed to test and benchmark agent behaviors. In the context of autonomous driving and multimodal systems that incorporate inertial and spatial data, the traffic simulators SUMO [8] and CARLA [9] are well-established tools.

The selection of environments involved systematically exploring the options available within these simulation tools. In particular, a wide range of Gymnasium-based environments was explored. These included Atari games, First-Party environments such as Gymnasium-Robotics and the PettingZoo multi-agent reinforcement learning suite, as well as a variety of Third-Party environments maintained by the broader research community rather than by the Farama Foundation itself [7]. Several criteria were used to include or exclude environments.

A fundamental requirement is the possibility to count the types and quantities of objects present in the scene at any given time step or iteration. For instance, in the Knights Archers Zombies game from PettingZoo, there are two archers, two knights, and a user-defined number of zombies. The archers shoot at the zombies, while the knights chase them as they appear at the top of the screen. Because the goal is to study controlled environments, it is necessary to be aware of which objects are present, where they are, and when they occur.

Additionally, the environment should include a variety of countable objects, agents, and scenes. This diversity is critical for evaluating whether models can generalize perceptual understanding beyond memorizing a limited set of patterns. Multi-agent simulators are ideal for this purpose. For example, while the Gymnasium-Robotics Franka Kitchen features only a 9-DoF Franka robot, it includes multiple household items that enable multitasking. Therefore, even without much diversity, we may still use this environment for testing purposes.

Besides, it is necessary that the environments support multimodal reasoning. In particular, audio channels are a desirable feature, as they open possibilities for deeper and more general scene understanding. Control over the scene is another key factor. Since the project focuses on modeling agent-agent, agent-object, and object-object interactions, environments have to provide access to information about all elements that are being simulated at every time step.

Ease of use is also an important criterion. Given the practical need to explore and run multiple simulations, environments that require a simple setup and can run on a CPU were prioritized. This ensured that the environments could be evaluated without exceeding a few hours.

The environments that satisfied the criteria presented above were selected for further analysis. In the next section, we present and compare them based on their alignment with the requirements. We also highlight their potential and limitations for interaction modeling and video-annotated data generation.

III. Results

Our search for simulated environments is summarized in Table I. Each column header in the table corresponds to one of the criteria used for selecting the environments.

The environments selected for further analysis include a range of gym-based simulations: specific Atari environments (Double Dunk and River Raid), Knights Archers Zombies, Franka Kitchen, Pokemon Red, and a basic custom grid-based board featuring dynamic agents that we developed. These environments share several characteristics that match our selection criteria. They contain numerous identifiable objects, present a manageable level of technical complexity, and operate on a CPU. Additionally, they support various interactions, allow for tracking and logging scene elements, and, despite being stylized, exhibit diversity along with some similarities to the real world, even in the absence of humanoid agents.

We initially focused on environments that support multiple agents acting simultaneously, which we refer to as multi-agent environments. However, we also included environments with only one agent operating within the scenes, which we call single-agent environments.

Traffic simulators were also considered in our initial search due to their popularity in action-planning applications. However, they were excluded from the final analysis. SUMO, for example, enables identifying agents in the environment at each time step and provides access to their spatial coordinates over time. It is lightweight and easy to use, but limited in diversity: the only countable agents and objects are vehicles and traffic lights. Additionally, there are no non-humanoid agents, and the visual rendering consists of schematic top-down maps that lack realism. SUMO does not support audio channels either. Since our goal is to generate data that can help train more

TABLE I
COMPARISON OF SIMULATED ENVIRONMENTS BASED ON KEY CRITERIA FOR SUPPORTING AGENT-OBJECT INTERACTION STUDIES. ALL OF THEM USE THE GYMNASIUM BASIC STRUCTURES IN THEIR IMPLEMENTATIONS.

