# Geometry, Physics, and Semantics from Videos: Making Sense of a Dynamic World through Perception, Exploration, Demonstrations, and Explanations

Research Statement
Katerina Fragkiadaki
June 1st 2025

My goal is to build intelligent machines that can perceive, reason, plan, act, and learn based on sensory input. A central theme of my work is exploring how physics-based perceptual representations—grounded in 3D structure and motion—can enhance perception, reasoning, planning, action, and learning.

One line of work that exemplifies this vision began with our early efforts to **develop end-to-end trainable 3D feature representations for mobile perception—representations that persist over time and account for camera motion** (31; 10; 9; 19; 24). At a time when most computer vision methods grounded their outputs directly in 2D image space, we introduced architectures that mapped 2D image features into persistent 3D or bird's-eye-view (BEV) coordinate frames. These frames moved with the agent and effectively compensated for its motion, all in a fully differentiable manner. **Rather than extracting explicit 3D representations** (such as meshes or voxel grids), **our key insight was to use the 3D structure as an inductive bias**, allowing the end-to-end learning of feature representations optimized for tasks such as object detection, tracking, and self-supervised objectives such as novel view synthesis. These 3D representations have since become the standard in autonomous driving, where multi-camera inputs are fused into BEV maps jointly trained for multiple downstream tasks.

Building on this foundation, we have recently developed architectures for **3D robot manipulation policies** that reason directly in 3D space using rich 3D scene feature representations (see Section 1). These models achieve strong performance on several benchmarks (5; 35; 15; 6), combining 3D scene understanding with generative formulations to effectively capture multimodality in demonstration learning, enabling generalization across tasks and viewpoints.

Beyond imitation learning, we extended this idea to show how **geometry- and physics-aware generative models can be used to *search* for high-reward behaviors**, even when the reward function is unknown at training time (38; 4; 37). Unlike demonstration-based approaches, our method enables **the same generative model to be guided at test time to produce reward-maximizing actions** for a wide variety of tasks—regardless of whether the reward functions are differentiable or not—without the need for additional training (see Section 2).

We investigated representations for motion estimation in video (Section 3). While prior methods predominantly relied on frame-to-frame optical flow, we introduced **multi-frame point trajectories that persist through point occlusions** (8; 39). To extract these trajectories, we developed **learned iterative neural networks trained on synthetic data** from simulation environments. This work marked a paradigm shift—point trajectories have since become the default motion representation, significantly outperforming optical flow in both accuracy and expressiveness. For example, they have been widely adopted as goal representations in imitation learning for robotics.

Recognizing the value for understanding physics in both fine-grained perception and robot interaction, we have focused on enhancing the usability and scalability of physics engines. This includes the development of *GENESIS*, a more accessible and powerful simulation platform, as well as efforts to **scale and automate data generation in the physics engine using generative models and reality-to-simulation translation** (14; 34) (see Section 4). In particular, we are advancing methods to simulate visible scenes and their dynamics—which we see as a natural and essential path for scaling simulation data—through techniques such as 3D point tracking in videos (39), amodal video completion in time and across viewpoints via analysis-by-synthesis and generative priors (3; 4), and inferring a structured scene from a single image by jointly generating a set of 3D meshes with compositional diffusion models (21).

Motivated by the desire to develop experienced-based dynamic scene understanding, we have been exploring the role of **explicit memory retrieval and attention** in vision-language-action reasoning (Section 5). We have shown that such models enable effective **personalization** by storing and reusing explicit user routines (28), and proposed methods to **optimize memory structure** in order to maximize task success (26). Additionally, we have extended pre-trained 2D vision-language models (VLMs) to operate on both 2D image-centric and 3D world-centric representations (11; 12). By introducing 3D feature maps as a form of short-term memory, we significantly improved the ability of these models to reason over long egocentric video sequences, scenarios that would otherwise overwhelm the context window due to the redundancy and frame-by-frame repetition inherent in 2D representations. Our ongoing work focuses on developing models that **reason step-by-step by grounding their thoughts in visual input** (25) as well as their past memories—learning to actively move the camera to gain better observations as part of their reasoning process, identifying which memories to retrieve to inform decisions, and improving memory consolidation to boost overall performance.

Additional contributions, achievements, and recognition include:

1. DARPA Young Faculty Award, AFOSR Young Investigator Award, NSF Career, numerous industry gifts and awards (Google, Amazon, Sony, UPMC, TRI, NVIDIA)

2. JPMorgan Chase Career Development Professorship

3. Started the AI4ALL summer camp in CMU.

4. Evidence of recognition through numerous invited talks in workshops and symposiums, e.g., 6 invited talks in CVPR 2025 workshops, 2 invited talks in workshops in RSS 2025, 2 invited talks in workshops for ICRA 2025, 5 invited talks at workshops in ICCV 2025, Future of ML Symposium in ISTA, Vienna 2024, keynote in MVA 2025, keynote in 3DV and BMVC 2021.

5. I have been Program Chair for ICLR 2024, Associate Editor in TPAMI, Senior Area Chair in ICLR 2025 and CVPR 2025, Area chair in all NeurIPS, ICML, ICCV, ECCV, CoRL conferences

# 1 3D Reasoning in Robot Action Prediction

Our lab has pioneered a new class of 3D robot manipulation policies that reason directly in 3D space, setting a new standard for generalization and control in complex manipulation tasks. Unlike prior methods that operate in 2D image space and predict 3D end-effector poses or body joint trajectories, **our approach represents both actions and visual observations as tokens embedded in a shared 3D coordinate frame. These tokens are fused using relative 3D positional encodings, allowing the policy to reason jointly about scene geometry, object layout, and task objectives.**

Our *Act3D* (5) paper introduced a coarse-to-fine 3D action inference framework and demonstrated strong generalization to unseen camera viewpoints—a setting where most existing methods fail. Building on this, *ChainedDiffuser* (35) proposed a hierarchical policy architecture that first predicts key interaction poses and then synthesizes full motion trajectories to reach them. This enabled robust execution of long-horizon tasks in contact-rich environments, outperforming traditional motion planners in constrained manipulation settings.

