V-IRL: Grounding Virtual Intelligence in Real Life

Jihan Yang[1]* Runyu Ding[1] Ellis Brown[2] Xiaojuan Qi[1] Saining Xie[2]
[1]The University of Hong Kong [2]New York University

https://jihanyang.github.io/projects/VIRL

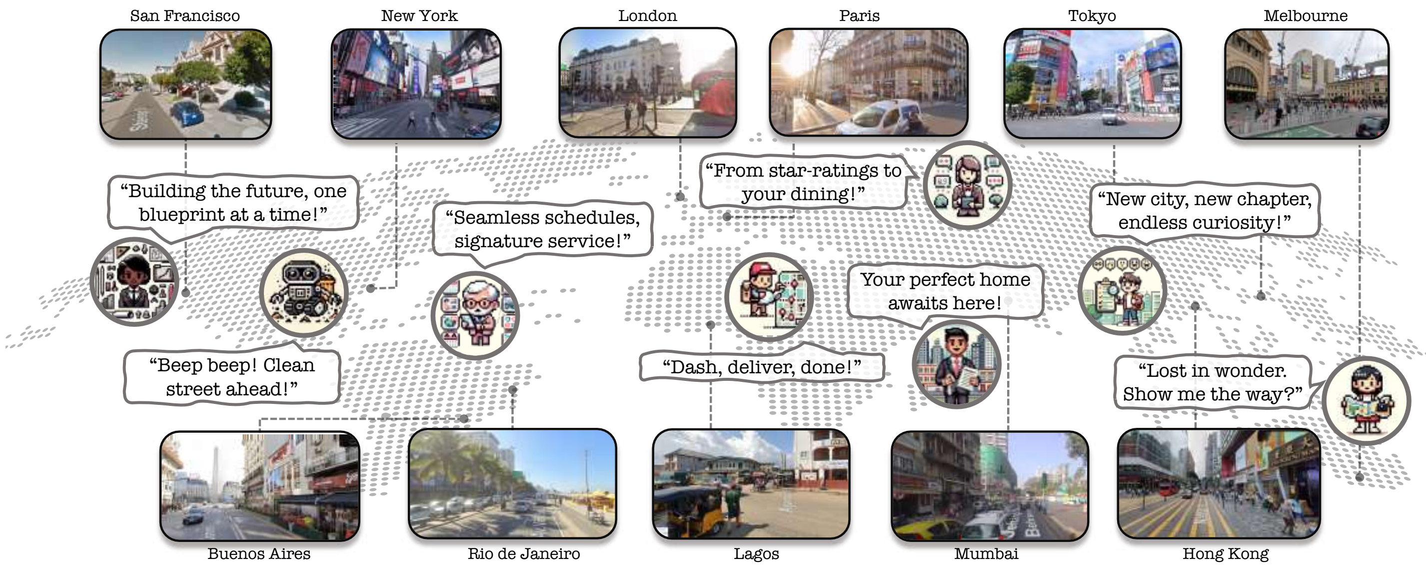


Figure 1. V-*IRL* agents leverage real-world geospatial information and street view imagery to navigate urban terrains, execute complex tasks, and interact in real-time scenarios. From recommending relevant destinations to assessing city infrastructure to collaboratively giving & following verbal directions—we develop agents that illustrate V-*IRL*'s current capabilities, flexibility, and utility. Above all else, we present a flexible platform for researchers to harness the endless available real-world data to create and test diverse autonomous agents.

Abstract

There is a sensory gulf between the Earth that humans inhabit and the digital realms in which modern AI agents are created. To develop AI agents that can sense, think, and act as flexibly as humans in real-world settings, it is essential to bridge the realism gap between the digital and physical worlds. How can we embody agents in an environment as rich and diverse as the one we inhabit, without the constraints imposed by real hardware and control? Towards this end, we introduce V-IRL: a platform that enables agents to scalably interact with the real-world in a virtual yet realistic environment. Our platorm serves as a playground for developing agents that can accomplish a variety of practical tasks, and as a vast testbed for measuring progress in capabilities spanning perception, decision-making, and interaction with real-world data spanning the entire globe.

1. Introduction

The advent of large language models (LLMs) has breathed new life into autonomous agent research by offering a universal interface for diverse capabilities, ranging from basic reasoning to complex planning and tool use [41]. While these developments are promising, a majority of these agents remain confined to text-based environments or simplistic simulations. Visual components in existing agents are either rudimentary—such as simulated tabletop environments [4, 14]—or rely on abstracted representations using ground-truth APIs [13, 39]. Furthermore, the prevalent visual models employed by these agents are trained on photogenic, object-centric Internet images, which fail to capture the unpredictability and diversity of real-world scenes.

This paper aims to bridge this gap between AI agents and the sensory world by grounding them in rich, real-world environments—a crucial step towards developing autonomous agents that can effectively operate in real-life sce-

*Part of the work conducted during a visit to NYU.

narios. Our novel setting for AI agents *necessitates* rich sensory grounding and perception: virtual embodiment within cities around the globe using real visual and geospatial data.

To this end, we introduce V-*IRL*, a versatile platform for building and testing virtual agents within this novel setting. V-*IRL* harnesses the power of mapping and street view data, enabling agents to navigate real-world locations, access up-to-date information about their surroundings, and perform practical tasks. With geospatial coordinates at its core, V-*IRL* is flexible and extensible, integrating with arbitrary geospatial platforms and APIs. Moreover, V-*IRL* opens up a vast sea of visual data, allowing a simple and extensible way for researchers to evaluate vision models on realistic data distributions.

We demonstrate the versatility and adaptability of V-*IRL* by developing a series of diverse exemplar agents, each solving a unique and practical task. As these agents hinge upon foundational language and vision models, it is critical to evaluate these models within this setting and their impact on agent performance. We leverage the vast data available through our platform to develop *global scale* benchmarks measuring the performance of underlying vision models on images from diverse geographic and cultural contexts—evaluating their adaptability to shifting environmental, architectural, and language-specific elements. Furthermore, we evaluate the contributions of language and vision models to agent performance on challenging tasks. Our results illustrate the potential of V-*IRL* in bridging the gap between virtual agents and visually rich real-world environments, paving the way for future research in this direction.

In summary, our contributions are:

- A **real-world setting** for virtual agents that *necessitates* rich sensory grounding and perception: embodiment using geospatial data and street-view imagery.
- **V-*IRL***: a platform for this setting that can be used to build and test perception-centric agents.
- Development of **diverse exemplar agents** that showcase the platform's versatility and adaptability.
- **Global benchmarks** measuring the performance of foundational language and vision models (1) in isolation on our platform's real-world data and (2) on end-to-end agent performance in challenging tasks.
- Discussion of the robustness of "open-world" vision models to *real-world* data from across the globe.

We are excited to see how the research community will leverage V-*IRL* to develop and test agents that can understand and interact with the real world.

2. Related Work

Here, we ground V-*IRL* to three streams of research.

2.1. AI Agents

Agents are autonomous entities capable of perceiving their environment and acting to achieve goals [40]. Historically, agent development has leveraged symbolic and reinforcement learning methods [15], which face issues of scalability and real-world utility. In contrast, the new wave of LLM-driven agents overcomes these challenges with text as a universal interface, enabling natural human interaction and adaptability to various tasks [26, 36]. Moreover, these models equip agents with diverse reasoning capabilities, such as complex planning and tool use [28, 32]. Yet a critical limitation persists: the agents in this new wave are entirely text-based, devoid of any tangible connection to the visual or sensory aspects of the real world.

2.2. Embodied AI

Embodied AI studies intelligent agents & robots perceiving and interacting with their environment. A significant challenge in this field is the acquisition of large quantities of realistic data. Consequently, robots are primarily trained in simulated environments to develop skills such navigation and manipulation [31, 42]. Recent advancements in LLMs [27] have enabled embodied agents to perform long-horizon tasks in game-engines [39]. However, the diversity of tasks and data is still too narrow and simplistic to enable them to operate flexibly in diverse real-world environments.

2.3. Open-World Computer Vision

Motivated by the success of vision-language models pre-trained on large-scale web-crawled data [2, 29, 44], open-world computer vision has received increasing attention in recent years [17, 19]. However, images and benchmarks sourced from the Internet [3, 9, 12, 16] are unavoidably biased towards specific distributions rather than truly reflecting the real *world* [30]. Because they are trained and evaluated on Internet data, existing "open-world" models are effectively more open-*Internet* than open-*world*.

3. Virtual Intelligence in Real Life

To show the versatility of the V-*IRL* platform, we use it to instantiate several exemplar agents in our virtual real-world environment. In this section, we engage these agents with tasks that highlight various capabilities of our platform. In Sec. 4, we discuss the technical details of our platform and how it enables agents to interact with the real world.

For illustration, we give V-*IRL* agents character metadata, including an 8-bit avatar, a name, a short bio, and an intention they are trying to accomplish. More concretely, agents are defined by pipelines that use this character metadata along with our platform's interface and models to address complex tasks (see Sec. 4). Here we provide a high-level overview of the tasks, highlight the V-*IRL* capabilities they require, and visualize the agents solving them.

3.1. Earthbound Agents

Agents in the V-*IRL* platform inhabit virtual representations of real cities around the globe. At the core of this representation are *geographic coordinates* corresponding to points on the Earth's surface. Using these coordinates, V-*IRL* allows agents to *ground* themselves in the world using maps, real street view imagery, information about nearby destinations, and additional data from arbitrary geospatial APIs.

Route Optimizer — ENV Map

Name: Peng Age: 21 Loc: NYC
Bio: Originally from Chengdu, Sichuan, Peng is a student at PKU. He just arrived for a semester abroad at NYC, and is couch surfing until he gets settled.
Intention: Peng needs to visit five locations around the city: his University Card Center, Residence Hall, Research Center, Library, and Student Center.