Simulator name	Element counting	Countable elements	Audio	Difficulty	Diversity	Scene control
Atari Environments	No	Some games have a fixed predetermined number of countable elements (eg. the 2v2 basketball game Double Dunk) while others have a random quantity (eg. River Raid)	No	Easy, because it has a very limited range of technical changes that can be done	Depends on game	Low. It is not possible to add and remove objects from an environment
Knights Archers Zombies [First-Party environment]	Yes	There are two knights and two archers that are fixed, as well as a variable number of zombies. The number of zombies may be set	No	It's easy to implement, but plotting RL training rewards requires some previous experience	Low. There is only one scene and three types of agents. Possible actions are limited to: shooting an arrow, a knight attack, and zombies moving around	Medium. We know the elements, but we can not perform fine-grained changes in their disposition
Franka Kitchen [First-Party environment]	Yes	It is single agent. There is only one environment; thus, we always know what objects are in the scene. Once we tell what action we want the robot to perform, we also know what objects are used in interaction	No	Medium. Implementing is easy, but training is not. There were lots of conflicts during training	Medium. There is one environment: a kitchen, but many (previously trained) actions are possible	Low. Actions and object quantity are fixed
Pokémon Red [Third-Party environment]	No	Theoretically, it is possible to maintain a log of all agents and objects present in a scene, as long as they are categorized within their respective classes. However, accessing this data for each episode can be challenging. Additionally, the names assigned to objects and agents may not accurately reflect their true descriptions	By default, no. But once it is a custom environment, it is possible to add. It is not easy, though	The pretrained version is easy to use. However, the custom version requires implementing and training all features. This is harder mainly because the documentation is incomplete, and the GitHub implementation uses deprecated libraries	High. There are many types of agents, such as "Pokémon" and the character "Ash", in addition to various objects, like buildings on the map. Besides, there are scenes of battles between "Pokémon" and menus for possible interactions with environmental objects. The game is designed in grayscale	In the pretrained version, there are only Ash's moves
Table with Dynamic Agents [Custom]	Yes	Anything one may want	Yes, it is possible to add	Low, if you implement a simple game. It may considerably increase with more complex movements, significant interactions between objects and agents, and interactions between objects themselves	High. One can add many objects were necessary and of any kind	Complete

generalizable models, environments that offer richer and more varied representations were prioritized.

In contrast, the CARLA simulator offers a wider variety of countable agents and objects, including different types of vehicles, pedestrians, traffic lights, and other elements. While CARLA does not have audio channels by default, they can be added through external libraries such as Pygame. It features detailed environments that allow for customized scenes but require significant processing power to run, even with a GPU. This high computational requirement, combined with a steep learning curve, makes its usability challenging. While CARLA might be a good candidate for future work, its technical constraints motivated us to focus on environments that are more accessible and flexible.

During implementation, several reproducibility issues occurred. It was common to find that environments depended on outdated library versions, often leading to compatibility errors. These barriers were mostly noticeable when preparing environments for training using reinforcement learning techniques.

To ensure the data generated from simulations could support supervised model training, we prioritized environments that either had a fixed number of elements or were able to generate

a complete log of which agents and objects appeared in each scene and at what time. This logging is critical for producing high-quality annotations.

Analyzing Table I, we observe that the Custom Table with Dynamic Agents environment stands out regarding element counting and control. It allows complete scene specification and supports logs for every interaction. Therefore, it provides superior scene control and is the most flexible in terms of countable content.

Some Atari environments also offer a reasonable number of distinct elements. However, they do not allow explicit specification of which elements are present at each time step, limiting their usefulness. Additionally, although abstraction is acceptable — and often desirable — for generalization, the extreme stylization and pixelated graphics in some Atari games can result in representations that may be too detached from real-world contexts. Because our goal is to generate training data that can generalize across diverse scenarios, we need environments with a more balanced level of abstraction and resemblance to real-world settings.

In terms of audio, none of the tested environments has this feature by default. Since they require manual implementation in any case, the most suitable option is the Custom Table. It offers a simplified and controlled structure, which makes it easier to add audio functionalities without interfering with other parts of the simulation or creating compatibility issues.

For ease of use, Atari environments offer extensive official documentation and tutorials on essential functions for beginners [7]. The Knights Archers Zombies environment is similarly straightforward to run and understand. Pokémon Red, when used with its pretrained model, is also relatively easy to interact with. Besides, excluding the Custom Table, Pokémon Red is the most diverse in terms of agents, actions, and environmental configurations.