In more recent work, we introduced the *3D Diffuser Actor* (15), which **formulates action inference as a generative denoising process in a joint action-vision 3D space.** This model introduces **position-aware attentions, where the positional encodings of action tokens are updated at each denoising step** based on intermediate predictions, allowing for precise spatial reasoning at each inference step. Our latest advance, *3D*

*Flow Actor* (6), extends generative 3D formulations to bimanual manipulation, and achieves state-of-the-art performance on both simulated and real-world benchmarks (Figure 1).

Together, these contributions have established a new paradigm for spatially grounded, instruction-conditioned robot policy learning. Our 3D diffusion policies are now widely adopted as sample-efficient action decoders, capable of learning from just a handful of demonstrations, and have become a preferred choice for labs seeking practical, generalizable manipulation systems.

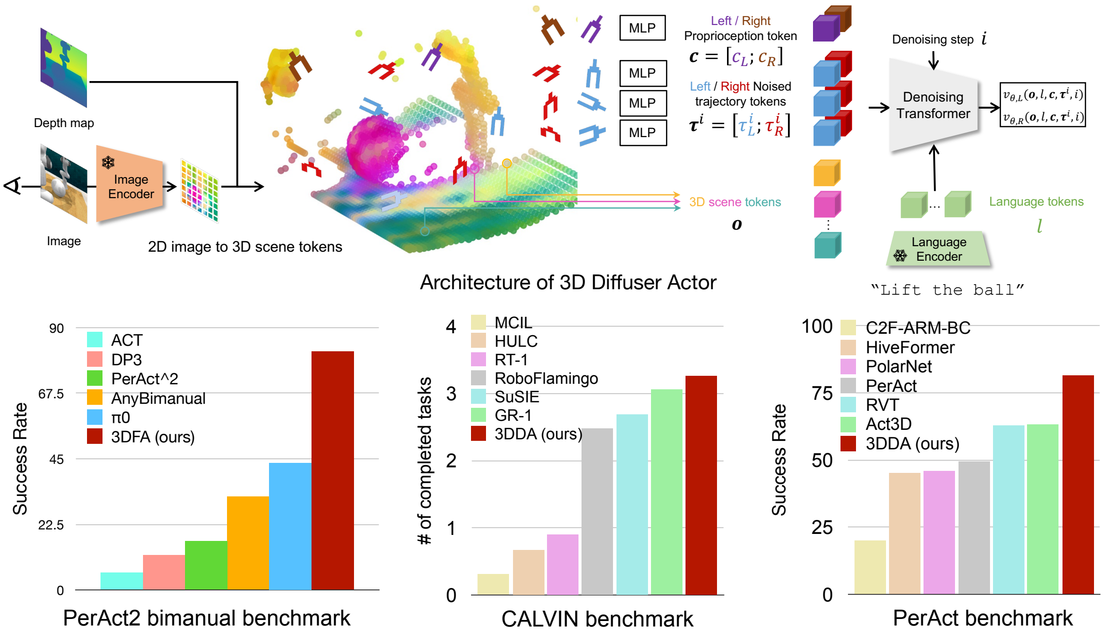


Figure 1: **Robot Manipulation policies with 3D Reasoning.** Our works 3D Diffuser Actor (15) and 3D Flow Actor (6) show state-of-the-art performance of RLbench PerAct(30) and PerAct2 (7) benchmarks, and the CALVIN (22) benchmarks.

## 2 Planning with Generative Visuo-Motor Models

Recent advances in generative modeling present a transformative opportunity for agent exploration and reinforcement learning. Traditional model-based control relies on learning next-state prediction functions and optimizing action sequences over time, a process that can quickly amplify errors in the learned dynamics. In contrast, generative models can learn distributions over entire action trajectories—or joint action-state trajectories—capturing the behavior of both the robot and its environment. This fully generative formulation enables more robust and efficient search for high-reward actions, providing a powerful alternative to conventional planning approaches.

In our work **Diffusion-ES (Diffusion Evolutionary Search)**, we introduced a method that integrates generative models of robot action trajectories—such as vehicle waypoint trajectories—with traditional evolutionary search for planning in autonomous driving. A key innovation is a **learned mutation operator**: rather than adding random noise to action sequences, we inject noise and then denoise it using a learned trajectory diffusion model. This ensures that mutated samples stay on the manifold of plausible behaviors, yielding a data-driven planner capable of efficiently optimizing even non-differentiable reward functions.

We demonstrated that Diffusion-ES achieves state-of-the-art performance in the nuPlan driving benchmark (2) as a real-time online planner. Furthermore, we showed that it can follow natural language instructions without additional training by using large language models (LLMs) to translate language instructions into constraints that guide the diffusion-ES planning process.

