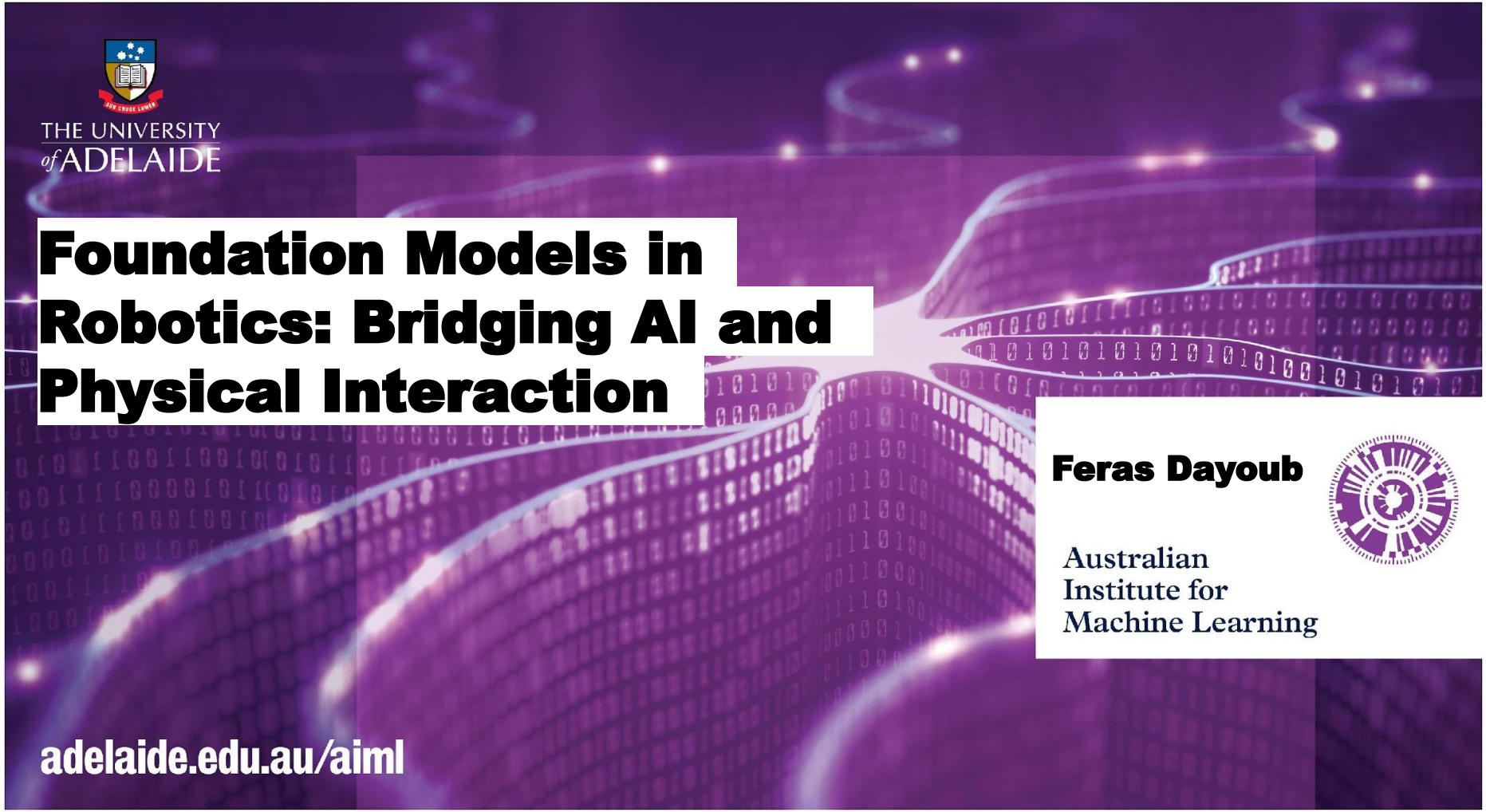

THE UNIVERSITY of ADELAIDE
Foundation Models in Robotics: Bridging AI and Physical Interaction
Feras Dayoub
Australian Institute for Machine Learning
adelaide.edu.au/aiml

Overview:

- Introduction to Foundation Models.
- Types and Applications.
- Grounding in Robotics.
- Action Generation with Foundation Models.
- Benefits, Challenges, and Future Directions.

THE UNIVERSITY of ADELAIDE

Definition of Foundation Models (FMs)

A Foundation Model is any large scale ML model **trained on broad data at scale**, typically using **self-supervised learning** methods. These models are designed to be **adaptable** to a wide range of **downstream tasks** through **fine-tuning or prompting**.

They serve as a **'foundation'** upon which more specialized applications can be built.

THE UNIVERSITY of ADELAIDE

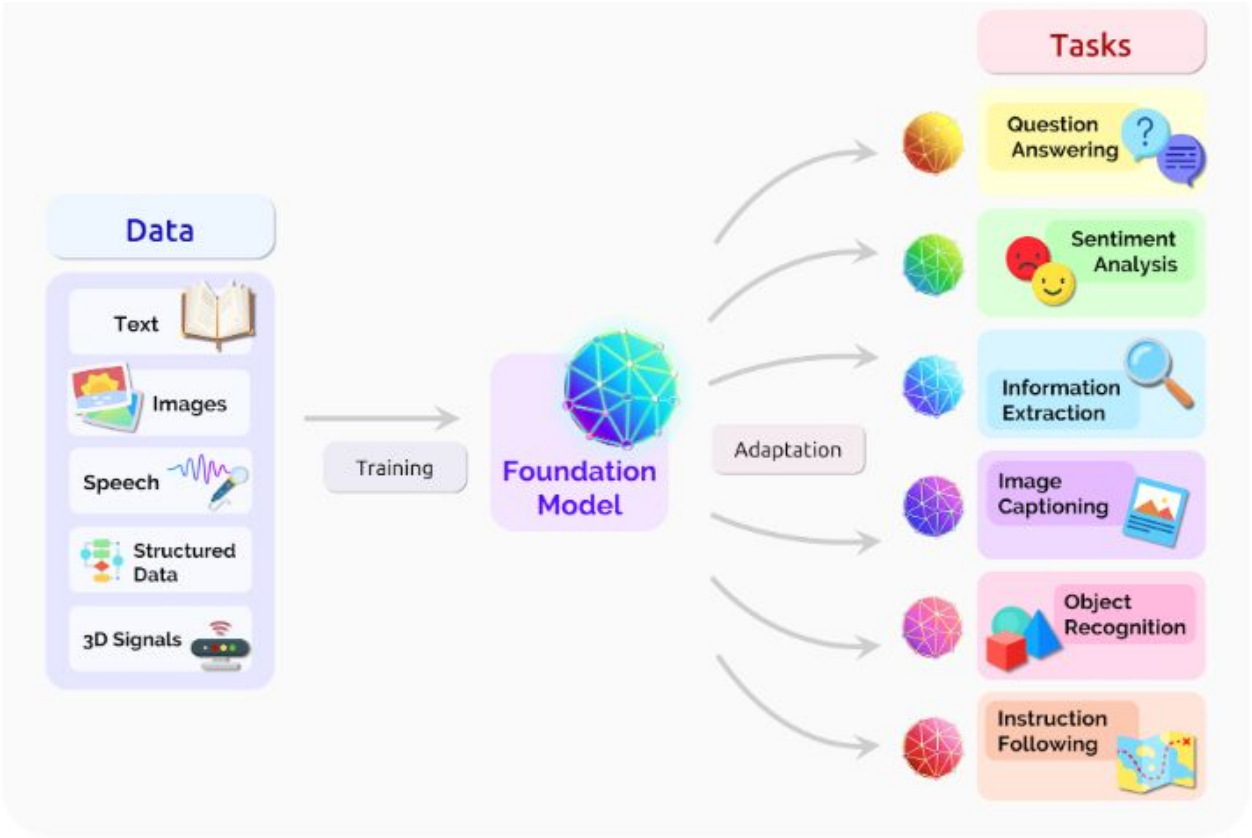


Fig. 2. A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks.

* Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein et al. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258* (2021).

THE UNIVERSITY of ADELAIDE

Brief history and evolution

- The concept has its roots in **transfer learning and pre-training**, which gained popularity in the early **2010s**.

- The term 'Foundation Model' itself was coined more recently, in **a 2021 paper from Stanford***.
    - "any model that is trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks"

* Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein et al. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258* (2021).

THE UNIVERSITY of ADELAIDE

Key milestones

2017: Transformer

2018: Bidirectional encoder representations from transformers (BERT)

2020: GPT-3, Vision Transformer (ViT)

2021: Contrastive Language–Image Pre-training (CLIP)

2022: DALL-E 2 and SAM (Segment Anything Model)

2023: Robotics-specific Foundation Models like RT-1 and RT-2. GPT-4, Llama 2, Claude.