Task: Given a starting address and a list of waypoints, plan the shortest route to all waypoints and then follow it on street view.

Takeaway: V-*IRL* instantiates agents with *real* geospatial information, and enables useful tasks like route optimization.

Peng needs to visit several locations throughout the city to get documents signed for registration as a visiting student...

Leveraging Geolocation & Mapping capabilities, Peng saves 7 minutes by walking along the shortest path as opposed to in order waypoint visitation as shown in Fig. 2.

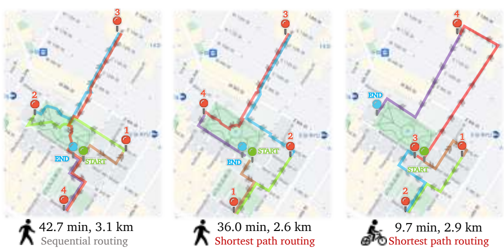


Figure 2. Finding the shortest path for Peng to travel to five places.

3.2. Language-Driven Agents

To tackle more complex tasks, we follow the pattern of language-driven agents [41]. LLMs enable agents to flexibly reason, plan and use external tools & APIs.

Place Recommender — ENV Map, LM LLM

Name: Aria Age: 26 Loc: NYC
Bio: A 3rd year graduate student who loves to try new restaurants. She is always looking for new places to try, and shares her favorite spots on her blog!
Intention: Pick out a lunch spot that Peng might like.

Name: Vivek Age: 35 Loc: NYC
Bio: A tech-savvy estate agent who combines his local knowledge with online tools like Zillow to find the perfect homes for his clients in the bustling city.
Intention: Help Peng find a place to live for the semester.

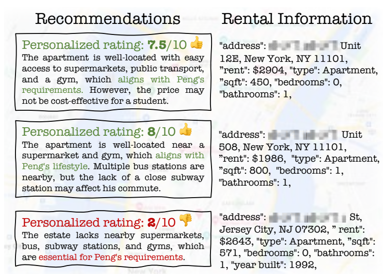
Task: Given specific location, background and intention, synthesize reviews of nearby businesses to provide a recommendation.

Takeaway: V-*IRL* exposes rich real-world information to agents that they can use for real-world tasks.

Peng is starving for some lunch but doesn't know where to eat... Luckily, he met a nice grad student Aria during his errands who might be able to help him find a good spot...

Aria searches for possible restaurants nearby. She then synthesizes public reviews to make final recommendations via GPT-4. As Peng is new to the city and originally from Sichuan, she recommends a spicy Chinese joint *Kwa Food Deep Fried Skewers* to give him a taste of home.

Peng hires Vivek to help him find an apartment in East Village, Jersey City, or Long Island City for $1k–$3k per month close to a gym, supermarket, and public transit...

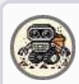


Vivek uses real estate APIs to find potential apartments in Peng's desired regions and price range. For each candidate, he researches its proximity to the places Peng cares about. Synthesizing these factors, Vivek provides a holistic rating and accompanying reasoning using GPT-4. His top recommendation is a cost-effective 1 bedroom apartment for $1986/mo, which is close to a supermarket, 2 bus stations, and a gym.

3.3. Visually Grounded Agents

Although language-driven agents can address some real-world tasks using external tools, their reliance on solely text-based information limits their applicability to tasks where *visual grounding* is required. In contrast, *real sensory input* is integral to many daily human activities—allowing a deep connection to and understanding of the real world around us. Agents can leverage street view imagery through the V-*IRL* platform to *visually ground* themselves in the real world—opening up a wide range of *perception-driven tasks*.

Urban Assistance Robot — ENV Map, CV Vision



Name: RX-399 Age: Unk. Loc: HK/NYC
Bio: This urban robot's advanced object detection, localization, and navigational telemetry systems allow it to perform perceptive tasks in busy city streets.
Intention: Report the locations of trash bins to the sanitation dept.

Task: Travel along a specified route and detect instances of a specified object (e.g., trash bins, hydrants, benches, etc.).

Takeaway: V-*IRL* agents can use perceptive input to understand and interact with their environment.

RX-399 is a state-of-the-art robot agent with advanced navigation and sensing capabilities. Its manufacturer is running a pilot program with sanitation departments in Hong Kong and New York City to assess its readiness for garbage duty...

RX-399 navigates along pre-defined city routes, tagging all trash bins using its open-world detector and geolocation module as depicted in Fig. 4. RX-399 can actively adjust its camera pose to the optimal view for each potential ob-

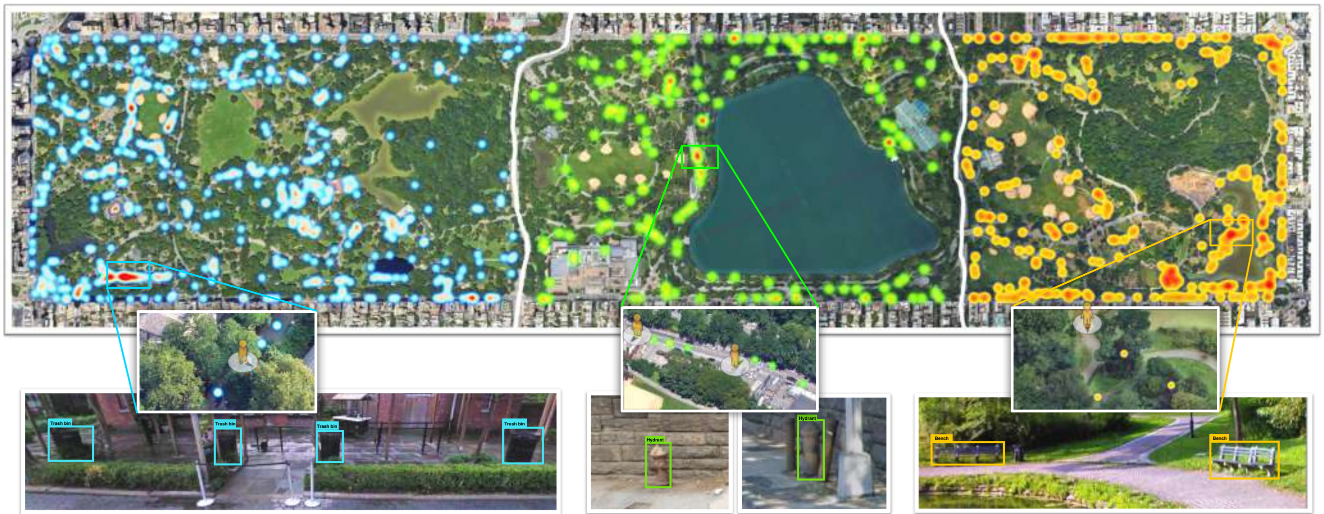


Figure 3. Imani's visualization of trash bins, fire hydrants, & park benches in NYC's Central Park using data collected by RX-399.

ject thanks to our interactive embodied environment and the sensor-rich visual input. *During the pilot in Hong Kong, RX-399 locates eight trash bins, correctly identifying five but overlooking one. In New York, it accurately detects all five trash bins but mistakenly reports two mail boxes.*

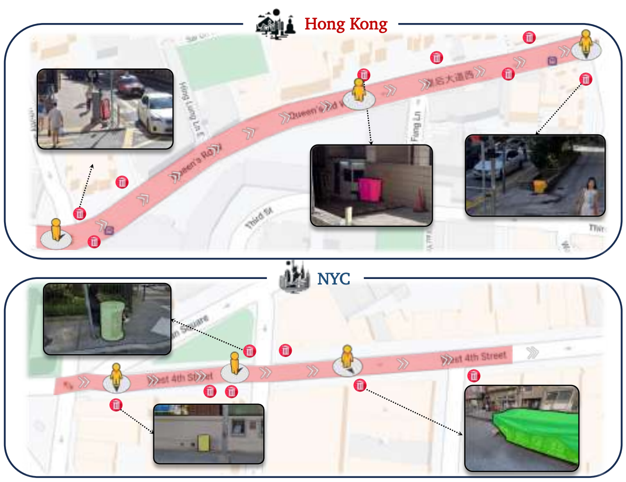


Figure 4. Portions of RX-399's system records in HK and NYC.

RX-399 can avoid double-counting seen objects by using feature matching to check for duplicates among prior detections (see Fig. 5).

Figure 5. RX-399 avoids double-counting trash cans by identifying duplicates across different viewpoints using feature matching.

Urban Planner ENV Map CV Vision

Name: Imani **Age:** 42 **Loc:** NYC
Bio: A sustainable urban development graduate, Imani is passionate about maintaining a harmonious balance between nature and urban ecosystems.
Intention: Use RX-399 to collect first-person data for her studies.

Task: Record the location of all instances of any specified objects (e.g., trash bins, hydrants, benches, etc.) in a specified region.

Takeaway: V-*IRL* enables realistic open-world applications requiring vast geospatial and first-person visual information.

Imani needs to analyze the distribution of trash bins, fire hydrants, and park benches in New York's Central Park for a project with the NYC Parks & Recreation department...

Imani sets routes spanning Central Park and objects of interest for RX-399, who traverses the routes and records all detected instances. After RX-399 finishes its route, Imani analyzes the collected data at different levels of detail. As depicted in Fig. 3, the coarsest level shows general distributions of trash bins, hydrants, and benches in the park. Imani can also zoom in to specific regions, where lighter colors represent positions with more unique instances identified. The following table presents RX-399's counting report:

Category	Trash Bin	Fire Hydrant	Park Bench*
Count	1059	727	1015

Table 1. RX-399's counting report in Central Park, New York City. (**Note: contiguous benches counted as one instance).*

By retrieving geotagged sensory-rich data within RX-399, Imani can also inspect the detection results for each object to help her verify the reliability of RX-399's reports as illustrated by the bottom level in Fig. 3.