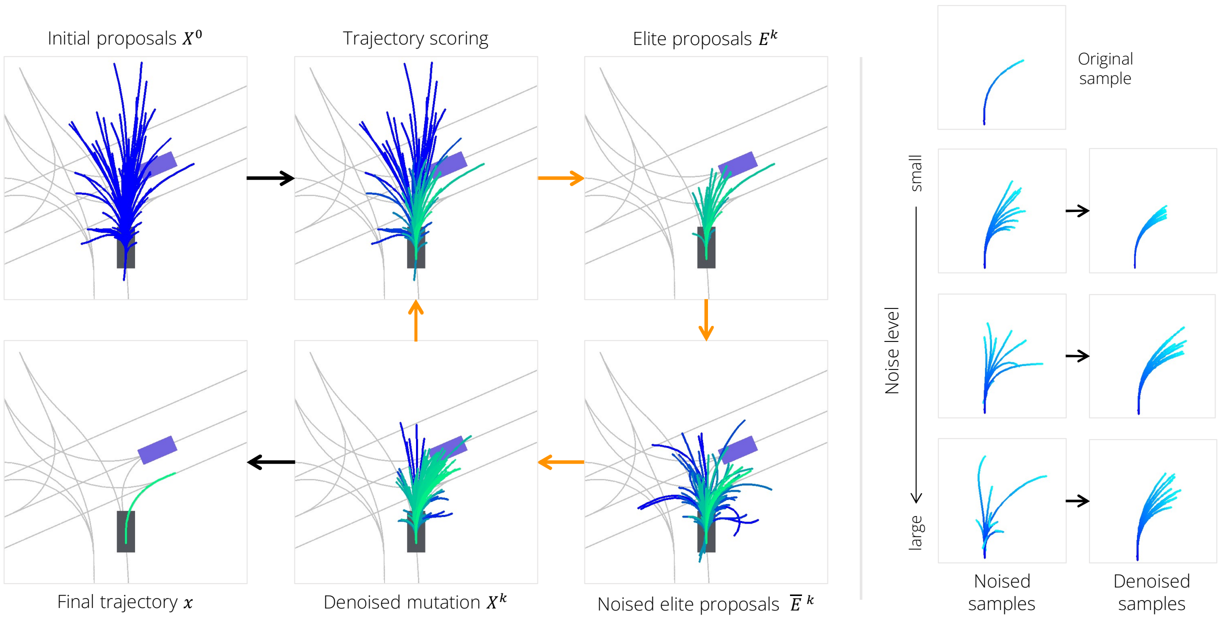


Figure 2: *Left:* Generative evolutionary search with Diffusion-ES. Trajectories are colored by rewards (blue is low, green is high). *Right:* We visualize the mutations for varying noise levels. Color denotes timestep along trajectory. While noise perturbations alone can lead to unrealistic trajectories, denoising helps project samples back onto the trajectory data manifold.

More recently, in our work on **Generative 3D Particle World Models** (4), we extended this idea to **jointly model agent action trajectories and 3D object particle trajectories of manipulated deformable objects** using a single diffusion model. This unified representation enables **goal-directed planning via guided denoising**—inferring actions that satisfy differentiable object-centric constraints (e.g., achieving a target object position or deformation), while staying consistent with learned priors over realistic interactions. This represents a step toward physically grounded, general-purpose planning from vision and interaction data.

# 3 Motion Understanding from Video

**Tracking Any Point Through Time in a Video:** Tracking objects, parts and points in video is crucial for action recognition and robot imitation. Traditionally, motion estimation in video has been framed as an optical flow problem: predicting per-pixel motion vectors between consecutive frames. However, this formulation fails under occlusions, where pixel-wise correspondences are undefined and information is lost.

Our paper, *Particle Video Revisited: Tracking Through Occlusions Using Point Trajectories* (8), introduced a paradigm shift: instead of estimating motion as frame-to-frame optical flow, we model each point with a full trajectory across time, enabling robust tracking through occlusions. The key contributions of this work are:

- **Trajectory-based representation:** We replace pairwise flow vectors with multi-frame point trajectories that persist across occlusions.
- **Iterative learned estimator:** We introduce a neural architecture that iteratively updates point trajectories using a learned refinement operator.

- **Sim-to-real generalization:** Our model is trained on synthetic data—where point supervision is easily obtained in a graphics engine—and then deployed without any finetuning on real-world video.

- **State-of-the-art performance:** Our method outperforms baselines trained directly on real videos via self-supervision, demonstrating the value of synthetic training for long-range correspondence.

This approach marked a paradigm shift in multiframe video processing, providing superior pixel tracking and correspondence compared to previous methods that mostly focused on self-supervising temporal correspondences on large amounts of real video. This multi-frame tracking approach has become a powerful tool for robot learning, particularly in representing manipulation subgoals, since they provide a natural abstraction of intended interactions.

Our latest work, *TAPIP3D* (39), extends these ideas to **persistent point tracking in the 3D world space.** It significantly improves accuracy over both 2D and 3D baselines—particularly when depth is available—and supports both camera-centric and world-coordinate inference. Our experiments show that compensating for camera motion is crucial for accurate, long-horizon tracking.

**From Videos to Complete 4D Scene Reconstructions** While point tracking captures motion of points visible in some frame in the video, it does not account for completely occluded or unobserved regions of the scene. In recent work, we have shown that advances in image generative models can enable **training-free 4D scene reconstruction** from monocular videos of dynamic scenes. This is achieved through test-time optimization of a set of differentiable 3D object representations, guided by differentiable rendering to minimize reprojection error and maximize compliance to object-centric generative view synthesis constraints (3; 4). We introduce a **fully compositional, multi-object framework**, in which deformable 3D representations of multiple objects are **jointly rendered in the camera frame** to handle inter-object occlusions (or reprojection losses and individually rendered in object-centric frames to leverage view-synthesis priors. Importantly, we employ object-centric view synthesis models—more accurate than scene-level models due to lower intra-object variability—to complete missing geometry. The system learns coordinate transformations between camera and object frames on-the-fly, enabling consistent optimization across viewpoints.

Together, our work on long-term tracking and 4D reconstruction advances the development of video-centric world models—capable of tracking visible content, reconstructing occluded structure, and supporting downstream applications such as policy learning, scene editing, and simulation.

## 4 Improving Physics Engines for Perception-Driven Simulation

Foundation models enable a shift from single-task robots to versatile agents that generalize across environments by leveraging open-world knowledge of scenes and tasks. However, robotic foundation models must also learn physics, sensory dynamics, and behaviour generation—skills that cannot be acquired from static internet data alone. These require sequences of real-world interactions, which are often unsafe during early, exploratory learning.

A central tool for scaling data in robotics is simulation—replicating the physical world within a physics engine. Sim2Real learning has already transformed areas like robot locomotion. To support broader and more accessible simulation, we developed *GENESIS* (1), a fully open-source, GPU-parallelizable physics engine designed to be more general and user-friendly than existing alternatives. GENESIS supports a wide range of materials and physical behaviors—including rigid bodies, deformable objects, fluids, gases, and granular media—using custom physics solvers written from scratch in Taichi. This multi-institutional effort, led by my PhD student Xian Zhou, has now been open-sourced and is actively used by many research labs. The project has 25000 stars on Github.