2024: o1, Gemini 2.0,

2025: DeepSeek's R1 and Janus Pro, OpenAI's Operator

Copilot

Google Gemini

deepseek

Meta

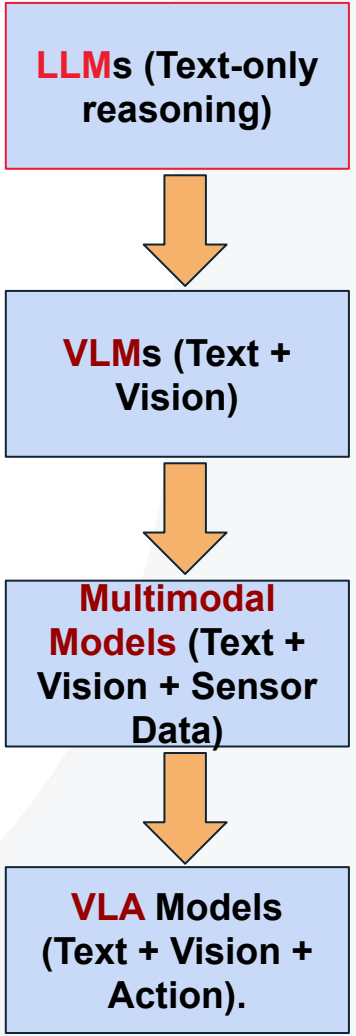
Types of Foundation Models

Category	Subtypes	Examples
Language Models	**LLMs, Code/Math LMs**	GPT-4, Codex, LLaMA 2, Claude
Vision Models	**Pure Vision, Vision-Language**	ViT, CLIP, SAM, DINOv2
Multimodal Models	**3+ Modalities**	GPT-4V, Gemini 2
Audio Models	**Speech, Music**	Whisper, AudioLM, WaveNet
Embodied/Vision-Language-Action (VLA) model	**Robotics, Simulations**	RT-1, RT-2, MineDojo, MuZero

THE UNIVERSITY of ADELAIDE

From Language to Embodied Intelligence

- **Large Language Models (LLMs) :**
  - Text generation, reasoning, and instruction following, but **blind to the physical world**.
- **Vision-Language Models (VLMs): Bridging Text and Images**
  - Align visual and textual embeddings.
  - Vision Transformer (ViT): Adapted transformer architecture to images seamless integration with language models
- **Multimodal Models**
  - Shared embedding spaces: All modalities (text, image, sensor data) are encoded into a unified latent space.
  - Cross-modal attention: Language could “query” visual features, and vice versa.
- **Vision-Language-Action (VLA) Models: Embodied AI**
  - Integrating action into the multimodal loop.



Integrating FM in Robotic Systems

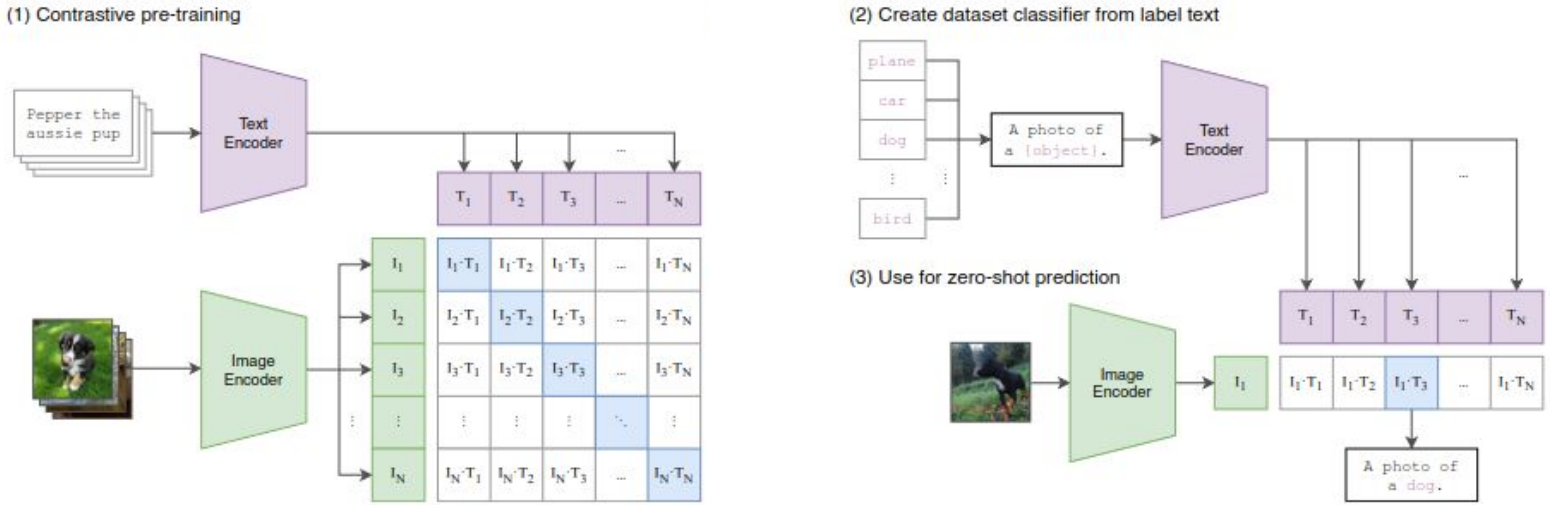
Large foundational models can enhance the capabilities of robotic systems by **combining semantic understanding** with **spatial reasoning**, leading to more intelligent and adaptable robots capable of performing complex tasks.

THE UNIVERSITY of ADELAIDE

CLIP

(Contrastive Language-Image Pre-training)

Unified Embedding Space: Both the image and text encoders map their inputs into a shared embedding space where similar images and texts are close to each other

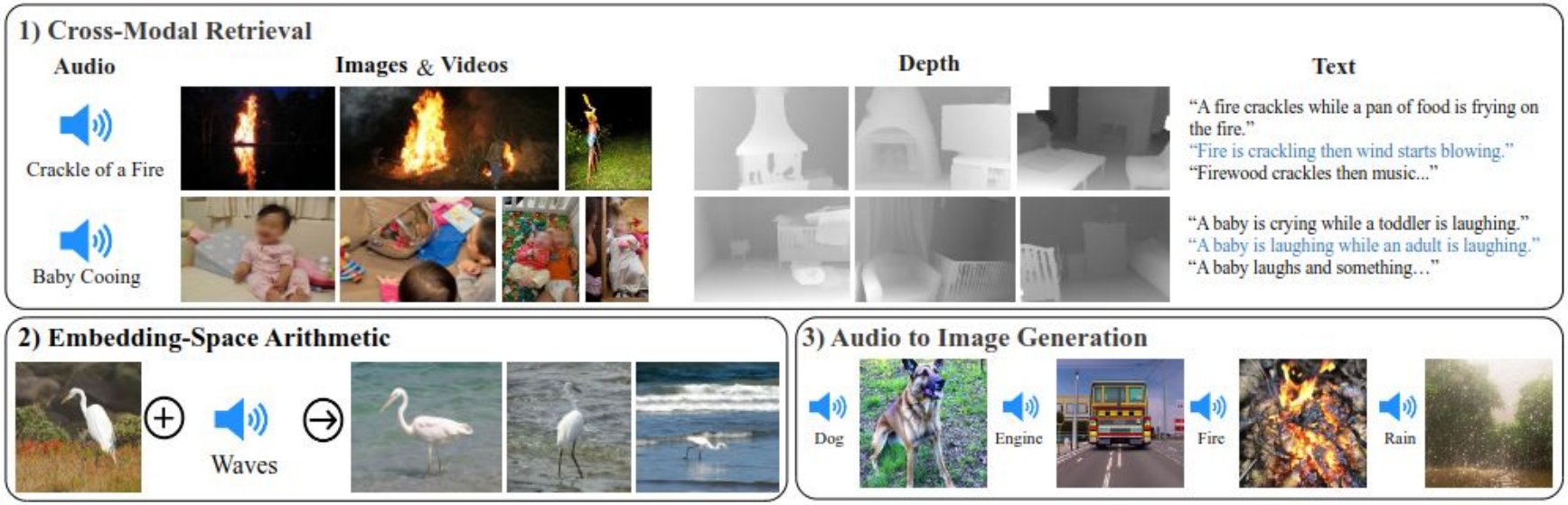


THE UNIVERSITY of ADELAIDE

IMAGEBIND: One Embedding Space To Bind Them All

Rohit Girdhar* Alaaeldin El-Nouby* Zhuang Liu Mannat Singh
Kalyan Vasudev Alwala Armand Joulin Ishan Misra*
FAIR, Meta AI

https://facebookresearch.github.io/ImageBind



CLIPORT: What and Where Pathways for Robotic Manipulation

Mohit Shridhar [1,†] **Lucas Manuelli** [2] **Dieter Fox** [1,2]
[1]University of Washington [2]NVIDIA
mshr@cs.washington.edu lmanuelli@nvidia.com fox@cs.washington.edu