Intentional Explorer ENV Map LM LLM CV Vision

Name: Hiro Age: 22 Loc: HK
Bio: A seasoned traveler, Hiro thrives in unknown territories. He enjoys getting lost in new places instead of following the travel guide.
Intention: Hiro is looking for an authentic lunch spot that is not too spicy.

Task: Explore on foot (in street view) looking for a destination that fulfills a certain intention (e.g., lunch, shopping, etc.)

Takeaway: Agents can utilize visual detectors, VLMs and LLMs to iteratively perceive, decide, and interact in the environment.

Hiro is starting a new journey in Hong Kong. He decides to explore without a specific destination in mind, looking for a good local lunch spot with food that's not too spicy...

As depicted in Fig. 6, starting at , Hiro walks down the street and encounters the first intersection. Thanks to the interactive and sensory-rich environment, he can adjust his pose to fetch street views for each possible path. Using VQA on these views, he decides to turn left:

★ *Residential buildings on the left road indicate cozy and family-run local food... A better choice than the others!*

Then, after exploring for a block, he encounters the second intersection where he looks around and decides to turn right:

★ *Looks like there are some local food spots this way...*

After a few steps, Hiro finds *A One Chinese Noodles* 阿一豬扒酸辣米線 using his open-world detector. He looks up its information, ratings, and reviews using our real-world environment which *connects street views to places*. Hiro ultimately decides to pass on it and keep exploring because:

★ *Most reviews mention the spicy pork chop noodles...*

Finally, at the end of this street block , Hiro discovers another lunch spot called *Xintianfa* 新天發. He decides to dine there after reading numerous online reviews praising its authentic cuisine and diverse menu.

Figure 6. Visualization for Hiro's lunch exploration in HK.

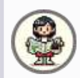
3.4. Collaborative Agents

Humans often work together to solve complex real-world tasks. This collaboration promotes efficiency and effectiveness by decomposing a complex task into simpler sub-tasks, allowing each to be handled by an expert in its domain. Grounded in the world via our platform, V-*IRL* agents can leverage geospatial data and street view imagery to collaborate with other agents as well as with human users.

3.4.1 Agent-Agent Collaboration

As with previous agents, collaborative agents are designed for specific tasks; however, they can handle objectives beyond their expertise through collaboration with each other.

Tourist ENV Map LM LLM CV Vision COL Colab



Name: Ling Age: 25 Loc: NYC/SF/HK
Bio: Ling is a spirited traveler from Taipei who is always eager to explore new cities and cultures. She is unafraid of asking locals for help when she's lost!
Intention: NYC: find gifts for friends back home; go to a famous restaurant. SF: find a store to repair a broken iPhone. HK: try some authentic local food.

Task: (*i*) Ask a nearby Local agent for directions to a specific location. The Locals will preview the route on the map and in streetview and then provide walking directions in natural language, mentioning major intersections and landmarks.

(*ii*) Follow these directions in streetview, and if lost, ask another Local agent for assistance.

Takeaway: Agents can collaborate to solve complex tasks that are beyond their individual expertise.

Ling travels to cities around the world. She seeks out authentic experiences and is always unafraid to ask for help from Locals whenever she finds herself lost...

After obtaining route descriptions from Locals, Ling starts her journey—as shown in Fig. 7. Grounded in our embodied platform, Ling can adjust her pose and identify visual landmarks along the streets using open-world recognition and her map. Correctly recognizing these landmarks helps GPT-4 to make correct decisions about where to turn direction, move forward and stop, as seen in the top two New York City cases in Fig. 7. The success of these decisions made by GPT-4 relies on the real-sensory input for visual grounding and the interactive environment from V-*IRL*.

Nevertheless, Ling may occasionally fail to find the destination. In the bottom left San Francisco example in Fig. 7, Ling passes by the Apple Store because only its stainless steel wall is visible from her viewpoint. In the bottom right Hong Kong example, Ling mistakes another restaurant for her destination and stops prematurely. Fortunately, when she makes these mistakes, Ling can ask another Local agent for new directions and start another round of navigation, which eventually leads her to the destination.

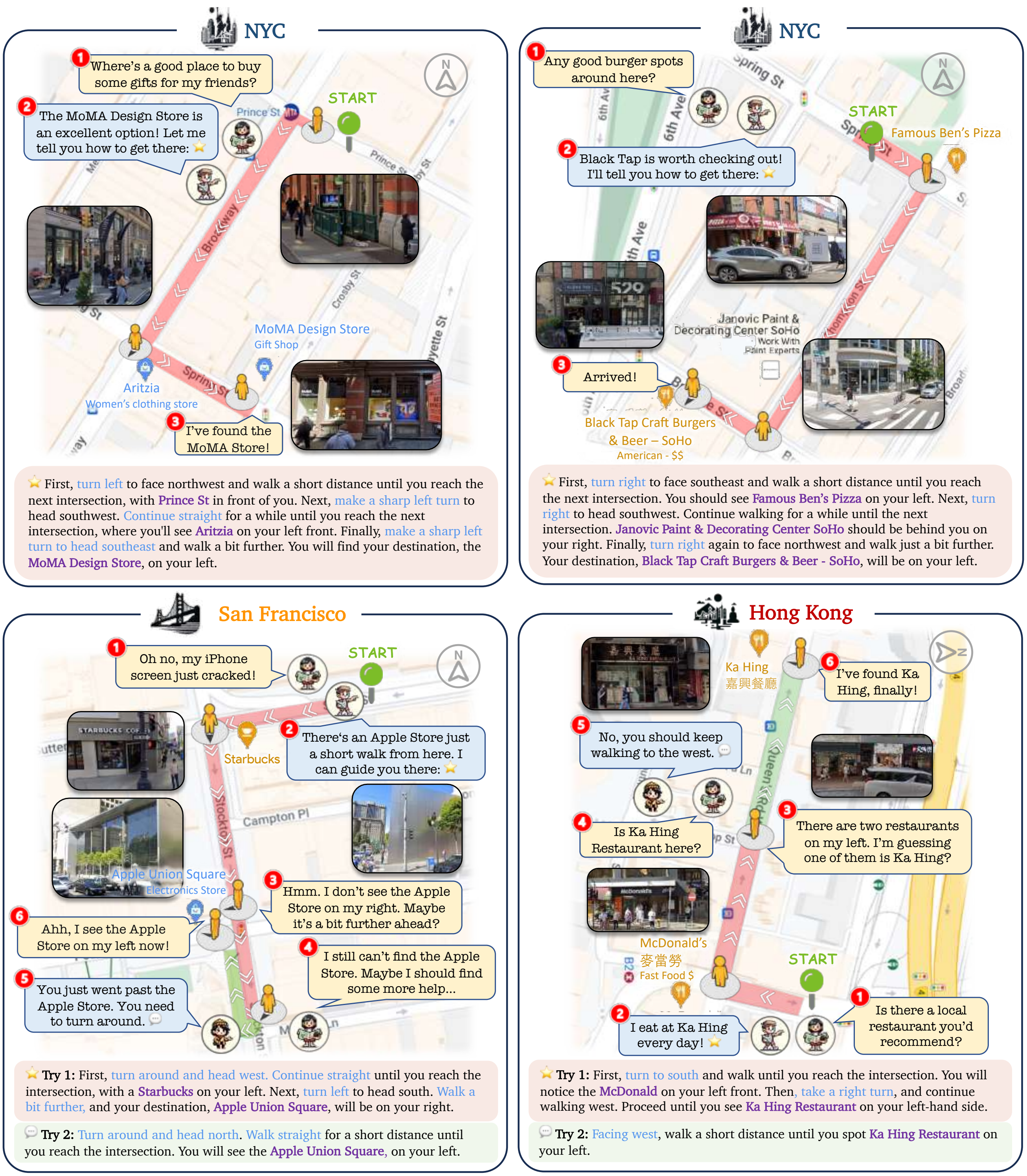


Figure 7. Ling and Local collaboration examples. Trajectories in red and green mean Ling's first and second attempts, respectively.

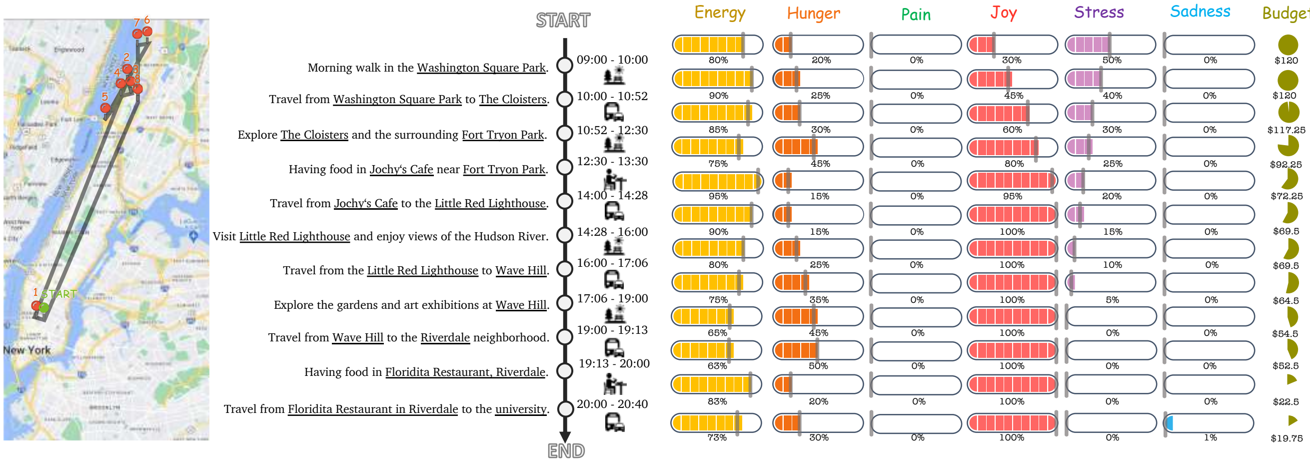


Figure 8. *The Perfect Day Itinerary*: Crafted by Diego, our iterative concierge agent, this schedule is meticulously tailored, accounting for your mental and physical well-being and budget variations as your day unfolds.

