Towards a Robotics Foundation Model

Keerthana Gopalakrishnan
March 3, 2023

Robotics at Google

Agenda

01 **The Ingredients for a Robotics Foundation Model**

02 RT-1

03 SayCan

04 Inner Monologue

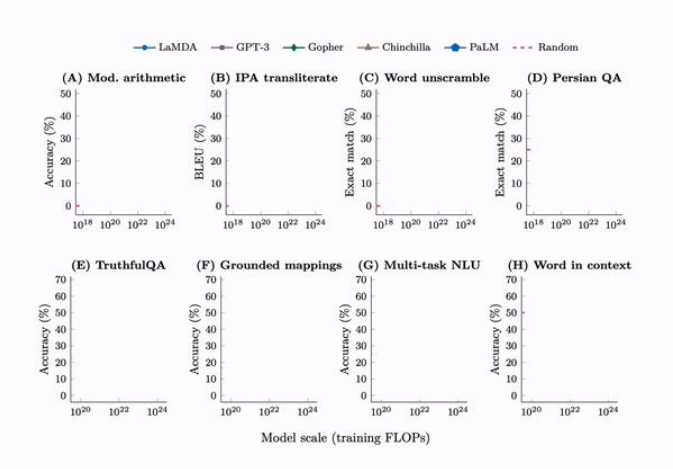
05 NL Map Saycan

06 MOO

07 What's next?

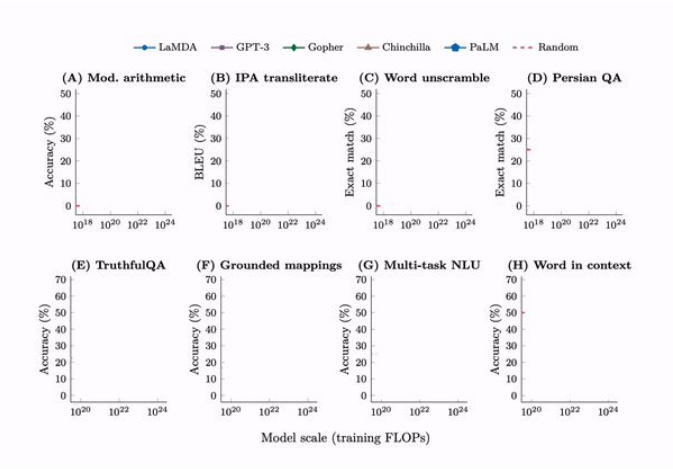
Why a Foundation Model for Robotics?

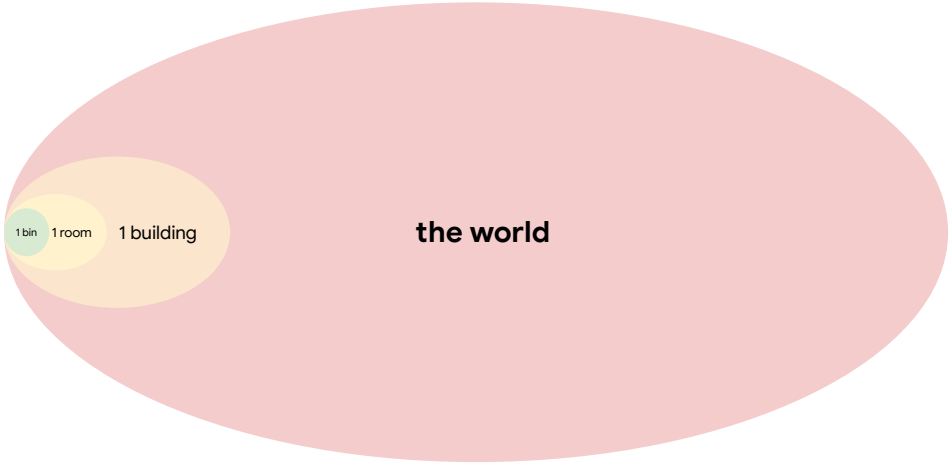
- Foundation models enable *emergent capabilities* and *homogenization*
  - *emergent capabilities:* emergence of more complex behavior not present in smaller models
  - *homogenization*: generalization to combinatorially many downstream use cases
  - More is Different for AI, Emergence in LLMs