cliport.github.io

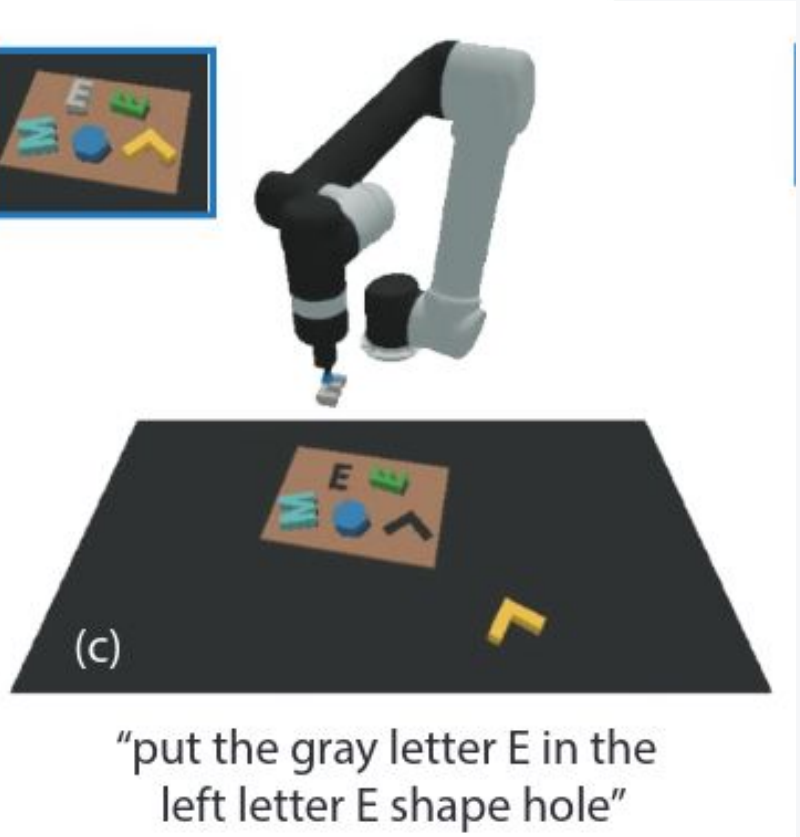
Abstract: How can we imbue robots with the ability to manipulate objects precisely but also to reason about them in terms of abstract concepts? Recent works in manipulation have shown that end-to-end networks can learn dexterous skills that require precise spatial reasoning, but these methods often fail to generalize to new goals or quickly learn transferable concepts across tasks. In parallel, there has been great progress in learning generalizable semantic representations for vision and language by training on large-scale internet data, however these representations lack the spatial understanding necessary for fine-grained manipulation. To this end, we propose a framework that combines the best of both worlds: a two-stream architecture with semantic and spatial pathways for vision-based manipulation. Specifically, we present CLIPORT, a language-conditioned imitation-learning agent that combines the broad semantic understanding (*what*) of CLIP [1]

THE UNIVERSITY of ADELAIDE

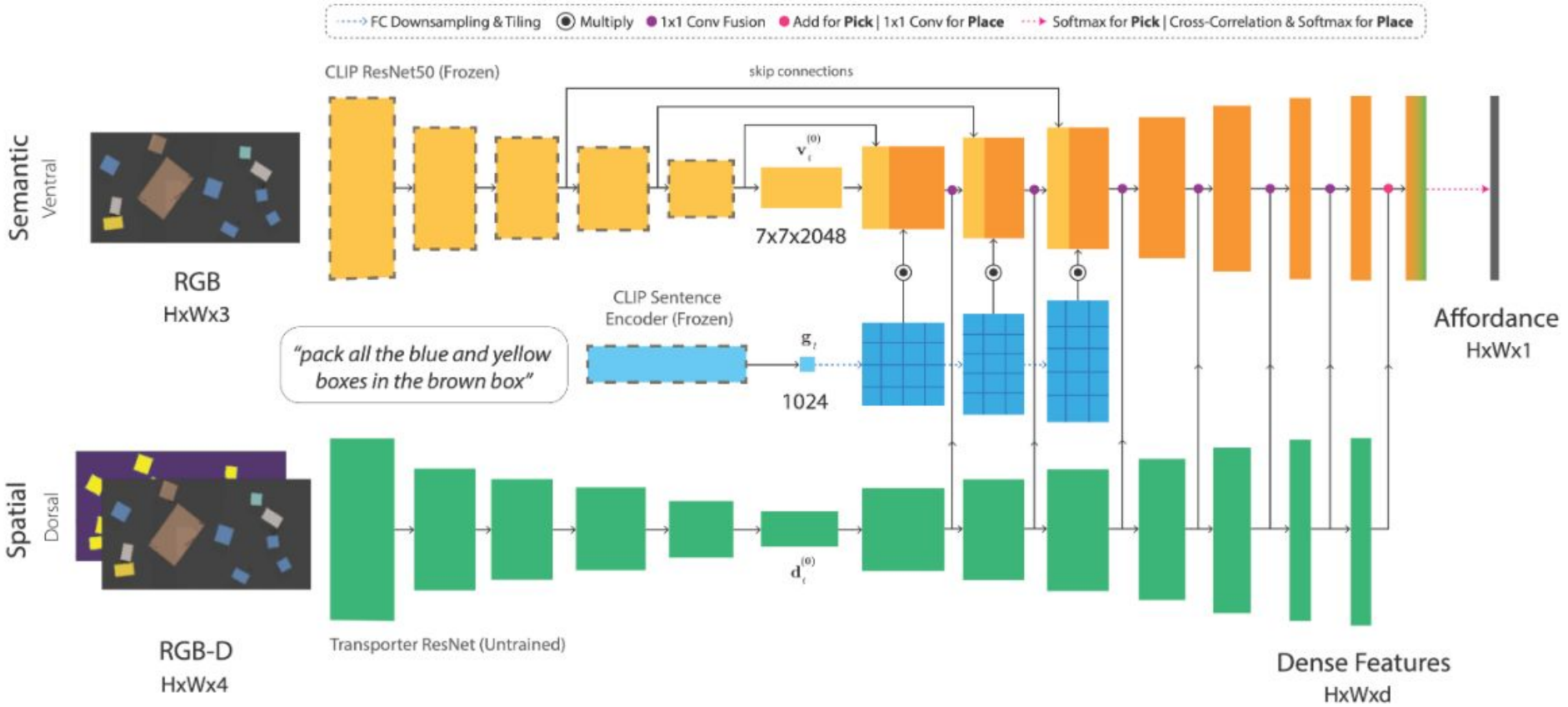
Case study: CLIPort for object manipulation

- Combines the **CLIP model** for semantic understanding and **Transporter Networks** for transport-based policies.

- **Uses a two-stream architecture:** one for encoding semantic information (via CLIP) and another for encoding spatial information (via Transporter Networks).

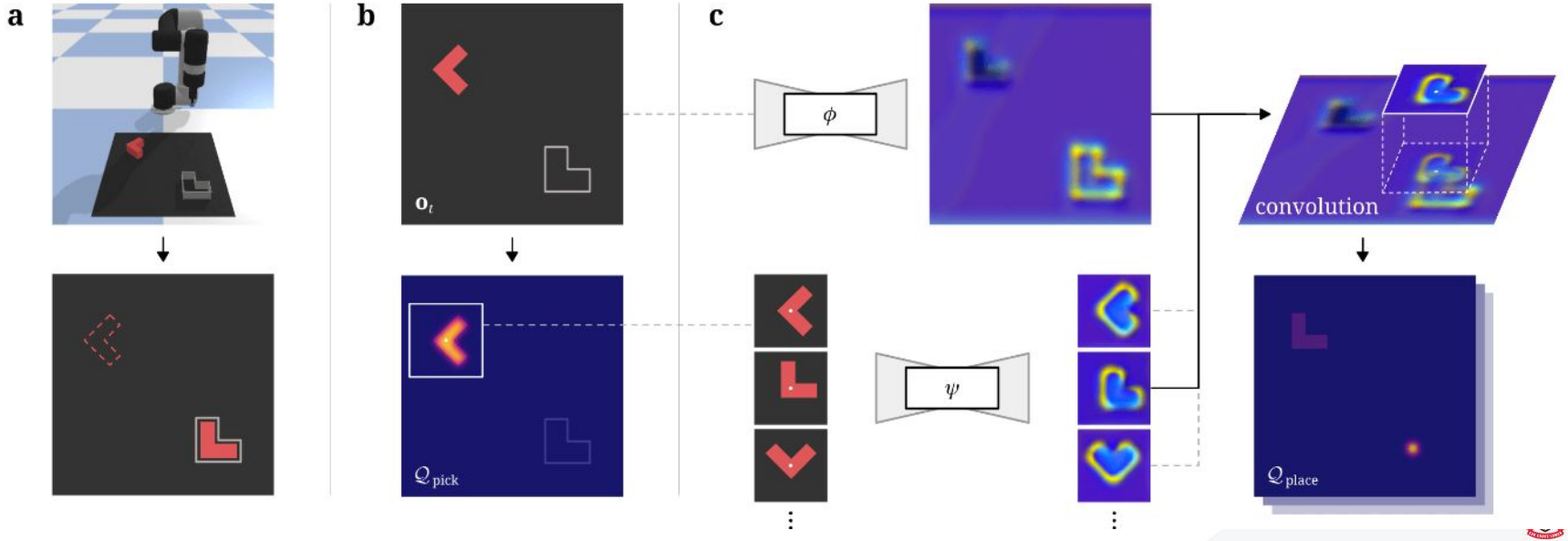


How CLIPort works



Transporter Networks

- It can predict pick-and-place affordances by attending to **local regions for picking** and **matching placements through cross-correlation of visual features**.