Finally, although running Gym environments with pretrained agents is simple, training these agents from scratch using reinforcement learning requires previous experience. This includes analyzing reward curves, interpreting TensorFlow graph outputs, and understanding training behaviors [10]. These factors may affect the ease with which new users can extend or customize the environments.

IV. Conclusion

In summary, the environments analyzed in Table I present various trade-offs across the criteria we defined. The Custom Table environment meets most requirements and provides the highest control and flexibility over scenes. While each environment has its strengths, they all exhibit limitations that make their use difficult as an experimental tool for annotated video-data generation specific to affordance research.

This gap highlights a lack of simulation infrastructure specifically designed to address the needs of visual reasoning and interaction modeling. Notably, the Custom Table emerged as the most promising option among those explored, which leads us to a broader insight. What is truly needed is not just a custom environment, but rather a flexible simulation platform that allows researchers to design and conduct their own experiments with minimal overhead.

Thus, we propose the concept of a User Simulator: a simulation platform that provides several semi-realistic, versatile scenes (such as a household kitchen, a mechanical workshop, or an outdoor park) in which users can specify which and how many agents or objects to include. By using this tool, there would be no need to implement all the physical dynamics and rendering from scratch. The User Simulator would combine the usability of existing datasets with the flexibility of simulation, offering full control over the content and dynamics of the environment. It further enables researchers to log which agents and objects are present, when and where they interact, and how these interactions unfold over time. Similar to annotated datasets such as Ego-Exo4D [4], it is outlined to operate at a significantly larger scale and with finer control.

By enabling subtle yet systematic variations in object and agent configurations while preserving the overall context of the scene, a User Simulator would support the creation of robust datasets for training models capable of generalizable affordance understanding. This direction appears to be the most viable path forward, providing the experimental framework that current tools lack and fulfilling the demands of affordance-based research in computer vision and AI.

Acknowledgment

This project was supported by the Brazilian Ministry of Science, Technology and Innovations, with resources from Law nº 8,248, of October 23, 1991, within the scope of PPI-SOFTEX, coordinated by Softex and published Arquitetura Cognitiva (Phase 3), DOU 01245.003479/2024 -10.

References

[1] J. J. Gibson, *The Ecological Approach to Visual Perception*. Psychology Press, 1979, ch. The Theory of Affordances, pp. 119–135, accessed: Nov. 5, 2024. [Online]. Available: https://www.taylorfrancis.com/books/mono/10.4324/9781315740218/ecological-approach-visual-perception-james-gibson

[2] OpenAI, "CLIP: Connecting Text and Images," 2021, accessed: Oct. 29, 2024. [Online]. Available: https://openai.com/index/clip/

[3] MetaAI, "Segment Anything Model (SAM): a new AI model from Meta AI that can "cut out" any object, in any image, with a single click," 2023, accessed: Nov. 5, 2024. [Online]. Available: https://segment-anything.com/

[4] K. Grauman et al., "Ego-Exo4D: Understanding Skilled Human Activity from First- and Third-Person Perspectives," 2024, accessed: Jun. 30, 2025. [Online]. Available: https://arxiv.org/abs/2311.18259

[5] D. Damen et al., "Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100," *International Journal of Computer Vision (IJCV)*, vol. 130, p. 33–55, 2022, accessed: Jun. 30, 2025. [Online]. Available: https://doi.org/10.1007/s11263-021-01531-2

[6] A. Diba et. al, "Large Scale Holistic Video Understanding," in *European Conference on Computer Vision*. Springer, 2020, pp. 593–610, accessed: Jun. 30, 2025.

[7] Farama Foundation, "Gymnasium Documentation," 2024, accessed: Apr. 27, 2025. [Online]. Available: https://gymnasium.farama.org/#

[8] A. Lopez et al., "Simulation of Urban Mobility (SUMO)," accessed: Jun. 30, 2025. [Online]. Available: https://doi.org/10.5281/zenodo.15359663

[9] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16, accessed: Jun. 30, 2025.

[10] Google Brain Team, "Tensorboard graphs," https://www.tensorflow.org/tensorboard/graphs, 2024, accessed: Jul. 4, 2025.