**Automating Robot Data Generation in Physics Engines with Generative Models and Video-to-Simulation Translation:** Beyond building the physics engine itself, creating diverse scenes, assets, tasks, and reward functions typically requires extensive human effort. A key area where our lab has established leadership is in using generative models of language and vision to automate this process. In our work *Gen2Sim* (14), we introduced a framework that generates 3D assets from 2D images using generative priors, infers task descriptions and their temporal decompositions by querying large language models (LLMs), and constructs reward functions accordingly. Robots are then trained with model-free reinforcement learning to complete the inferred tasks by optimizing these rewards—forming a fully automated pipeline for data generation in simulation. Our follow-up paper *RoboGen* (34) scaled this approach across multiple robot embodiments, and augmented training with classical motion planners in addition to RL, enabling broader and more efficient behavior synthesis.

**RobotArena: Scalable and Reproducible Robotics Evaluations in Simulation** With the rapid progress in robot manipulation and growing momentum across academia and industry toward building robot generalists—agents capable of following instructions and performing diverse tasks in dynamic settings—there is an urgent need for rigorous, scalable benchmarking frameworks to systematically measure research progress. Real-world evaluation of robot policies remains fundamentally unscalable due to challenges related to logistics, safety, and reproducibility—issues that become even more pronounced as policy capabilities expand in scope and complexity.

We are developing RobotArena, a scalable, general, and continually evolving benchmark for evaluating robot policies in simulation. Our initial prototype (13), illustrated in Figure 3, leverages **an automated reality-to-simulation translation method** built upon recent advances in vision-language models, 3D generative modeling, and differentiable rendering to automatically convert video demonstrations from popular real-world robot datasets into simulated evaluation arenas. Within these arenas, robot policies are evaluated directly—without additional training—and scored via a combination of vision-language models and human preference judgments.

Simulated benchmarking enables controlled perturbations of the environment—such as changes in camera viewpoint, background texture, object placement, lighting, friction and other material and surface properties, and object inertial, elastic, plastic, and viscous parameters—allowing us to systematically assess generalization and robustness. The result is a richly diverse yet reproducible suite of evaluation scenarios that meaningfully stress-test the adaptability of modern robot generalists.

Our goal is to expand RobotArena into a continually growing set of environments and a progressively challenging task curriculum, to provide a fair, transparent, and scalable way to measure progress in robotics.

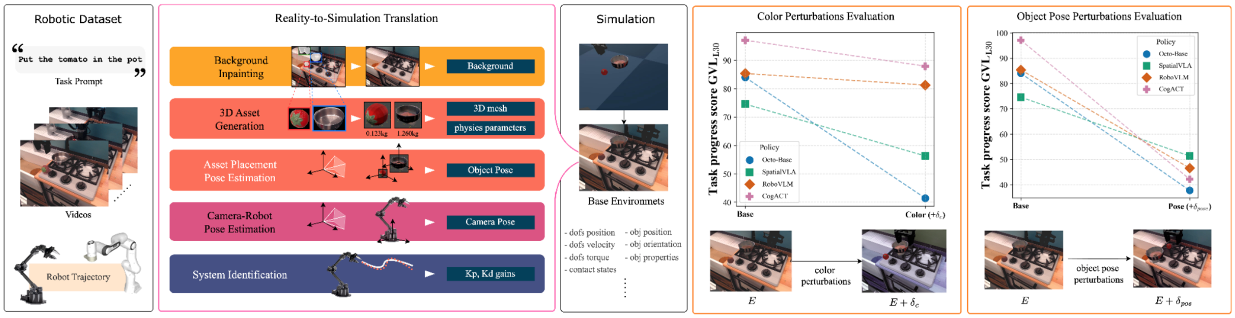


Figure 3: RobotArena (13) generates simulated environments and evaluation schemes from video demonstrations. It evaluates robot policies in the nominal scene and on controlled perturbations. It offers a way to obtain apples-to-apples comparisons across robot manipulation policies trained on any data, real or simulation.

# 5 Memory-Augmented Vision-Language Reasoning

Current vision-language models (VLMs) can only process input information in their context windows so that any VLM will only have the ability to utilize "short-term memory" for decision-making or question-answering. For example, Gemini downsamples videos at 1FPS to handle the explosion of visual tokens across frames. Meaningful collaboration between humans and AI systems requires recalling past events, tendencies, plans, and routines, as well as handling long continuous streams of video.

An area our lab has established leadership in is **memory-augmented vision-language reasoning, with models that encode knowledge using external repositories of experiences**, or on-the-fly constructed short-term 3D feature maps, alongside parametric networks, that learn to write, retrieve and fuse information during inference. The *HELPER* model (29) enhances instruction-following in dynamic environments through retrieval-augmented generation (RAG). It maintains an evolving memory of dialogue and action-plan pairs, retrieving relevant past examples as in-context prompts to guide future responses. This allows the model to store, recall, and adapt user-specific routines—such as "tidying up my kitchen", enabling personalization.

Storing raw action plans in memory places the burden of adaptation on the model's parameters. Our follow-up work, *ICAL* (26), addresses this by editing and refining memories to improve task performance, as shown in Figure 4. ICAL transforms suboptimal demonstrations into generalized, multimodal programs of thought—natural language plans enriched with abstractions like subgoals, causal dependencies, and state transitions. These structured memories enable more effective retrieval-augmented generation, improving generalization and reducing reliance on human demonstrations. ICAL achieved state-of-the-art results on the TEACh benchmark (23) and VisualWebArena benchmark (16), as shown in Tables 1 and 2.