Zeng, A., Florence, P., Tompson, J., Welker, S., Chien, J., Attarian, M., ... & Lee, J. (2021, October). Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning* (pp. 726-747). PMLR.


Put in Bowl Task
* put the red blocks in the green bowl *
(unseen red-green goal combo)

Grounding

Definition of Grounding

- Connecting abstract concepts or symbols (like words or phrases) to concrete, real-world entities or actions.

- **For robots,** this means linking language understanding to physical perceptions and actions in the real world.

THE UNIVERSITY of ADELAIDE

Why is Grounding Important?

- LLMs are trained on vast amounts of text data, giving them broad knowledge and language understanding.
- However, this knowledge is abstract and disconnected from the physical world.
- For a robot to use this knowledge effectively, it needs to ground it in its physical environment and capabilities.

THE UNIVERSITY of ADELAIDE

Challenges of Grounding in Robotics

- **Symbol Grounding Problem:** This is the challenge of connecting symbolic representations (like words) to their physical referents in the world.
- **Embodiment:** Robots have specific physical capabilities and limitations that may not align perfectly with language descriptions.
- **Context Dependency:** The meaning of language can change based on the physical context, which the robot needs to understand.

THE UNIVERSITY of ADELAIDE

Approaches to Grounding in Robotics

- **Perception-based Grounding: Linking language to visual or sensory inputs.**
  - Example: Associating the word "cup" with visual features of cups the robot has seen.
- **Action-based Grounding: Connecting language to the robot's possible actions.**
  - Example: Understanding "pick up" in terms of the robot's grasping capabilities.
- **Context-aware Grounding: Interpreting language based on the current environment and task.**
  - Example: Understanding "stay close" as maintaining a short distance in a crowded corridor but a larger distance in an open lobby.

THE UNIVERSITY of ADELAIDE

Example: Grounding in Task Planning

Consider a robot instructed to **"bring me a cold drink."**

Grounding this instruction involves:

a) Understanding what constitutes a "cold drink" in the current context.

b) Locating the fridge in the environment.

c) Knowing how to open the fridge based on its specific design.

d) Identifying suitable object inside the fridge.

e) Understanding how to grasp and carry the drink safely.

THE UNIVERSITY of ADELAIDE

Ongoing Challenges in Grounding

- **Handling Ambiguity:** Language can be ambiguous, and robots need to resolve this ambiguity based on context.
- **Dealing with Novel Situations:** Robots need to ground language in situations they haven't encountered before.
- **Feedback and Learning:** Developing systems that can learn and improve their grounding over time through interaction and feedback.

THE UNIVERSITY of ADELAIDE

Action Generation with FMs

A. LLMs for task planning: SayCan approach

B. LLMs for scalable task planning in complex environments: SayPlan approach

C. Reward function generation: Language to Rewards (L2R)

D. Action generation: RT-1 and RT-2 models

THE UNIVERSITY of ADELAIDE

Do As I Can, Not As I Say: Grounding Language in Robotic Affordances

[1] Michael Ahn*, Anthony Brohan*, Noah Brown*, Yevgen Chebotar*, Omar Cortes*, Byron David*, Chelsea Finn*, Chuyuan Fu†, Keerthana Gopalakrishnan*, Karol Hausman*, Alex Herzog†, Daniel Ho†, Jasmine Hsu*, Julian Ibarz*, Brian Ichter*, Alex Irpan*, Eric Jang*, Rosario Jauregui Ruano*, Kyle Jeffrey*, Sally Jesmonth*, Nikhil J Joshi*, Ryan Julian*, Dmitry Kalashnikov*, Yuheng Kuang*, Kuang-Huei Lee*, Sergey Levine*, Yao Lu*, Linda Luu*, Carolina Parada*, Peter Pastor†, Jornell Quiambao*, Kanishka Rao*, Jarek Rettinghouse*, Diego Reyes*, Pierre Sermanet*, Nicolas Sievers*, Clayton Tan*, Alexander Toshev*, Vincent Vanhoucke*, Fei Xia*, Ted Xiao*, Peng Xu*, Sichun Xu*, Mengyuan Yan†, Andy Zeng*

*Robotics at Google, †Everyday Robots

Abstract: Large language models can encode a wealth of semantic knowledge about the world. Such knowledge could be extremely useful to robots aiming to act upon high-level, temporally extended instructions expressed in natural language. However, a significant weakness of language models is that they lack real-world experience, which makes it difficult to leverage them for decision making within a given embodiment. For example, asking a language model to describe how to clean a spill might result in a reasonable narrative, but it may not be applicable to a particular agent, such as a robot, that needs to perform this task in a particular environment. We propose to provide real-world grounding by means of pretrained skills, which are used to constrain the model to propose natural language actions that are both feasible and contextually appropriate. The robot can act as the language model's "hands and eyes," while the language model supplies high-level

THE UNIVERSITY of ADELAIDE

A. LLMs for Task Planning: SayCan Approach

Bridge the gap between abstract language understanding and concrete, contextually appropriate robotic actions:

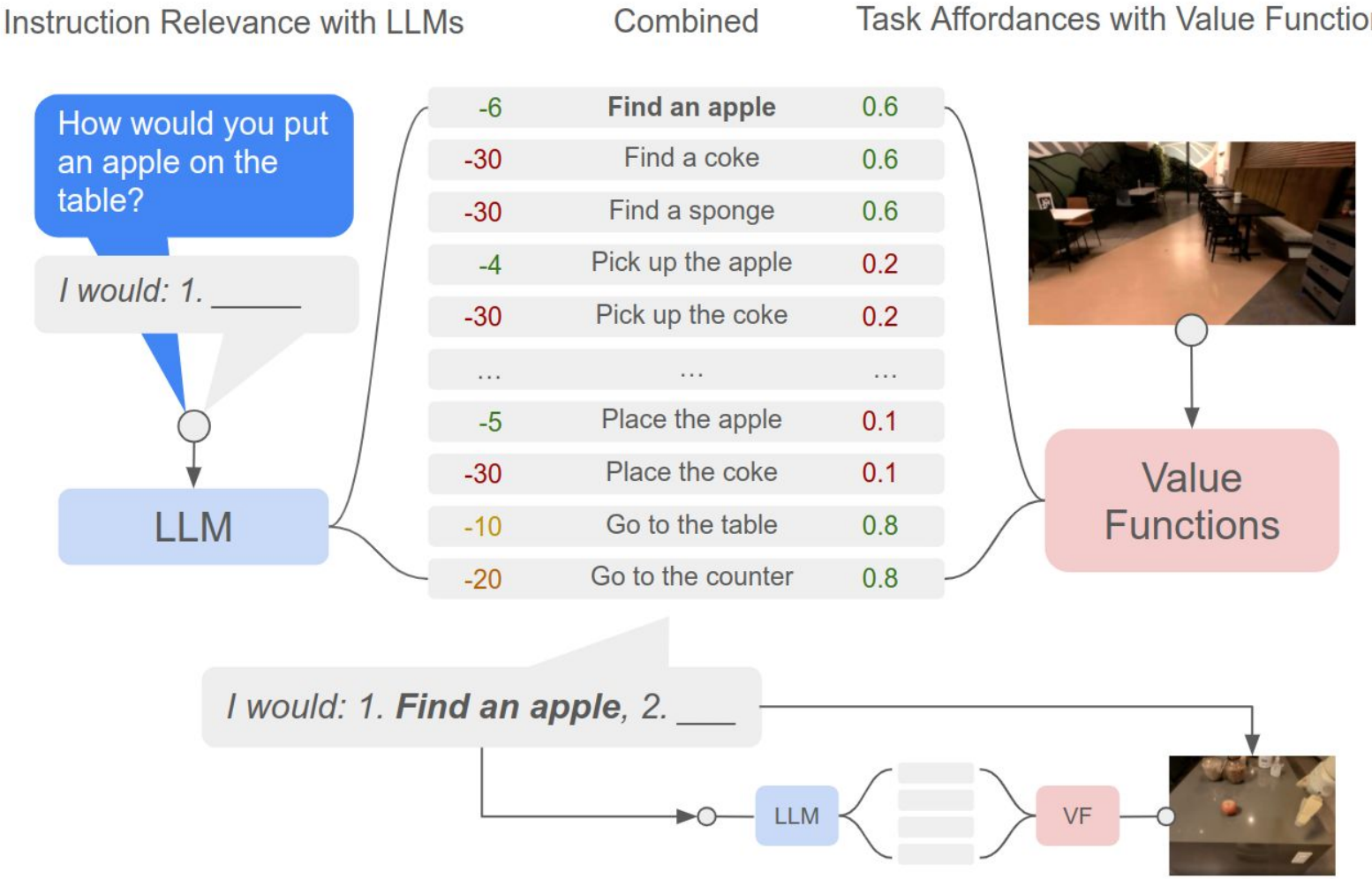
- **Pre-trained Behaviors:** The system uses pretrained robotic behaviors to provide grounding for the language model.
- **Robot as "Hands and Eyes":** The robot acts as the physical interface for the language model, providing real-world context and execution capabilities.
- **Language Model for High-Level Knowledge:** The language model supplies high-level semantic knowledge about tasks and procedures.
- **Value Functions for Grounding:** Value functions associated with low-level tasks provide the necessary grounding to connect the language model's knowledge to the physical environment.