Figure 9. Diego traverses regions of interest to find scenic locations to add to your itinerary.

3.4.2 Human-Agent Collaboration

Grounded in the same environment we humans inhabit, V-*IRL* agents can collaborate with and assist real human users.

Interactive Concierge ENV Map LM LLM CV Vision COL Colab

Name: Diego Age: 62 Loc: NYC
Bio: Diego is an expert concierge at a hotel. He's a master at creating intricate itineraries and providing valuable local advice.
Intention: Plan personalized and practical itinerary for customer!

Task: Given a user's location, background, and intention for a day, plan a full itinerary balancing their mental/physical state & budget.

Takeaway: V-*IRL* agents can collaborate with users to solve complex tasks that require understanding the user's internal state.

As a university student in NYC, you are excited to spend a day exploring lesser-known and tranquil places. Your friend recommended Diego, who is known for his professionalism in planning practical and personalized itineraries.

As depicted in Fig. 8, Diego's itinerary is tailored to *your* (the user's) needs. Diego not only considers your physical and mental interoception status, budget for each activity, but also anticipates your status changes and cost when you follow each event. He is able to take into account *real* travel times from the V-*IRL* platform and select suitable destinations by collaborating with another recommendation agent.

In contrast, Fig. 10 shows that a simpler "ungrounded" LLM-only concierge agent is unable to consider the real distance and travel time between locations without access to V-*IRL*, resulting in an impractical itinerary. For example, lacking real geospatial information, the ungrounded concierge allocates only *30 minutes* for travel between the "Brooklyn Botanic Garden" and "Wave Hill" in the Bronx, which actually requires *60–100 minutes* * The hallucinated travel times overlook geospatial realities and result in a plan with excessively distant destinations.

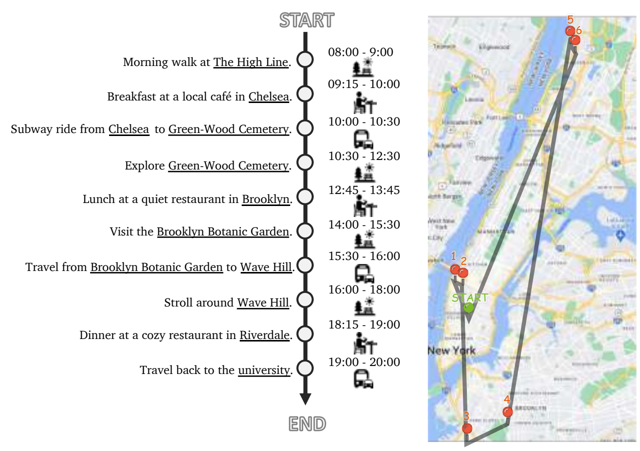


Figure 10. An ungrounded LLM-only concierge agent's itinerary.

Also, as shown in Fig. 11, you can intervene in Diego's

*(per Google Maps).

planning process by adjusting your interoceptive status or by providing verbal feedback. In response, Diego promptly revises his original plan to accommodate your demands, and re-estimates your state changes after his revision.

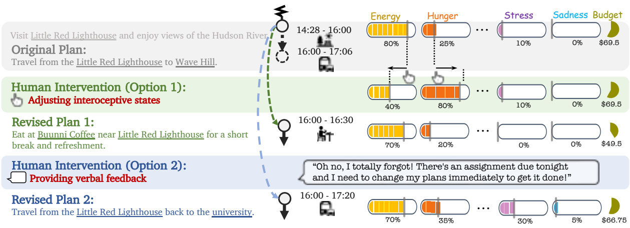


Figure 11. Diego adapts original plan to suit user's intervention.

Finally, using V-*IRL*'s street views and Map, Diego can traverse regions of interest scouting for potential scenic viewpoints for you to visit as shown in Fig. 9. He uses VQA to rate and assess each captured view, and adds the highest rated locations to your itinerary.

4. System Fundamentals

This section introduces our system's core: a platform designed for perception-driven agents that transforms real-world cities around the world into a vast virtual playground where agents can be constructed to solve practical tasks. At its heart, V-*IRL* is comprised of a hierarchical architecture (see Fig. 12). The *platform* lies at the foundation—providing the underlying components and infrastructure for agents to employ. Higher level *capabilities* of CV Perception, LM Reasoning, ENV Action, and COL Collaboration emerge from the platform's components. Finally, *agents* leverage these capabilities and user-defined metadata in task-specific routines to solve tasks.

4.1. Agent Definition

In our system, agent behavior is shaped by user-defined metadata, including a background, an intended goal, and an interoceptive state. The *background* provides the context necessary to instantiate the agent in the real world (location), and to guide its reasoning and decision making (biography). *Intentions* outline agents' purpose within the environment. An agent's *interoceptive state* reflects its internal mental and physical status—varying over time and influencing its behavior. This novel concept to AI agents is crucial for enhancing collaboration with humans (see Sec. 3.4.2).

Concretely, agents are developed by writing task-specific `run()` routines that leverage the various components of our platform and the agent's metadata to solve tasks.

4.2. Platform Components

Next, we delve into the platform components, which provide the infrastructure to instantiate capabilities, execute agent actions, and ground agents in the real world.

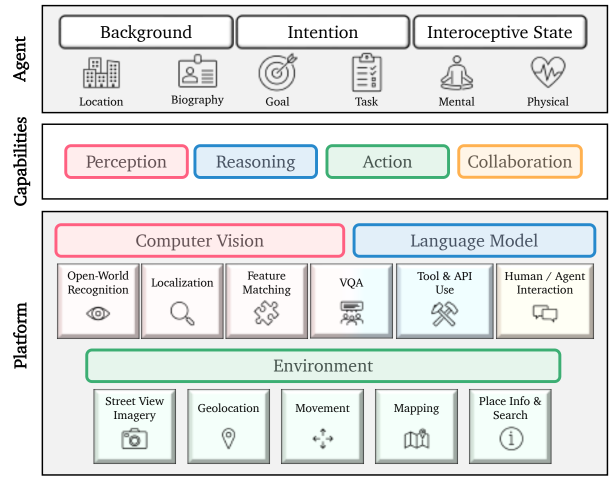


Figure 12. Hierarchical V-*IRL* architecture described in Sec. 4.

4.2.1 Environment (Action)

ENV Environment components are responsible for grounding agents in the world around them: providing a navigable representation of real cities (see Sec. 3.1). Geographic coordinates serve as the link between the world and our virtual representation of it. Leveraging the Google Maps Platform (GMP) [1], V-*IRL* enables agents to access street view imagery, query valid movements, retrieve information about nearby locations, and plan routes. As these coordinates and location information are bound to the real world, they also provide a natural interface with external tools that leverage geolocation—such as real estate APIs (see Sec. 3.2).

4.2.2 Vision (Perception)

CV Perception components enable agents to process the sensory-rich data provided by the environment, especially street view imagery. Pretrained localization models [19] give agents a precise spatial understanding of their environment. This allows RX-399 to identify and count instances of objects, and Hiro to pick out specific businesses to look up with the GMP (Sec. 3.3). While localization models allow for precise interaction with perceptive input, open-world recognition models [29] are more general, and allow agents to detect a wider range of objects in their field of view (*e.g.*, Tourist searches for the Apple Store). Pretrained feature matching models [20] provide an understanding of continuity across views of the same location, and enable agents to identify & deduplicate instances of the same object from different viewpoints (Sec. 3.3). Multimodal models with VQA & Captioning capabilities [18] bridge the perceptual world with natural language, and are essential for integration with reasoning (Sec. 3.3).

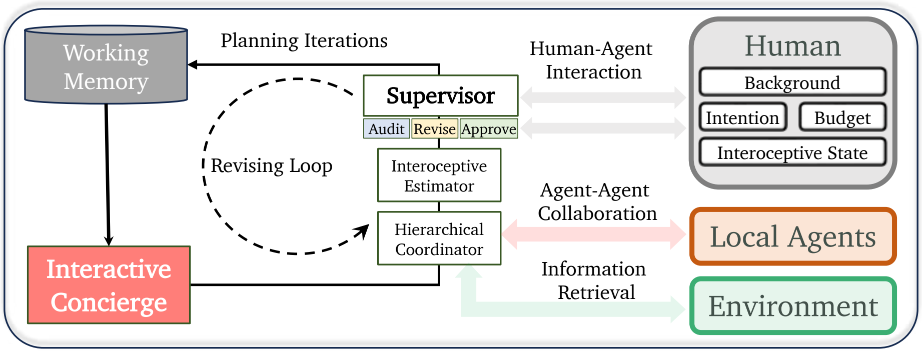


Figure 13. Pipeline overview of interactive concierge agent Diego (Sec. 3.4.2). See pipeline description in Sec. 4.4.

4.2.3 Language (Reasoning & Collaboration)

LM Reasoning components allow decision making based on information from perception and the environment. LLMs such as GPT-4 [27] and Llama 2 [38] interface across various APIs (Sec. 3.2), transforming environmental data and perceptual outputs into actionable insights. They also enable COL Collaboration between agents or with humans through natural language (Sec. 3.4) Custom prompts facilitate this interaction (details in appendix).