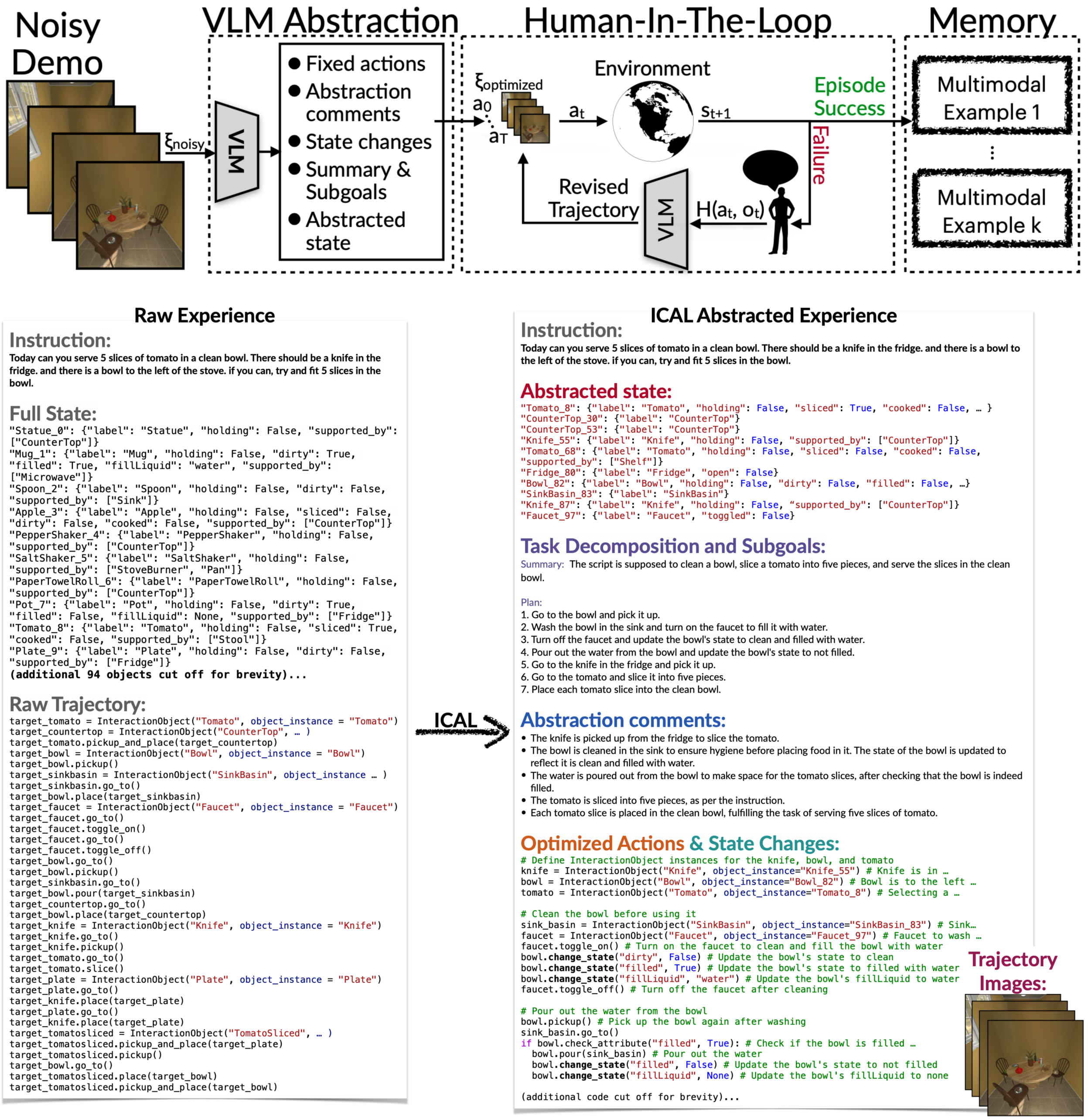
Table 1: **Evaluation of our ICAL method (26) on TEACh unseen validation set.** All evaluations are done using GPT3.5-turbo-1106 unless otherwise noted. Visual Demos = demonstrations labeled with inverse dynamics model. Kinesthetic Demos = demos labeled with GT actions. GC = goal-condition success

| | Success | GC |
|---|---|---|
| *Ground truth segm, depth, attributes* | | |
| HELPER hand-written (27) | 34.5 | 36.7 |
| Zero-Shot CoT (18) | 11.8 | 24.6 |
| Raw Visual Demos | 17.2 | 26.6 |
| Raw Kinesthetic Demos | 26.5 | 29.5 |
| ICAL retrieval (ours) | **35.1** | **49.3** |
| w/o programs of thought phase | 29.4 | 44.9 |
| w/o human-in-the-loop | 29.9 | 41.0 |
| w/ retrieval re-ranking | **35.3** | **51.7** |
| w/ GPT4 | 41.7 | 63.6 |
| ICAL SFT | 23.2 | 40.3 |
| ICAL SFT + retrieval | **35.8** | **54.2** |
| *Estimated perception* | | |
| HELPER hand-written (27) | 8.3 | 14.1 |
| ICAL (ours) | **10.5** | **15.4** |

Table 2: **Results of our ICAL method (26) on VisualWebArena.** ICAL outperforms the prior best, GPT4o/V + Set of Marks. All VLM baselines are given Image + SoM + Captions representation (see (17)). Ablation studies were conducted with GPT4V on a subset of 257 episodes.

| | Seen | Unseen | Average |
|---|---|---|---|
| *Open-source VLMs* | | | |
| CogVLM (33; 17) | – | – | 0.33 |
| IDEFICS-80B-Instruct (20; 17) | – | – | 0.99 |
| Qwen2-VL-7B (32) | – | – | 2.9 |
| ICAL Qwen2-VL-7B SFT (ours) | 16.7 | 7.4 | **8.2** |
| *Proprietary VLMs* | | | |
| Gemini-Pro-1.5 | – | – | 11.9 |
| GPT4o (17) | – | – | 18.9 |
| ICAL GPT4o (ours) | 32.3 | 22.3 | **23.4** |
| GPT4V$^*$ (17) | 16.3 | 14.1 | 14.3 |
| ICAL GPT4V$^*$ (ours) | 38.8 | 20.9 | **22.7** |
| *Ablations* | | | |
| GPT4V (17) | 11.5 | 12.9 | 12.7 |
| ICAL (ours) | 28.0 | 21.6 | 22.2 |
| w/o image | 28.0 | 17.3 | 19.0 |
| w/ full text trajectory | 57.7 | 21.6 | 25.5 |

**Using 3D Scene Representations as Short-Term Memory for VLM Reasoning:** Videos rapidly overwhelm the context window of today's vision-language models (VLMs). To address this, we propose



Raw Experience

Instruction:

Today can you serve 5 slices of tomato in a clean bowl. There should be a knife in the fridge. and there is a bowl to the left of the stove. if you can, try and fit 5 slices in the bowl.