THE UNIVERSITY of ADELAIDE

Key Components of SayCan

1. **LLM for Task Decomposition:**
   - The language model breaks down high-level instructions (e.g., "I spilled my drink, can you help?") into sub-tasks.
2. **Skill Scoring:**
   - The LLM scores the likelihood of each available skill contributing to the completion of the high-level instruction.
   - This helps in selecting contextually appropriate actions.
3. **Affordance Functions:**
   - Each skill has an associated affordance function (e.g., a learned value function).
   - This function quantifies the likelihood of the skill's successful execution in the current state.

THE UNIVERSITY of ADELAIDE


Instruction Relevance with LLMs
Combined
Task Affordances with Value Functions
How would you put an apple on the table?
I would: 1. _____
LLM
-6 Find an apple 0.6
-30 Find a coke 0.6
-30 Find a sponge 0.6
-4 Pick up the apple 0.2
-30 Pick up the coke 0.2
… … …
-5 Place the apple 0.1
-30 Place the coke 0.1
-10 Go to the table 0.8
-20 Go to the counter 0.8
Value Functions
I would: 1. Find an apple, 2. ___
LLM
VF

Key Components of SayCan

4. **Probability Combination:**
   - combines two probabilities:

     a) The LLM's probability of a skill being useful for the instruction.

     b) The value function's probability of successfully executing the skill.
5. **Iterative Skill Selection and Execution:**
   - The system selects the skill with the highest combined probability.
   - The selected skill is executed on the robot.
   - The executed skill is appended to the robot's response in the dialogue.
6. **Continuous Planning:**
   - After each skill execution, the process repeats:
     - The language model is queried again with the updated context.
     - New probabilities are calculated.
     - The next most appropriate skill is selected.
7. **Termination:**
   - The process continues until the language model outputs a termination step (e.g., "Done").

THE UNIVERSITY OF ADELAIDE

User input: I just worked out, can you bring me a drink and a snack to recover?
Robot: I would 1. find a water bottle, 2. pick up the water bottle 3. bring it to you, 4.___

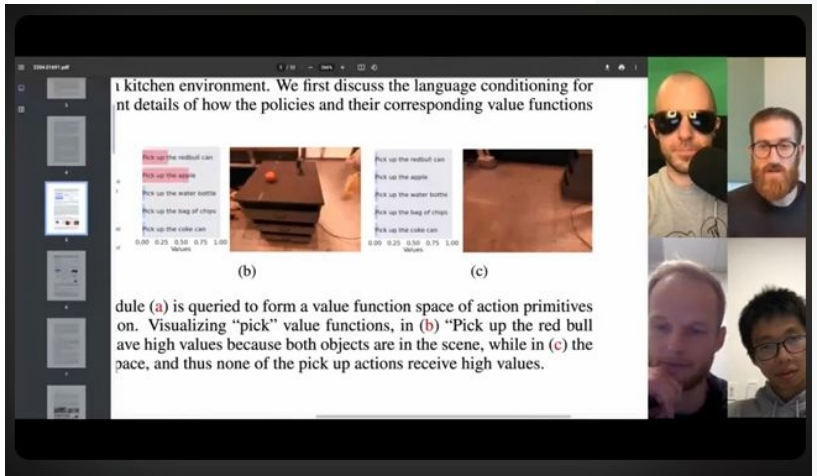
Challenges and Limitations

- **Dependence on Predefined Skills:** The system relies on a predefined set of low-level robotic skills. If a required action is not within this set, SayCan cannot perform the task.
- **Handling Negation and Ambiguity:** when a user specifies a preference against certain items (e.g., "I don't like caffeinated soda"), the system may misinterpret or fail to appropriately filter out the undesired options.
- **Early Termination in Long Instructions:** The algorithm sometimes prematurely terminates when processing lengthy, multi-step instructions.
- **Sequential Task Execution:** SayCan operates by selecting and executing one skill at a time.
- **Limited Environmental Feedback Integration:** SayCan primarily uses value functions to assess the feasibility of actions but lacks a robust mechanism for real-time environmental feedback during task execution.

THE UNIVERSITY of ADELAIDE

How was SayCan developed?

- **68,000 demonstrations** over 11 months.
- **12,000** successful episodes were sourced from **autonomous operations**.
- Language-conditioned policies were trained through **behavioral cloning**.
- The value functions for affordance probabilities were trained using **RL**.
- **Simulation** with **RetinaGAN** to minimize the reality gap between simulation and the real world.



Action Generation with FMs

A. LLMs for task planning: SayCan approach

B. LLMs for scalable task planning in complex environments: SayPlan approach

C. Reward function generation: Language to Rewards (L2R)

D. Action generation: RT-1 and RT-2 models

THE UNIVERSITY of ADELAIDE

SayPlan: Grounding Large Language Models using 3D Scene Graphs for Scalable Robot Task Planning

Krishan Rana[†1], **Jesse Haviland**[*1,2], **Sourav Garg**[*3], **Jad Abou-Chakra**[*1], **Ian Reid**[3], **Niko Sünderhauf**[1]
[1]QUT Centre for Robotics, Queensland University of Technology
[2]CSIRO Data61 Robotics and Autonomous Systems Group
[3]University of Adelaide
[*]Equal Contribution
[†]ranak@qut.edu.au

Abstract:

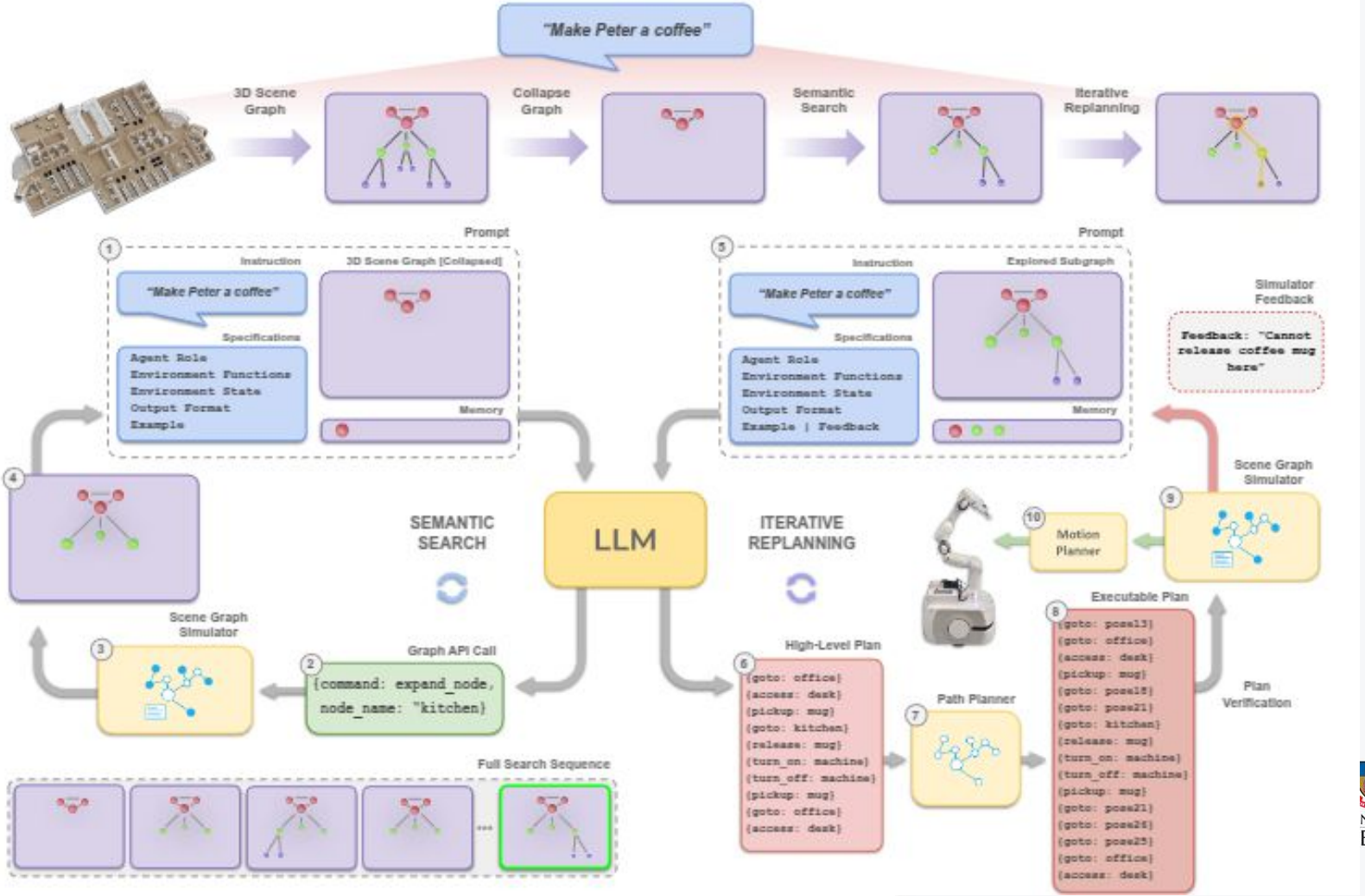
Large language models (LLMs) have demonstrated impressive results in developing generalist planning agents for diverse tasks. However, grounding these plans in expansive, multi-floor, and multi-room environments presents a significant challenge for robotics. We introduce SayPlan, a scalable approach to LLM-based, large-scale task planning for robotics using 3D scene graph (3DSG) representations. To ensure the scalability of our approach, we: (1) exploit the hierarchical nature of 3DSGs to allow LLMs to conduct a *semantic search* for task-relevant