4.3. V-*IRL* Capabilities

Our platform's components can be flexibly combined to exhibit a vast array of capabilities. In Sec. 3, we present agents that exhibit increasingly complex behaviors, each requiring more components of the platform. From simple combinations, like the Route Optimizer (Sec. 3.1), to more complex arrangements, like the Tourist (Sec. 3.4.1), our system showcases the versatility and potential of the V-*IRL* framework to be applied to various real-world scenarios. Next we perform a high-level case study of how V-*IRL*'s components are combined to create our most complex agent; in the Appendix (Sec. 7), we delve deeper into the low-level platform details that underpin creating a V-*IRL* agent.

4.4. High-Level System Case Study: Interactive Concierge "Diego"

By studying Diego (Sec. 3.4.2), we illustrate how our platform's components are combined to create complex agents.

Behind Diego's proficiency in developing itineraries is his iterative planning pipeline (depicted in Fig. 13). The process begins with Diego creating an initial draft plan for the first activity using GPT-4, taking into account the user's biography, requirements, and previous activities in working memory. This draft is then meticulously refined. First, a `hierarchical coordination` module retrieves real transportation time and asks a recommendation agent for dining recommendations. Subsequently, an `interoceptive estimation` module evaluates the effect of the proposed activity on the user's mental/physical state and budget.

The crucial final step involves a `supervisor` module, which reviews ("audits") the incoming activity in light of the current user status, remaining budget, and potential interactions (exemplified in Fig. 11). If the `supervisor` deems the plan unsuitable, it initiates revisions. The revised plan is then looped back to the `hierarchical coordinator` and `interoceptive estimator` for reliability, followed by another review from the `supervisor` (see the revising loop in Fig. 13). This iterative process between the `hierarchical coordinator`, the `interoceptive estimator`, and the `supervisor` continues until the `supervisor` approves the activity and adds it to its `working memory`.

After finalizing an activity, Diego proceeds to plan the subsequent activity by repeating this process until the day's itinerary is complete.

5. V-*IRL* Benchmarks

In the previous sections, we illustrate the primary benefit of the V-*IRL* environment: seamless access to first-person street-view imagery and descriptive information about real-world cities across the globe. This *scalable* source of *truly open-world* data can be harnessed to test core component models and agent capabilities. We propose three V-*IRL* benchmarks: two evaluating vision-language models on open-world vision tasks (Secs. 5.2 and 5.3), and one evaluating end-to-end agent performance (Sec. 5.4). Benchmark details are in appendix (Sec. 8).

5.1. Automated Data and Annotation Collection

To allow our V-*IRL* benchmarks to scale globally, we develop an automatic data/annotation construction pipeline instead of crawling and manually annotating limited data. This allows models to be conveniently tested worldwide, provided there is access to Google Street Views [1].

Region Selection. Though our benchmark is feasible across all regions covered by the GMP, we select 14 districts across 12 cities from 6 continents to ensure coverage of a diverse data distribution while keeping inference costs affordable. The detailed locations of these regions are listed in Tab. 2.

Place Types. We collect place information in each region for all 96 places types annotated by GMP†. Our V-*IRL* place: localization, recognition and VQA benchmarks are built upon all or part of these place types.

Vision and Place Data Collection. Within each region, we collect geolocations with available street views, place information, and place-centric images.

Data Cleaning. Though scalable, automated data collection can introduce noise due to the absence of human supervision. To this end, we design three automatic data cleaning strategies: i) *distance-based filtering* to exclude places not easily visible from any street views due to their distance; ii) *human-review filtering* to remove "zombie" places with no reviews which might no longer be valid or relevant; and iii) *CLIP-based filtering* to retain only *place-centric images* with a high CLIP likelihood of being `storefronts`.

Continent	**City**	**District**
Africa	Johannesburg	Rosebank
	Lagos	Surulere
Asia	Mumbai	Khar
	New Delhi	Lajpat Nagar
	Hong Kong	Prince Edward
	Tokyo	Shinjuku
Australia	Melbourne	CBD
	Melbourne	SouthBank
Europe	Milan	Brera
	London	Oxford St
North America	New York City	Chinatown, Manhattan
	New York City	SoHo, Manhattan
	San Francisco	Union Square
South America	Buenos Aires	Monserrat

Table 2. Region list for global V-*IRL* benchmarks.

5.2. V-*IRL* Place: Localization

Every day, humans traverse cities, moving between varied places to fulfill a range of goals, like the Intentional Explorer agent (Sec. 3.3). We assess the performance of vision models on the everyday human activity of *localizing places* using street view imagery and associated place data.

Setups. We modify RX-399 (Sec. 3.3) to traverse polygonal areas while localizing & identifying 20 types of places. We subsample 28 polygonal areas from the 14 districts.

Benchmarked Models. We evaluate three prominent open-world detection models: GroundingDINO [23], GLIP [19] and Owl-ViT [25]. We also implement a straightforward baseline, CLIP (w/ GLIP proposal), which involves reclassifying the categories of GLIP proposals with CLIP [29].

†https://developers.google.com/maps/documentation/places/web-service/supported_types#table1

Evaluation. We evaluate the models based on localization recall, which is quantified as $\frac{N_{\text{tp}}}{N_{\text{tp}}+N_{\text{fn}}}$, where N_{tp} and N_{fn} represents the number of correctly localized places and missed places, respectively.

Matching between Object Proposals and Places. As mentioned in Sec. 5.1, we do not annotate bounding boxes for places on each potential street view image. Such human annotation diverges from our initial motivation of providing plug-and-play and sensor-rich (V-*IRL*) benchmarks. To assign ground truth for each object proposal in this scenario, we develop a simple matching strategy to assign object proposals from street view object detections to nearby places.

As illustrated in Fig. 14, we first project the bounding box of each object proposal onto a frustum in the 3D space, subject to a radius. We then determine if any nearby places fall within this frustum and radius. If any nearby place is found, the closest one is assigned as the *ground truth* for the object proposal. Otherwise, the object proposal is regarded as a *false positive*. When multiple places are inside the frustum, we consider the nearest one as the ground truth since it would likely block the others in the image. *This process is also used in Intentional Explorer agent Hiro to parse object proposals on image to place information.*

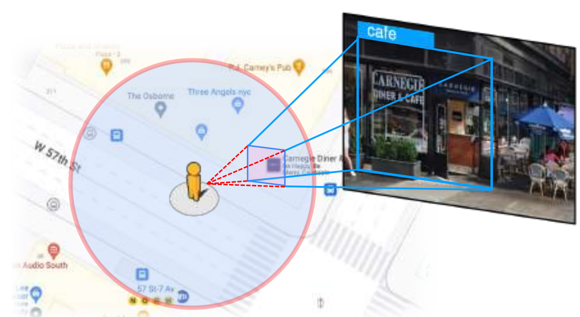


Figure 14. Matching between 2D object proposal and street place.

Results. Tab. 3 shows that open-world detectors like GroundingDINO [23], Owl-ViT [25] and GLIP [19] are biased towards certain place types such as `school`, `cafe`, and `convenience store`, respectively. In contrast, CLIP (w/ GLIP proposal) can identify a broader spectrum of place types. This is mainly caused by the category bias in object detection datasets with a limited vocabulary. Hence, even if detectors like Owl-ViT are initialized with CLIP, their vocabulary space narrows down due to fine-tuning. These results suggest that cascading category-agnostic object proposals to zero-shot recognizers appears promising for "real" open-world localization—especially for less common categories in object detection datasets.

Place Types											AR^{10}	AR^{20}
GroundingDINO [23]	0.0	0.0	0.0	0.0	0.0	7.8	0.0	0.0	16.8	0.0	2.5	1.2
Owl-ViT [25]	0.0	58.0	0.0	0.0	6.4	1.6	0.9	0.0	0.0	0.0	6.7	4.4
GLIP [19]	24.6	0.0	19.2	0.0	0.0	0.0	16.6	0.0	0.0	0.0	6.0	3.7
CLIP [29] (w/ GLIP proposal)	58.5	8.8	28.8	41.2	33.6	23.0	13.0	25.0	0.0	14.5	24.6	20.1

Table 3. Benchmark results on V-*IRL* Place Localization. AR^{10} and AR^{20} denote average recall on subsampled 10 and all 20 place categories, respectively. Full results in Appendix (Sec. 8.1).

5.3. V-*IRL* Place: Recognition and VQA

In contrast to the challenging V-*IRL* place localization task using street view imagery alone, in real life, humans can recognize businesses by taking a closer, place-centric look. We assess existing vision models in this manner on two perception tasks based on place-centric images: i) recognizing specific place types; ii) identifying human intentions via Vision Question Answering (VQA), dubbed "intention VQA".

Setups. For recognition, we assess 10 open-world recognition models at identifying a place's type (from 96 options) using place-centric images (see Tab. 4). For intention VQA, we evaluate 8 multi-modal large language models (MM-LLM) to determine viable human intentions from a four-option multiple-choice. The *V-IRL Place* VQA process is illustrated in Fig. 15, where the candidate and true choices are generated by GPT-4 [27] given the place types and place names corresponding to the image.

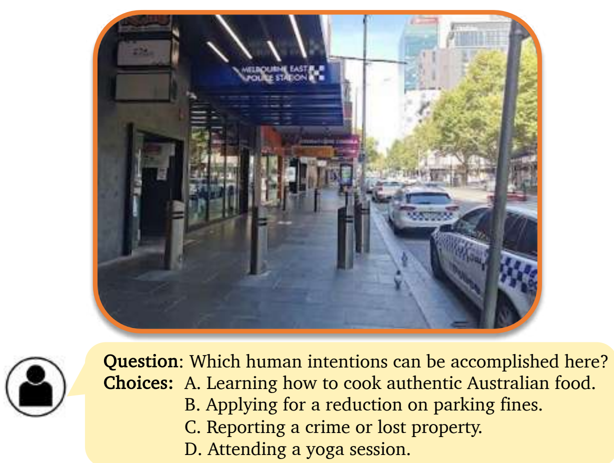


Figure 15. Example of *V-IRL Place* VQA process.