Full State:

```
"Statue_0": {"label": "Statue", "holding": False, "supported_by":
["CounterTop"]}
"Mug_1": {"label": "Mug", "holding": False, "dirty": True,
"filled": True, "fillLiquid": "water", "supported_by":
["Microwave"]}
"Spoon_2": {"label": "Spoon", "holding": False, "dirty": False,
"supported_by": ["Sink"]}
"Apple_3": {"label": "Apple", "holding": False, "sliced": False,
"dirty": False, "cooked": False, "supported_by": ["CounterTop"]}
"PepperShaker_4": {"label": "PepperShaker", "holding": False,
"supported_by": ["CounterTop"]}
"SaltShaker_5": {"label": "SaltShaker", "holding": False,
"supported_by": ["StoveBurner", "Pan"]}
"PaperTowelRoll_6": {"label": "PaperTowelRoll", "holding": False,
"supported_by": ["CounterTop"]}
"Pot_7": {"label": "Pot", "holding": False, "dirty": True,
"filled": False, "fillLiquid": None, "supported_by": ["Fridge"]}
"Tomato_8": {"label": "Tomato", "holding": False, "sliced": True,
"cooked": False, "supported_by": ["Stool"]}
"Plate_9": {"label": "Plate", "holding": False, "dirty": False,
"supported_by": ["Fridge"]}
(additional 94 objects cut off for brevity)...
```

Raw Trajectory:

```
target_tomato = InteractionObject("Tomato", object_instance = "Tomato")
target_countertop = InteractionObject("CounterTop", … )
target_tomato.pickup_and_place(target_countertop)
target_bowl = InteractionObject("Bowl", object_instance = "Bowl")
target_bowl.pickup()
target_sinkbasin = InteractionObject("SinkBasin", object_instance … )
target_sinkbasin.go_to()
target_bowl.place(target_sinkbasin)
target_faucet = InteractionObject("Faucet", object_instance = "Faucet")
target_faucet.go_to()
target_faucet.toggle_on()
target_faucet.go_to()
target_faucet.toggle_off()
target_bowl.go_to()
target_bowl.pickup()
target_sinkbasin.go_to()
target_bowl.pour(target_sinkbasin)
target_countertop.go_to()
target_bowl.place(target_countertop)
target_knife = InteractionObject("Knife", object_instance = "Knife")
target_knife.go_to()
target_knife.pickup()
target_tomato.go_to()
target_tomato.slice()
target_plate = InteractionObject("Plate", object_instance = "Plate")
target_plate.go_to()
target_knife.place(target_plate)
target_plate.go_to()
target_knife.place(target_plate)
target_tomatosliced = InteractionObject("TomatoSliced", … )
target_tomatosliced.pickup_and_place(target_plate)
target_tomatosliced.pickup()
target_bowl.go_to()
target_tomatosliced.place(target_bowl)
target_tomatosliced.pickup_and_place(target_bowl)
```

ICAL Abstracted Experience

Instruction:

Today can you serve 5 slices of tomato in a clean bowl. There should be a knife in the fridge. and there is a bowl to the left of the stove. if you can, try and fit 5 slices in the bowl.

Abstracted state:

```
"Tomato_8": {"label": "Tomato", "holding": False, "sliced": True, "cooked": False, … }
"CounterTop_30": {"label": "CounterTop"}
"CounterTop_53": {"label": "CounterTop"}
"Knife_55": {"label": "Knife", "holding": False, "supported_by": ["CounterTop"]}
"Tomato_68": {"label": "Tomato", "holding": False, "sliced": False, "cooked": False,
"supported_by": ["Shelf"]}
"Fridge_80": {"label": "Fridge", "open": False}
"Bowl_82": {"label": "Bowl", "holding": False, "dirty": False, "filled": False, …}
"SinkBasin_83": {"label": "SinkBasin"}
"Knife_87": {"label": "Knife", "holding": False, “supported_by": ["CounterTop"]}
"Faucet_97": {"label": "Faucet", "toggled": False}
```

Task Decomposition and Subgoals:

Summary: The script is supposed to clean a bowl, slice a tomato into five pieces, and serve the slices in the clean bowl.

Plan:
1. Go to the bowl and pick it up.
2. Wash the bowl in the sink and turn on the faucet to fill it with water.
3. Turn off the faucet and update the bowl's state to clean and filled with water.
4. Pour out the water from the bowl and update the bowl's state to not filled.
5. Go to the knife in the fridge and pick it up.
6. Go to the tomato and slice it into five pieces.
7. Place each tomato slice into the clean bowl.

Abstraction comments:

- The knife is picked up from the fridge to slice the tomato.
- The bowl is cleaned in the sink to ensure hygiene before placing food in it. The state of the bowl is updated to reflect it is clean and filled with water.
- The water is poured out from the bowl to make space for the tomato slices, after checking that the bowl is indeed filled.
- The tomato is sliced into five pieces, as per the instruction.
- Each tomato slice is placed in the clean bowl, fulfilling the task of serving five slices of tomato.

Optimized Actions & State Changes:

```
# Define InteractionObject instances for the knife, bowl, and tomato
knife = InteractionObject("Knife", object_instance="Knife_55") # Knife is in …
bowl = InteractionObject("Bowl", object_instance="Bowl_82") # Bowl is to the left …
tomato = InteractionObject("Tomato", object_instance="Tomato_8") # Selecting a …

# Clean the bowl before using it
sink_basin = InteractionObject("SinkBasin", object_instance="SinkBasin_83") # Sink…
faucet = InteractionObject("Faucet", object_instance="Faucet_97") # Faucet to wash …
faucet.toggle_on() # Turn on the faucet to clean and fill the bowl with water
bowl.change_state("dirty", False) # Update the bowl's state to clean
bowl.change_state("filled", True) # Update the bowl's state to filled with water
bowl.change_state("fillLiquid", "water") # Update the bowl's fillLiquid to water
faucet.toggle_off() # Turn off the faucet after cleaning

# Pour out the water from the bowl
bowl.pickup() # Pick up the bowl again after washing
sink_basin.go_to()
if bowl.check_attribute("filled", True): # Check if the bowl is filled …
  bowl.pour(sink_basin) # Pour out the water
  bowl.change_state("filled", False) # Update the bowl's state to not filled
  bowl.change_state("fillLiquid", None) # Update the bowl's fillLiquid to none

(additional code cut off for brevity)...
```