THE UNIVERSITY of ADELAIDE

Key Components of SayPlan

- **3D Scene Graph:** Capturing the spatial and relational information of objects across multiple rooms and floors.
- **Semantic Search:** Ask the LLM to identify a suitable subgraph that contains the required items to solve the tas.
- **Classical Path Planning Integration:** Reduces the planning burden on the LLM by handling navigation between key locations.
- **Iterative Replanning:** Use feedback from a scene graph simulator to refine the initial plan.

THE UNIVERSITY of ADELAIDE


"Make Peter a coffee"
3D Scene Graph
Collapse Graph
Semantic Search
Iterative Replanning
Prompt
Instruction
"Make Peter a coffee"
3D Scene Graph [Collapsed]
Specifications
Agent Role
Environment Functions
Environment State
Output Format
Example
Memory
Explored Subgraph
Example | Feedback
Simulator Feedback
Feedback: "Cannot release coffee mug here"
SEMANTIC SEARCH
LLM
ITERATIVE REPLANNING
Scene Graph Simulator
Graph API Call
{command: expand_node, node_name: "kitchen}
High-Level Plan
{goto: office}
{access: desk}
{pickup: mug}
{goto: kitchen}
{release: mug}
{turn_on: machine}
{turn_off: machine}
{pickup: mug}
{goto: office}
{access: desk}
Path Planner
Motion Planner
Executable Plan
{goto: pose13}
{goto: office}
{access: desk}
{pickup: mug}
{goto: pose15}
{goto: pose21}
{goto: kitchen}
{release: mug}
{turn_on: machine}
{turn_off: machine}
{pickup: mug}
{goto: pose21}
{goto: pose26}
{goto: pose25}
{goto: office}
{access: desk}
Plan Verification
Full Search Sequence


SayPlan: access(trash_can)
Executing Plan
Instruction: A postdoc spilled their soda. Help them clean it up.

Action Generation with FMs

A. LLMs for task planning: SayCan approach

B. LLMs for scalable task planning in complex environments: SayPlan approach

C. Reward function generation: Language to Rewards (L2R)

D. Action generation: RT-1 and RT-2 models

THE UNIVERSITY of ADELAIDE

Language to Rewards for Robotic Skill Synthesis

Wenhao Yu,[*] Nimrod Gileadi[*], Chuyuan Fu,[†] Sean Kirmani[†], Kuang-Huei Lee[†], Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, Brian Ichter, Ted Xiao, Peng Xu, Andy Zeng, Tingnan Zhang, Nicolas Heess, Dorsa Sadigh, Jie Tan, Yuval Tassa, Fei Xia

Google DeepMind

https://language-to-reward.github.io/ [‡]

Abstract: Large language models (LLMs) have demonstrated exciting progress in acquiring diverse new capabilities through in-context learning, ranging from logical reasoning to code-writing. Robotics researchers have also explored using LLMs to advance the capabilities of robotic control. However, since low-level robot actions are hardware-dependent and underrepresented in LLM training corpora, existing efforts in applying LLMs to robotics have largely treated LLMs as semantic planners or relied on human-engineered control primitives to interface with the robot. On the other hand, reward functions are shown to be flexible representations that can be optimized for con-

THE UNIVERSITY of ADELAIDE

C. Reward Function Generation:

Language to Rewards (L2R)

- **Low-level robot actions** are hardware-dependent and underrepresented in LLM training data.
- Existing approaches often use **LLMs only as high-level semantic planners** or rely on **pre-engineered control primitives**, limiting their potential in robotic control.
- L2R is an approach that uses large language models to generate reward functions for robotic tasks based on natural language descriptions.

THE UNIVERSITY of ADELAIDE

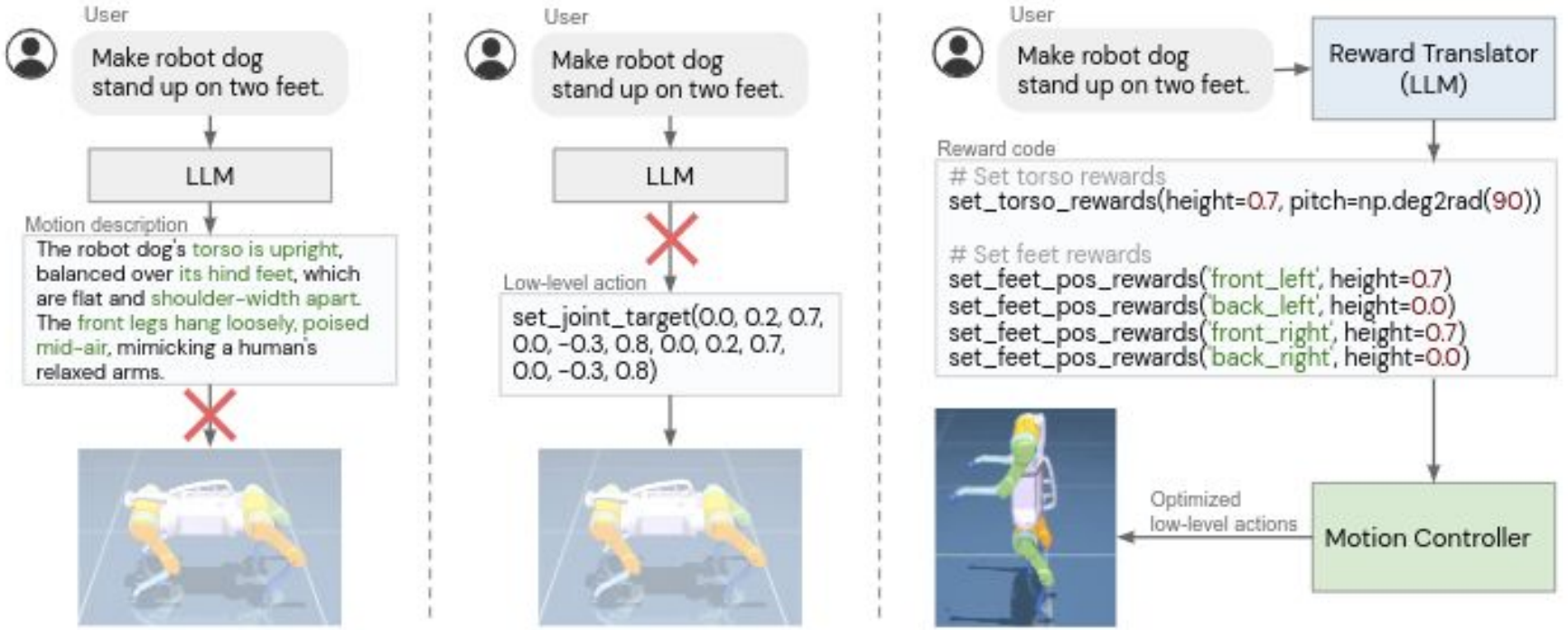
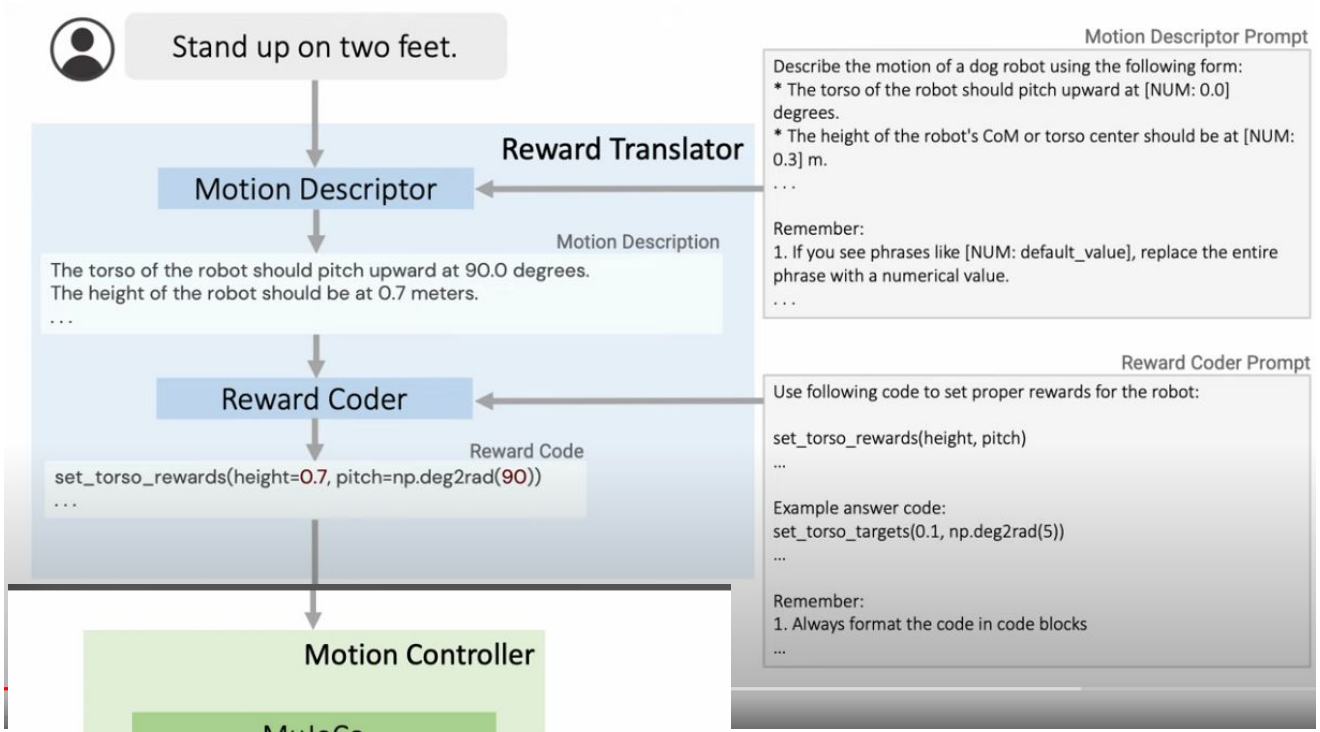