Place-centric Images *vs*. Street View Images. In contrast to the street view imagery utilized in the *V-IRL Place* localization benchmark, the *V-IRL Place* recognition and VQA benchmarks use place-centric images. To illustrate the distinction between these image types, we present examples in Fig. 16. The figure shows that street view images, sourced from the Google Street View database[‡], are taken from the street and encompass a broader view of the surroundings, including multiple buildings and possible occluding objects/vehicles. In contrast, place-centric images, drawn from the Google Place database[§], are taken on foot and focus more closely on the specific place—providing a more concentrated view.

Figure 16. Top row: examples of street view imagery. Bottom row: corresponding place-centric images.

Evaluation. We adopt mean accuracy (mAcc) to evaluate both place recognition and VQA tasks. For place VQA, we follow MMBench [24] to conduct circular evaluation and GPT-assisted answer parsing.

	Model	#Param	mAcc (%)
	V-IRL Place Recognition		
CLIP [29]	ViT-B/32	151M	18.2
CLIP [29]	ViT-L/14	428M	37.2
CLIP [29]	ViT-L/14@336px	428M	41.3
OpenCLIP [7]	ViT-B/32	151M	21.2
OpenCLIP [7]	ViT-L/14	428M	31.0
Eva-02-CLIP [37]	ViT-B/16	150M	19.5
Eva-02-CLIP [37]	ViT-L/14	428M	34.2
Eva-02-CLIP [37]	ViT-L/14@336px	428M	40.7
SigLIP [45]	ViT-B/16	203M	29.5
SigLIP [45]	ViT-L/16@384px	652M	37.3
	V-IRL Place VQA		
MiniGPT-4 [46]	Vicuna-13B-v0	14.0B	3.9
mPLUG-Owl [43]	LLaMA-7B	7.2B	5.5
Shikra [6]	Vicuna-7B	7.2B	10.9
BLIP-2 [18]	$FlanT5_{XXL}$	12.1B	69.6
InstructBLIP [8]	$FlanT5_{XXL}$	12.0B	68.0
LLaVA [22]	Vicuna-13B-v1.3	13.4B	23.5
LLaVA-1.5 [21]	Vicuna-7B-v1.5	7.2B	60.1
LLaVA-1.5 [21]	Vicuna-13B-v1.5	13.4B	61.9

Table 4. Benchmark results on *V-IRL Place* recognition and *V-IRL Place* VQA. Green indicates increased resolution models, while Blue denotes model parameter scaling.

Results. Tab. 4 shows that CLIP (L/14@336px) outperforms even the biggest version of Eva-02-CLIP and SigLIP in the V-*IRL* recognition task, highlighting the high-quality data used to train CLIP [29]. The bottom of the table shows that BLIP2 [18], InstructBLIP [8], and LLaVA-1.5 [21] excel at intention VQA, whereas others struggle. We note that

[‡]https://developers.google.com/maps/documentation/streetview/request-streetview

[§]https://developers.google.com/maps/documentation/places/web-service/photos

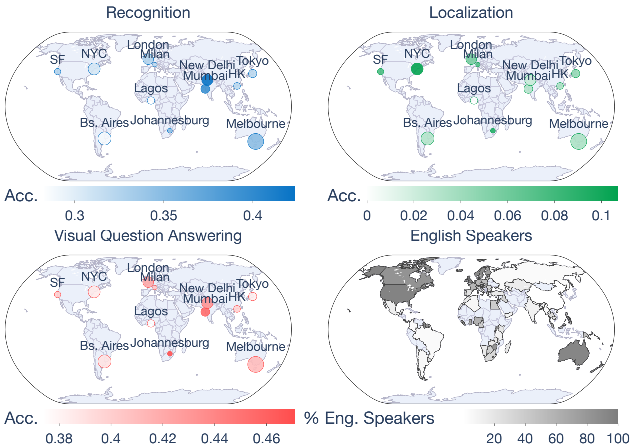


Figure 17. City-level visualization of V-*IRL* benchmark results.

these three top-performing MM-LLMs provide consistent answers in the circular evaluation, while others frequently fail due to inconsistent selections. Moreover, vision models perform better on intention VQA over place-type recognition, suggesting direct prompts about human intention could be more effective for intention-driven tasks. We provide place-type perspective analysis in the Appendix (Sec. 8.2).

5.4. V-*IRL* Vision-Language Navigation

As discussed in Sec. 3.3, Intentional Explorer and Tourist agents require coordination between vision models and language models to accomplish complex tasks. To investigate the effect of various models on end-to-end agent performance, we develop an embodied task that jointly tests vision and language models: Vision-Language Navigation (VLN). In VLN, agents navigate to a desired destination by following textual directions using only raw street-view imagery.

Setup. We adapt the Tourist implementation from Sec. 3.4 and swap its recognition component with the various benchmarked models. These models are used to identify visual landmarks during navigation. Subsequently, GPT-4 [27] predicts the next action according to the recognition results. Navigation instructions are generated using the Local agent. Recent work VELMA [35] attempts to enhance VLN by leveraging LLMs on existing datasets [5, 34]. In contrast, our V-*IRL* VLN benchmark evaluates vision models and their coordination with language models across a global data scale. See more details in Appendix (Sec. 8.3).

Benchmarked methods. Four approaches are evaluated to recognize landmarks during navigation: (*i*) Oracle that searches nearby landmarks with GMP [1]; (*ii*) Zero-shot recognizers CLIP [29] & EVA-CLIP [37]; (*iii*) Multi-modal LLM LLaVA-1.5 [21]; (*iv*) An OCR model [10] to extract text in street views followed by GPT answer parsing. Implementation details provided in Appendix (Sec. 8.3).

Evaluation. We primarily measure navigation success rate (*Success*), defining success as the navigator stopping within

Method		Start	Intersection		Stop	
	Success	Reac	Arr	Reac	Arr	Reac
Oracle (No Vision)	1.0	1.0	1.0	1.0	1.0	1.0
CLIP (B/32) [29]	0.22	1.0	0.86	0.84	0.83	0.22
CLIP (L/14@336px) [29]	0.44	0.83	0.73	0.94	0.67	0.44
EVA-02-CLIP (BigE/14-plus) [37]	0.39	0.89	0.77	0.94	0.72	0.39
EVA-02-CLIP (L/14@336px) [37]	0.22	1.0	0.82	0.83	0.78	0.22
LLaVA-1.5-13B [21]	0.11	0.61	0.55	1.0	0.56	0.11
PP-OCR [10] (+ GPT3.5)	0.28	0.89	0.73	0.94	0.72	0.28

Table 5. Results on V-*IRL* VLN-mini. We test various CLIP-based models, MM LLM , and OCR model with GPT postprocessing.

25 meters of the destination. In addition, as navigation success is mainly influenced by the agent's actions at key positions (*i.e.*, start positions, intersections and stop positions), we also evaluate the arrival ratio (*Arr*) and reaction accuracy (*Reac*) for each route. *Arr* denotes the percentage of key positions reached, while *Reac* measures the accuracy of the agent's action predictions at these key positions. To save GPT-4 resources, we mainly compare vision modules on a 10% mini-set comprising 18 routes from 9 regions. See Appendix (Sec. 8.3) for full-set results with CLIP and Oracle.

Results. Table 5 shows that, with oracle landmark information, powerful LLMs can impressively comprehend navigation instructions and thus make accurate decisions. However, when relying on vision models to fetch landmark information from street views, the success rate drops dramatically—suggesting that the perception of vision models is noisy and misguides LLMs' decision-making. Among these recognizers, larger variants of CLIP [29] and EVA-02-CLIP [37] perform better, highlighting the benefits of model scaling. LLaVA-1.5 [21] shows inferior performance with CLIP (L/14@336px) as its vision encoder, possibly due to the alignment tax [27] introduced during instruction tuning. Further, PP-OCR [10] (+ GPT-3.5) achieves a 28% success rate, signifying that OCR is crucial for visual landmark recognition.

5.5. Geographic Diversity

Spanning 12 cities across the globe, our V-*IRL* benchmarks provide an opportunity to analyze the inherent model biases across different regions. As depicted in Fig. 17, vision models demonstrate subpar performance on all three benchmark tasks in Lagos, Tokyo, Hong Kong, and Buenos Aires. Vision models might struggle in Lagos due to its non-traditional street views relative to more developed cities (see street views in Fig. 1). For cities like Tokyo, Hong Kong, and Buenos Aires, an intriguing observation is their primary use of non-English languages in street views, as shown in Fig. 17 bottom right ¶ and Fig. 1. This suggests that existing vision models may face challenges when deployed in non-English-dominant countries.