Figure 4: ICAL transforms raw experience into useful programs of thought for in-context learning. *Top:* Given a noisy trajectory, It prompts a VLM to optimize actions and add language annotations. The optimized trajectory is executed, incorporating human feedback on failures. Successful examples are stored for future VLM in-context action generation. *Bottom:* An example of the raw, noisy trajectory (left), and the final abstracted example after ICAL (right).

compressing the input video into a 3D feature map that serves as a form of short-term memory. Instead of using standard temporal positional encodings, we encode frame tokens with their corresponding XYZ scene coordinates and merge tokens from nearby locations—significantly reducing the number of tokens the model must reason over. Our approach supports joint training on both RGB images and posed RGB-D video sequences by adjusting token positional encodings to reflect either 2D pixel locations or 3D world coordinates, as illustrated in Figure 5. Our models excel in answering queries that require integrating information spread across long egocentric videos, where existing VLMs struggle (36).

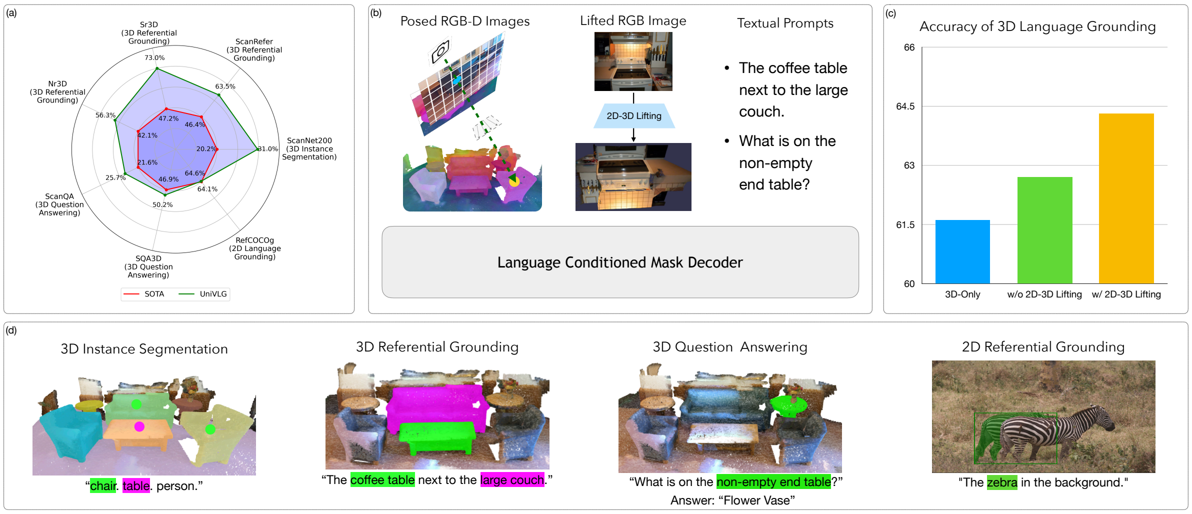


Figure 5: **Joint 2D and 3D Reasoning in Vision-Language Models.** In (11; 12), we introduce vision-language models that can be trained across both 2D images and egocentric RGB-D videos simply by changing the positional encoding of the visual tokens to be 2D or 3D aware, reflecting $(x, y)$ pixel coordinates or $(X, Y, Z)$ world coordinates, respectively.

# 6 Future Directions: Robot Reasoning Fast and Slow for Novice-to-Expert Progression

Looking ahead, a central direction of my research is to **unify perception, memory, and action into interactive, grounded world models that support intelligent exploration through generalization across tasks, environments, and embodiments**. We aim to develop agents that explore through **memory-driven, analogical reasoning**, enabling them to draw on past experiences—reasoning about objects, their 3D geometry, orientations, and relational context—to adapt their interaction strategies in a compositional and interpretable manner. Building on our recent work on "thinking" vision-language models (VLMs) (25), we are extending this paradigm to vision-language-action agents (VLAs) that incorporate classical computer vision operations—such as object detection, 3D reconstruction, memory retrieval, and analogy-based prediction—as structured components in a chain of thought. This form of "slow thinking"—which decomposes tasks step by step and plans object and end-effector trajectories—serves as a foundation for learning fast, reactive policies that generalize out-of-distribution and gradually eliminate the need for explicit task decomposition. Through this framework, our robots will be able to explore more intelligently, generate high-quality training data autonomously, and greatly reduce reliance on brute-force interaction—whether in the real world or within self-generated simulation twins.

**Real-to-Sim and Sim-to-Real Learning** We are advancing our research on reality-to-simulation translation, enabling the real-time creation of editable, interactive 4D environments from video to support safe exploration, internal simulation, and scalable data generation. In parallel, we are automating sim-to-real learning for a wide range of robotics applications, including locomotion, manipulation, human-robot collaboration. The overarching goal is to build a universal data engine for robotics, through an automated curriculum of tasks, scenarios, and data generation pipelines enabling seamless real-to-sim and sim-to-real learning across different robot embodiments and task domains. We find particular value in assistive robotics for the elderly and are working on sim-to-real methodologies for that. It includes modeling elderly humans as humanoid agents with limited joint mobility or torque, and training robots in simulation to assist with a variety of everyday tasks.

**From Explicit Physics Engines to Neural Simulators** Generative modeling holds great promise for the next generation of model-based learning, by accurately modeling the manifold of state-action trajectories, and facilitating searches for high-performing actions. We are working towards generative 3D world models that can explain scene dynamics primarily through particle-based object, robot and camera motion, while also predicting necessary appearance changes—serving as an intuitive, neural physics engine. We envision learning such models using rich supervision in simulation environments, and extending them to the real world through rendering-based supervision. Guiding these diffusion-based world models to achieve specific scene configurations, would generate corresponding end-effector actions or entity motions, through differentiable (4) or non-differentiable guidance (38).

## References

[1] https://github.com/genesis-embodied-ai/genesis.

[2] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari. Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles, 2022.