Figure 1: LLMs have some internal knowledge about robot motions, but cannot directly translate them into actions (left). Low-level action code can be executed on robots, but LLMs know little about them (mid). We attempt to bridge this gap, by proposing a system (right) consisting of the Reward Translatorthat interprets the user input and transform it into a reward specification. The reward specification is then consumed by a Motion Controller that interactively synthesizes a robot motion which optimizes the given reward.



THE UNIVERSITY of ADELAIDE


Instruction: Turn on faucet

Advantages of L2R

- **Flexibility:** Can handle a wide range of task descriptions.
- **Interpretability:** The language-based approach makes it easier for humans to understand and verify the reward functions.
- **Scalability:** Can leverage improvements in LLMs to handle more complex task specifications.

THE UNIVERSITY of ADELAIDE

Challenges and Considerations

- **Ambiguity:** Natural language can be ambiguous, which can lead to incorrect reward functions.
- **Safety:** Ensuring that generated reward functions don't lead to unsafe behaviors is crucial.
- **Generalization:** The system needs to handle a wide variety of task descriptions and environments.

THE UNIVERSITY of ADELAIDE

Action Generation with FMs

A. LLMs for task planning: SayCan approach

B. LLMs for scalable task planning in complex environments: SayPlan approach

C. Reward function generation: Language to Rewards (L2R)

D. Action generation: RT-1 and RT-2 models

THE UNIVERSITY of ADELAIDE

D. Action generation: RT-1 and RT-2 models

- RT stands for 'Robotic Transformer'
- A step closer towards end-to-end learning for robotic control.
- Aim to directly map language instructions and visual inputs to robot actions

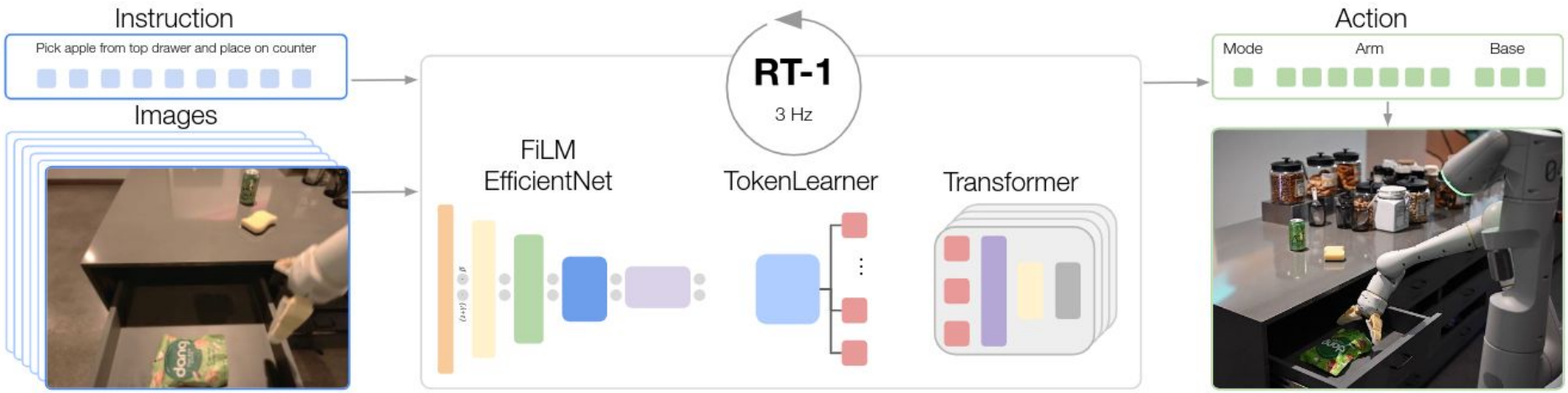
RT-1

- Trained on over 130,000 episodes of robot interaction data
- 13 robots over 17 months.
- Data includes a wide variety of objects, tasks, and environments
- Uses imitation learning to mimic expert demonstrations

Input and Output:

- **Inputs:** RGB images, language instructions, and robot state
- **Outputs:** Low-level control commands for the robot

THE UNIVERSITY of ADELAIDE



The **actions** consist of **seven dimensions** for the **arm movement (x, y, z, roll, pitch, yaw, opening of the gripper), three dimensions for base movement (x, y, yaw)** and a **discrete dimension to switch between three modes**:

- controlling the arm,
- the base,
- or terminating the episode.

RT-1 performs closed-loop control and commands actions **at 3 Hz** until it either yields a “terminate” action or hits a pre-set time step limit.

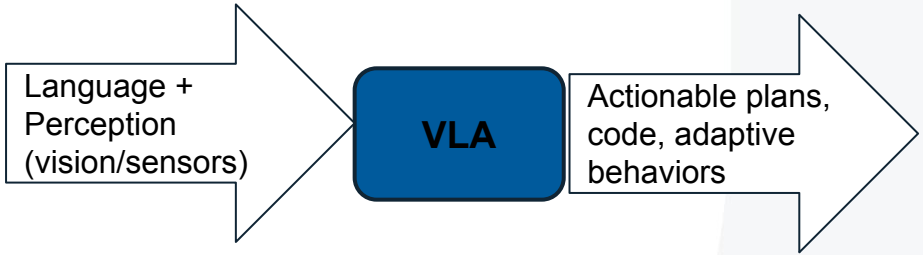
THE UNIVERSITY of ADELAIDE


RT-1

Vision-Language-Action (VLA) models

"Cognitive Engines" for Robots.