¶Source: https://en.wikipedia.org/wiki/List_of_countries_by_English-speaking_population

References

[1] Google map platform. https://mapsplatform.google.com/. 8, 9, 12

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 2

[3] Yuki M Asano, Christian Rupprecht, Andrew Zisserman, and Andrea Vedaldi. Pass: An imagenet replacement for self-supervised pretraining without humans. *arXiv preprint arXiv:2109.13228*, 2021. 2

[4] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, pages 287–318. PMLR, 2023. 1

[5] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547, 2019. 12

[6] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 11

[7] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023. 11

[8] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 11, 1

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[10] Y Du, C Li, R Guo, X Yin, W Liu, J Zhou, Y Bai, Z Yu, Y Yang, Q Dang, et al. Pp-ocr: A practical ultra lightweight ocr system. arxiv 2020. *arXiv preprint arXiv:2009.09941*. 12

[11] Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, et al. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*, 2020. 4

[12] Abhimanyu Dubey, Vignesh Ramanathan, Alex Pentland, and Dhruv Mahajan. Adaptive methods for real-world domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14340–14349, 2021. 2

[13] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022. 1

[14] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022. 1

[15] Heinrich Küttler, Nantas Nardelli, Alexander Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The nethack learning environment. *Advances in Neural Information Processing Systems*, 33:7671–7684, 2020. 2

[16] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 2

[17] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 2

[18] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 8, 11

[19] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 2, 8, 10, 11

[20] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. *arXiv preprint arXiv:2306.13643*, 2023. 8

[21] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 11, 12, 4

[22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 11, 1

[23] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 10, 11

[24] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023. 11

[25] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pages 728–755. Springer, 2022. 10, 11

[26] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. 2

[27] R OpenAI. Gpt-4 technical report. *arXiv*, pages 2303–08774, 2023. 2, 9, 11, 12

[28] Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023. 2

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 8, 10, 11, 12

[30] Vikram V Ramaswamy, Sing Yu Lin, Dora Zhao, Aaron Bryan Adcock, Laurens van der Maaten, Deepti Ghadiyaram, and Olga Russakovsky. Geode: a geographically diverse evaluation dataset for object recognition. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 2

[31] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019. 2

[32] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023. 2

[33] John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, et al. Chatgpt: Optimizing language models for dialogue. *OpenAI blog*, 2022. 4

[34] Raphael Schumann and Stefan Riezler. Generating landmark navigation instructions from maps as a graph-to-text problem. *arXiv preprint arXiv:2012.15329*, 2020. 12

[35] Raphael Schumann, Wanrong Zhu, Weixi Feng, Tsu-Jui Fu, Stefan Riezler, and William Yang Wang. Velma: Verbalization embodiment of llm agents for vision and language navigation in street view. *arXiv preprint arXiv:2307.06082*, 2023. 12, 4

[36] Significant Gravitas. AutoGPT. 2

[37] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 11, 12

[38] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 9

[39] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023. 1, 2

[40] Michael Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(2):115–152, 1995. 2

[41] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023. 1, 3

[42] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079, 2018. 2

[43] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 11

[44] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 2

[45] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*, 2023. 11

[46] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 11

V-IRL: Grounding Virtual Intelligence in Real Life

Supplementary Material

6. Appendix Outline

In these supplementary materials, we provide additional details for our V-*IRL* platform, including:

- A low-level case study of Intentional Explorer agent Hiro, delving into implementation details of our system such as LLM prompts (Sec. 7);
- More detailed setups and results for our V-*IRL* benchmarks (Sec. 8).

7. Low-Level System Case Study: Intentional Explorer "Hiro"

This section delves deeper into the low-level implementation details of the Intentional Explorer agent "Hiro" (Sec. 3.3), focusing on the prompts utilized to interact with various parts of our system. Concretely, we present the prompts in four subparts: *identifying a type of place to search using the user-defined intention* (Sec. 7.1), *selecting appropriate roads* (Sec. 7.2), *summarizing reviews of places* (Sec. 7.3), and *making action decisions* (Sec. 7.4). These four components jointly enable Hiro to explore in our interactive embodied environment driven by his initial intention.

7.1. Intention to Place Type

Starting with a user-defined agent intention, Hiro first determines the type of place that could fulfil this intention using GPT-4 and the following prompt:

```
[Role]
You are PlaceSuggesterGPT, an expert
in recommending types of places
based on user-specified intentions.

[Task Description]
Given a user-specified intention,
determine the type of "place"
one should seek to fulfill the
intention. Your response should
be in the following JSON format:
{"place": "Desired Place Type"}

[Example]
Input: "Intention: <buy a book>"
Output: {"place": "bookstore"}

[Input]
Intention: <{agent_intention}>

[Output]
Your recommended place type based on
the user-specified intention, in the
required JSON format:
```

Using this prompt with the intention

> *Hiro is hungry and looking for a place where he can try some good local food. He cannot handle spicy food.*

returns the result

```
{"place": "restaurant"}.
```

The identified place type (here, `restaurant`) is extracted and set as the target category for Hiro's open-world detector during his exploration.

7.2. Road Selection

Whenever Hiro is at a crossroads, he determines the best road to follow using his multi-modal LLM and GPT-4. The primary goal of the road selection process is to identify the road most likely to lead to the desired place type that aligns Hiro's intention. First, Hiro fetches the street view towards each potential road using the V-*IRL* environment. Then he utilizes his multi-modal LLM (such as InstructBLIP [8] or LLaVA [22]) to generate captions for each road using the following prompt:

```
I am looking for a {place_type}.
Please detail information that might
be helpful for me along this road:
```

Captions for each road are then formatted in the style of

```
{road_idx}: {road_description}
```

and concatenated to form `all_road_descriptions`. These road captions, along with Hiro's user-defined intention, are then fed into GPT-4 to determine the most promising road to follow using the following prompt:

```
[Role]
You are PathSelectorGPT, an expert
in choosing the optimal road from
multiple candidates based on a
user-specified intention.

[Task Description]
Given an intention, the road
previously traveled, and
descriptions of available candidate
roads, select the best road from the
crossroad. Your response must be in
the following JSON format:
```

```
{"idx": "Selected road index",
"reason": "Justification for your
selection"}

[Example]
For the intention "find a grocery
store", the road previously traveled
as "1", and with candidates "2:
Leads to residential area, 3: Leads
to a shopping district", the output
might be: {"idx": "3", "reason":
"Road 3 leads to a shopping district
which is more likely to have a
grocery store."}

[Input]
User Intention: <{agent_intention}>
Road Descriptions:
{all_road_descriptions}
Previously Traveled Road: Road
{from_road_idx}

[Output]
Your chosen road index and the
reasoning behind your selection,
in the required JSON format:
```

We design such a two-stage captioning and decision-making pipeline for road selection because Multi-modal LLMs cannot process multiple images simultaneously. However, with the recent advancements of GPT-4V, it may be possible to perform road selection using several road images with a single prompt at once. Empirical findings suggest that GPT-4V yields more reasonable choices with the following prompt:

```
[Role]
You are PathSelectorGPT, an expert
in choosing the optimal road from
multiple road images according to a
user-specified intention.

[Task Description]
Given a set of road images, select
the best road from the crossroad.
Your answer must be in the following
JSON format:
{"idx": "Selected road index (start
by 0)", "reason": "Justification
for your selection"}

[Input]
User Intention: <{agent_intention}>

[Output]
Please answer with the road index
and the reasoning behind your
selection, in the required JSON
format:
```

An example road selection response for the first crossroad selection in Fig. 6 is as follows:

"idx": "0", "reason": "Choosing road 0 promises a genuine taste of local cuisine in a less commercialized setting. It's likely to have family-run eateries where I can request non-spicy dishes and savor authentic flavors. This road offers a tranquil dining atmosphere, allowing for a more engaged and leisurely culinary exploration among the locals."

7.3. Summarize Place Reviews

When Hiro discovers a place in the street view imagery, he retrieves its corresponding name and Google reviews from the V-*IRL* environment. Then, he summarizes the reviews into a place overview (to aid in decision-making) using the following prompt:

```
[Role]
You are SummarizeGPT, skilled at
condensing multiple reviews into a
concise overview of a location.

[Task Description]
Given multiple reviews with ratings,
craft a brief overview of the place.
Your response should be in the
following JSON format:
{"summarization": "Concise
description (limited to 80 words)"}

[Example]
For reviews "Great ambiance but
average food (Rating: 3)" and
"Loved the decor, food could be
better (Rating: 3.5)", the output
might be:
{"summarization": "The place
boasts great ambiance and decor,
but the food quality receives mixed
reviews."}

[Input]
Reviews: {all_reviews}

[Output]
Your concise overview (max 80 words)
based on the provided reviews, in
the prescribed JSON format:
```

7.4. Action Decision

After obtaining the overview of the identified place, Hiro decides to visit the place or keep exploration using GPT-4 and the following prompt:

```
[Role]
You are ActionSelectorGPT,
proficient in choosing the most
appropriate action based on a
user's background, intention, and
an overview of a place.

[Task Description]
Evaluate the provided user
background, intention, and place
overview to select the most suitable
action from the list.  Your response
should be in the following JSON
format:
{"action":  "Selected Action",
"reason":  "Justification for your
choice"}

Possible actions:
- enter_place():  Enter the
designated place.
- continue():  Continue searching
for another appropriate place.

[Example]
For the background "loves historical
sites", intention "discover local
history", and place overview
"This is a 200-year-old preserved
mansion", the output might be:
"action":  "enter_place()",
"reason":  "The historical mansion
aligns with the user's interest in
historical sites."

[Input]
User Background:  <{background}>
User Intention:  <{intention}>
Place Overview:  <{place_intro}>

[Output]
Your chosen action and the rationale
behind your decision in the
prescribed JSON format:
```

Hiro's exploration will continue if he decides to `continue()` and will terminate if he opts for `enter_place()`.

8. V-*IRL* Benchmarks: Details

8.1. V-*IRL* Places: Localization (Details)

All category results. Due to page limit of the main paper, we only present the results of 10 categories in Tab. 3. Here, we present the place recall for all 20 categories in Fig. 18.