[3] W.-H. Chu, L. Ke, and K. Fragkiadaki. Dreamscene4d: Dynamic multi-object scene generation from monocular videos. *NeurIPS*, 2024.

[4] W.-H. Chu, C. Zhang, L. Ke, S. Zakharov, P. Tokmakov, and K. Fragkiadaki. Generative 3d particle world models. In submission, 2025.

[5] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. *CoRL*, 2023.

[6] N. Gkanatsios, J. Xu, M. Bronars, A. Mousavian, T.-W. Ke, and K. Fragkiadaki. 3d flow actor: An efficient generative 3d policy for bimanual control, 2025.

[7] M. Grotz, M. Shridhar, T. Asfour, and D. Fox. Peract2: Benchmarking and learning for robotic bimanual manipulation tasks, 2024.

[8] A. W. Harley, Z. Fang, and K. Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. *ECCV*, 2022.

[9] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki. Simple-bev: What really matters for multi-sensor bev perception? *ICRA*, 2022.

[10] A. W. Harley, F. Li, S. K. Lakshmikanth, X. Zhou, H.-Y. F. Tung, and K. Fragkiadaki. Visual representation learning with 3d view-contrastive inverse graphics networks. *ICLR*, 2020.

[11] A. Jain, P. Katara, N. Gkanatsios, A. W. Harley, G. Sarch, K. Aggarwal, V. Chaudhary, and K. Fragkiadaki. Odin: A single model for 2d and 3d segmentation. *CVPR*, 2024.

[12] A. Jain, A. Swerdlow, Y. Wang, S. Arnaud, A. Martin, A. Sax, F. Meier, and K. Fragkiadaki. Unifying 2d and 3d vision-language understanding. *ICML*, 2025.

[13] Y. Jangir, Y. Zhang, K. Yamazaki, C. Zhang, K.-H. Tu, T.-W. Ke, L. Ke, Y. Bisk, and K. Fragkiadaki. Robotarena: Towards unlimited robot evaluation via demonstration-to-simulation translation. *In submission*, 2025.

[14] P. Katara, X. Zhou, and K. Fragkiadaki. Gen2sim: Generating simulation environments from generative models. *ICRA*, 2023.

[15] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *CoRL*, 2024.

[16] J. Y. Koh, R. Lo, L. Jang, V. Duvvur, M. C. Lim, P.-Y. Huang, G. Neubig, S. Zhou, R. Salakhutdinov, and D. Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks, 2024.

[17] J. Y. Koh, R. Lo, L. Jang, V. Duvvur, M. C. Lim, P.-Y. Huang, G. Neubig, S. Zhou, R. Salakhutdinov, and D. Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024.

[18] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[19] S. Lal, M. Prabhudesai, I. Mediratta, A. W. Harley, and K. Fragkiadaki. Coconets: Continuous contrastive 3d scene representations. *CVPR*, 2021.

[20] H. Laurençon, L. Saulnier, L. Tronchon, S. Bekman, A. Singh, A. Lozhkov, T. Wang, S. Karamcheti, A. M. Rush, D. Kiela, M. Cord, and V. Sanh. Obelics: An open web-scale filtered dataset of interleaved image-text documents, 2023.

[21] Y. Lin, C. Lin, P. Pan, H. Yan, Y. Feng, Y. Mu, and K. Fragkiadaki. Partcrafter: Compositional 3d scene reconstruction with structured diffusion, 2025.

[22] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks, 2022.

[23] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, and D. Hakkani-Tur. Teach: Task-driven embodied agents that chat, 2021.

[24] M. Prabhudesai, F. Tung, S. Javed, A. Harley, M. Sieb, and K. Fragkiadaki. Embodied language grounding with implicit 3D visual feature representations. In *CVPR*, 2020.

[25] G. Sarch, A. Jain, S. Saha, N. Khandelwal, A. K. Mike Tarr, and K. Fragkiadaki. Grounded reinforcement learning for visual reasoning. *in submission*, 2025.

[26] G. Sarch, L. Jang, M. J. Tarr, W. W. Cohen, K. Marino, and K. Fragkiadaki. Vlm agents generate their own memories: Distilling experience into embodied programs of thought. *NeurIPS*, 2024.

[27] G. Sarch, Y. Wu, M. Tarr, and K. Fragkiadaki. Open-ended instructable embodied agents with memory-augmented large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

[28] G. Sarch, Y. Wu, M. J. Tarr, and K. Fragkiadaki. Open-ended instructable embodied agents with memory-augmented large language models. *EMNLPfindings*, 2023.

[29] G. Sarch, Y. Wu, M. J. Tarr, and K. Fragkiadaki. Open-ended instructable embodied agents with memory-augmented large language models. *EMNLP findings*, 2023.

[30] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[31] H.-Y. F. Tung, R. Cheng, and K. Fragkiadaki. Learning spatial common sense with geometry-aware recurrent networks. *arXiv:1901.00003*, 2018.

[32] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[33] W. Wang, Q. Lv, W. Yu, W. Hong, J. Qi, Y. Wang, J. Ji, Z. Yang, L. Zhao, X. Song, J. Xu, B. Xu, J. Li, Y. Dong, M. Ding, and J. Tang. Cogvlm: Visual expert for pretrained language models, 2023.

[34] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. *ICML*, 2024.

[35] Z. Xian, N. Gkanatsios, T. Gervet, T.-W. Ke, and K. Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *CoRL*, 2023.

[36] R. Xu, Z. Huang, T. Wang, Y. Chen, J. Pang, and D. Lin. Vlm-grounder: A vlm agent for zero-shot 3d visual grounding, 2024.

[37] B. Yang, J. Schneider, and K. Fragkiadaki. Gandalf: Grounding autonomous driving agents via language feedback, 2025. In submission.

[38] B. Yang, H. Su, N. Gkanatsios, T.-W. Ke, A. Jain, J. Schneider, and K. Fragkiadaki. Diffusion-es: Gradient-free planning with diffusion for autonomous driving and zero-shot instruction following. *CVPR*, 2024.

[39] B. Zhang, L. Ke, A. W. Harley, and K. Fragkiadaki. Tapip3d: Tracking any point in persistent 3d geometry, 2025.