- **Integrate vision and language understanding to inform and control robotic actions.**



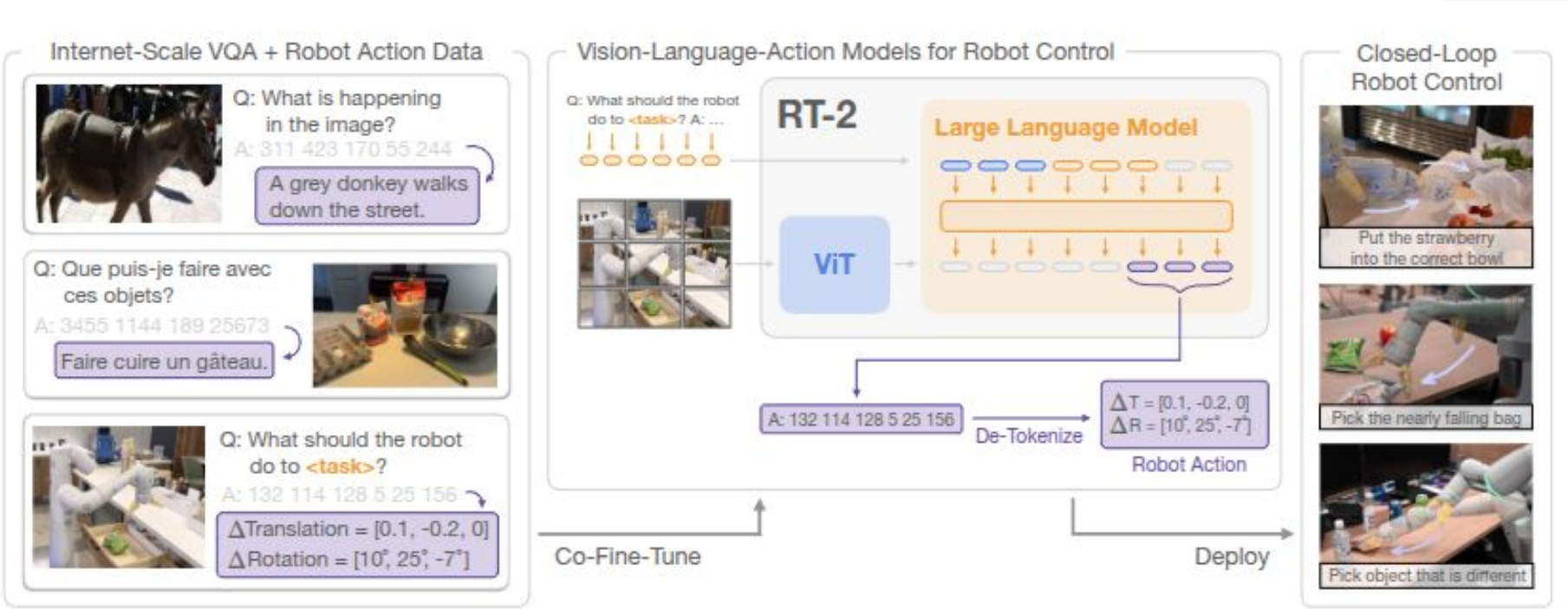
Key Capabilities

- Process visual and textual inputs to generate actionable outputs.
- Enhance robots' ability to interpret complex instructions and interact with their environment.

- Task Understanding
- Code and Action Generation
- Affordance Grounding
- Self-Reflection

THE UNIVERSITY of ADELAIDE

RT-2: Vision-Language-Action Models



THE UNIVERSITY of ADELAIDE

RT-2: Vision-Language-Action Models

Advancements over RT-1:

- Introduces the concept of **Vision-Language-Action (VLA)** models
- Leverages large-scale vision-language models pre-trained on internet data
- Aims for even greater generalization and flexibility

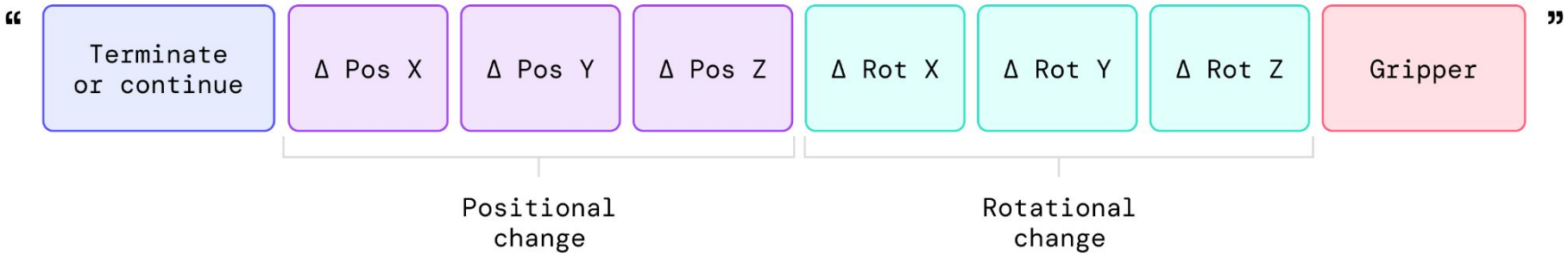
Key Ideas:

- Treats robot actions as another form of language
- Uses a large vision-language backbone (like PaLM-E) fine-tuned for robotics
- Enables more complex reasoning about tasks and environments

THE UNIVERSITY of ADELAIDE

Robot actions as text strings. An example of such a string could be a sequence of robot action token numbers:

“1 128 91 241 5 101 127 217”.



THE UNIVERSITY of ADELAIDE

Challenges and Limitations

Computational Requirements:

- Especially for RT-2, the large model size poses challenges for real-time operation

Data Requirements:

- Both models require large amounts of diverse robotics data

Safety Concerns:

- Ensuring safe operation in all scenarios remains a challenge

Interpretability:

- Understanding the decision-making process of these complex models can be difficult

THE UNIVERSITY of ADELAIDE

What might come next?

Models that **act** and **learn** in the **physical world** by integrating **sensory inputs, language, and action**, enabling **fast dynamic interaction** with environment and users.

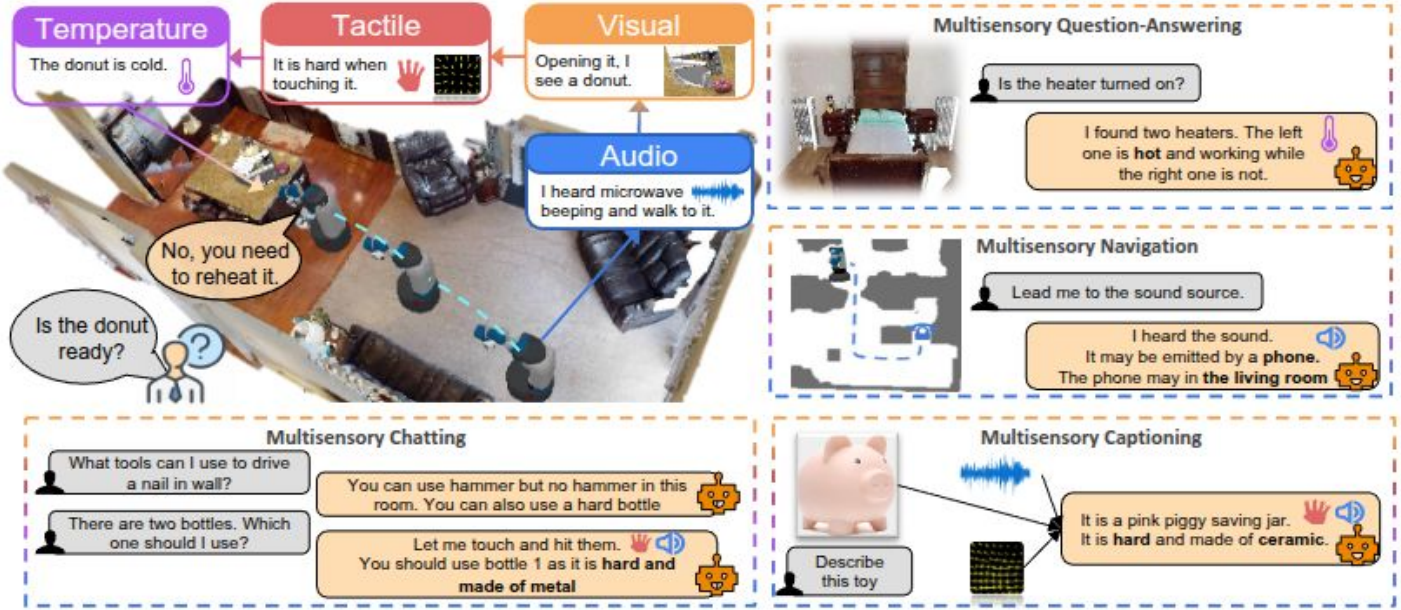
- **Low-latency Full-Sensory Integration:** Beyond text, images, and sensors to audio, tactile feedback, smell, and proprioception.
- **World Models:** Internal simulations of how objects behave to predict outcomes of actions with 3D Spatial Understanding.
- **Active Learning:** learn by doing.
- **Causal and Counterfactual Reasoning:** inferring cause-effect relationships.
- **Personalized Context Awareness.**

THE UNIVERSITY of ADELAIDE

MultiPLY: A Multisensory Object-Centric Embodied Large Language Model in 3D World

Yining Hong[2,3], Zishuo Zheng[1], Peihao Chen[1], Yian Wang[1], Junyan Li[1], Chuang Gan[1,3]
[1]UMass Amherst, [2] UCLA, [3]MIT-IBM Watson AI Lab

https://vis-www.cs.umass.edu/multiply



THE UNIVERSITY of ADELAIDE


THE UNIVERSITY
of ADELAIDE
Australian
Institute for
Machine Learning
Q&A
adelaide.edu.au/aiml