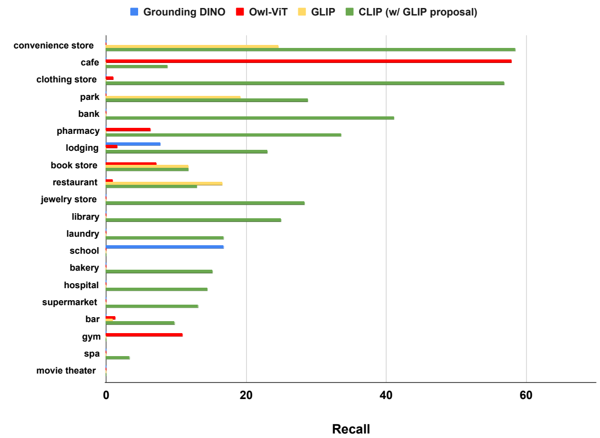


Figure 18. Recalls in *V-IRL Place* localization

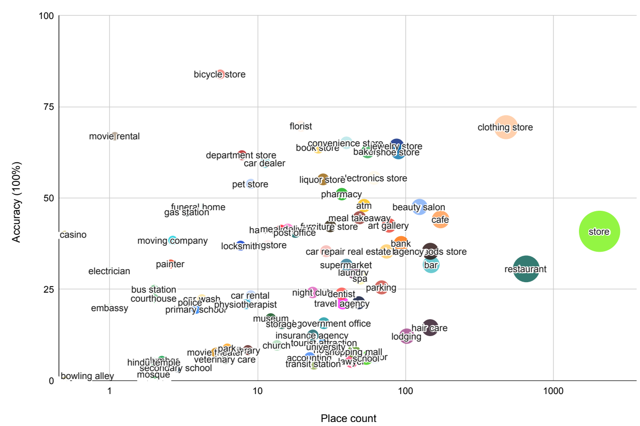


Figure 19. Category-wise accuracy and numbers for V-*IRL* Place Recognition benchmark.

Example illustrations. To facilitate the understanding of *V-IRL Place* localization benchmark, we present some examples of CLIP (w/ GLIP proposals) in Fig. 21.

8.2. V-*IRL* Places: Recognition and VQA (Details)

Place types performance for recognition. In Figure 19, we present the averaged accuracy for each place type across 10 benchmarked vision models. The size and the x-axis position of each bubble correspond to the number of places within each type. A clear trend emerges: accuracy tends to correlate with the frequency. Common categories such as `clothing store`, `cafe` exhibit higher accuracy, whereas vision models often struggle with infrequent place types like `bowling alley` or `mosque`.

Place types performance for VQA. The place types performance of the V-*IRL* place VQA in Fig. 20 further verifies the correlation between accuracy and frequency from a human intention perspective. The top-10 categories are closely aligned with the most common human activities, purchasing and dining. In contrast, the bottom-10 place types relate to places that are less frequently encountered and serve a more diverse purpose, such as `mosque`, `plumber` and `embassy`.

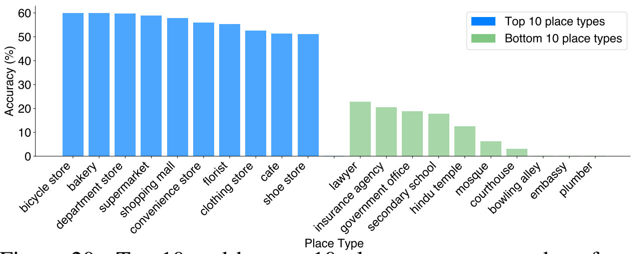


Figure 20. Top-10 and bottom-10 place types averaged on four vision models of V-*IRL* Place VQA.

8.3. V-*IRL* Vision-Language Navigation (Details)

Navigation pipeline. The navigation pipeline of our V-*IRL* VLN benchmark is similar to [35], leveraging vision models, the map, and LLMs. At each position, we start by capturing eight street views around the agent, corresponding to `front`, `left front`, `left`, `left behind`, `behind`, `right behind`, `right` and `right front`. Vision models use these street views to identify landmarks mentioned in route descriptions, which are then verbalized as *landmark observations*. Also, intersection information is retrieved from the mover to formulate an *intersection observation*. LLMs play a crucial role in processing landmark & intersection observations along with the agent's previous working history to determine the next action. After each action, current observations and actions are stored into the agent's working history. This auto-regressive process continues until the agent decides to `stop`.

In contrast to [35], our benchmark offers greater scalability through the worldwide V-*IRL* platform and an automated data collection pipeline, as opposed to the manual annotation of a specific region. Furthermore, our benchmark emphasizes the analysis of the *vision* component in the VLN pipeline, as opposed to [35], which aims to enhance performance on existing VLN datasets using LLMs.

Implementation Details. Here, we introduce the implementation details for LLaVA-1.5 [21] and PP-OCR [11] (+ GPT-3.5). For LLaVA-1.5 [21], we transform the landmark recognition task to a *multiple choice VQA* problem, asking

```
Which of the following landmarks
can be identified with a high
degree of confidence?
```

The VQA options include all potential landmarks mentioned in the route description, along with a "`None of above`" choice. The model's response to this question is then parsed as the landmark observation.

For PP-OCR [11] (+ GPT-3.5), we first extract all recognized text using PP-OCR [11] for each street view image. Then, GPT-3.5 [33] determines the presence of each landmark in this street view image, jointly considering the OCR text and landmark name.

Full set results. Apart from the mini-set results presented in Sec. 5.4, we also provide the full set results of Oracle and CLIP (L/14@336px) in Tab. 6. The Oracle results, interestingly, do not achieve a 100% success rate, due to incorrect decisions made by the LLM at stop positions. This is evidenced by the high arrival ratio and low reaction accuracy at stop positions. Empirically, we observe that the LLM occasionally decides to keep moving, despite clear destination indications in the observations.

When we substitute the map in oracle with the CLIP model to gather landmark observations from street view imagery, we observe a significant drop in the success rate, due to the inevitable model prediction errors. To improve the success rate in VLN, we can focus on two important factors: (i) designing better vision models; (ii) developing LLMs and prompt techniques that are robust to vision-related noise. Especially, our empirical findings suggest that sophisticated prompt designs significantly improve the robustness of LLMs to visual observation noise.

Method		Start	Intersection		Stop	
	Success	Reac	Arr	Reac	Arr	Reac
Oracle (No Vision)	0.88	1.0	0.95	0.99	0.96	0.88
CLIP (L/14@336px)	0.22	0.84	0.66	0.90	0.61	0.22

Table 6. Results of V-*IRL* VLN-full.

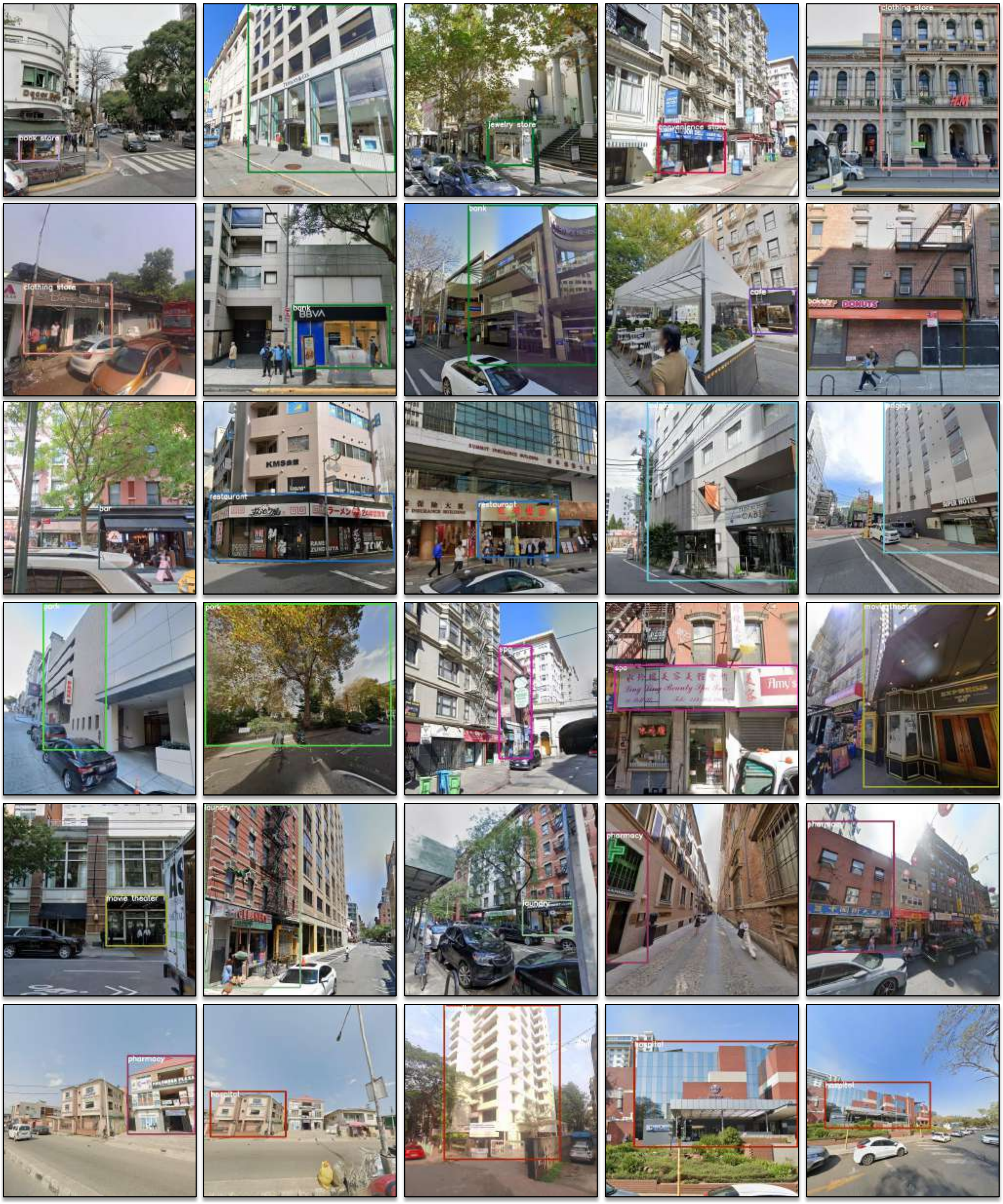


Figure 21. Samples of *V-IRL Place* localization using CLIP (w/ GLIP proposals).