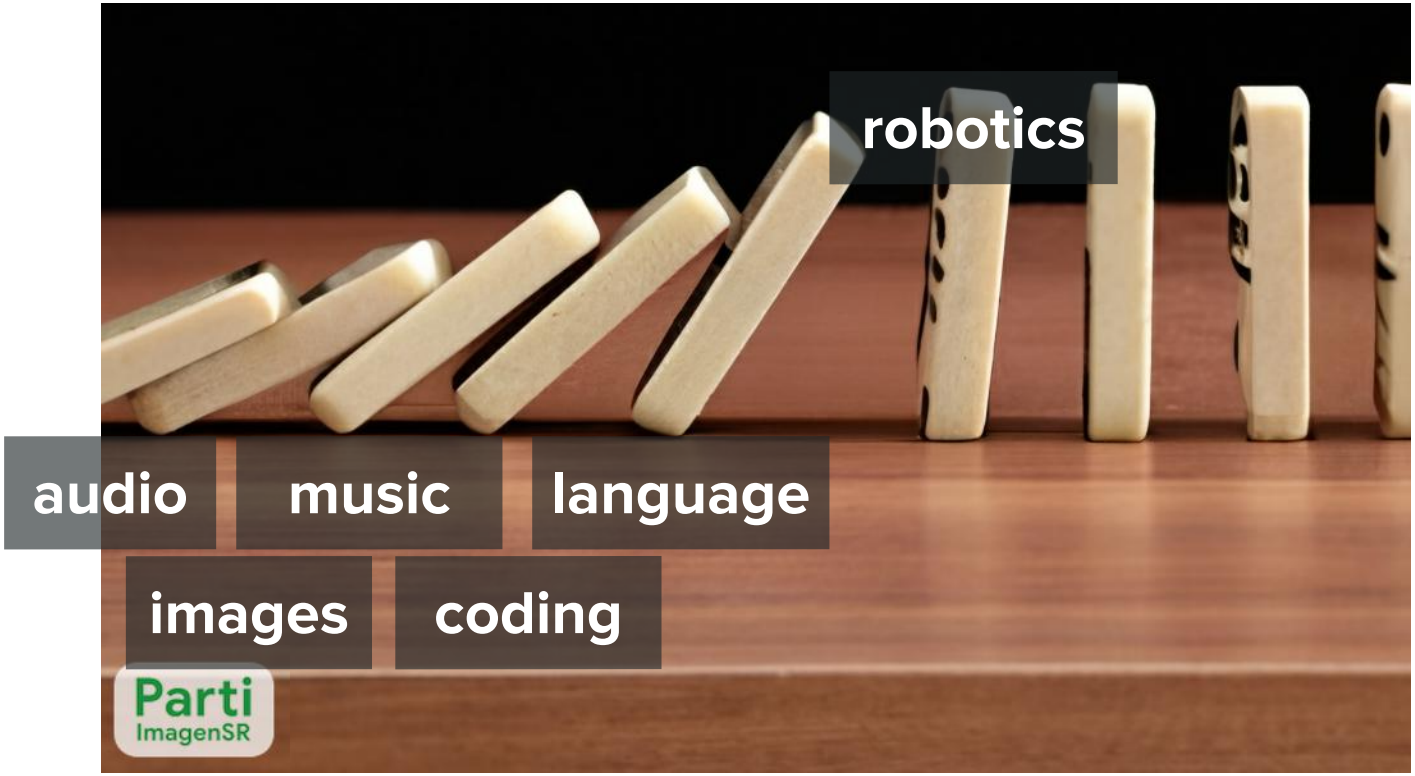
Why a Foundation Model for Robotics?

- Foundation models enable *emergent capabilities* and *homogenization*
  - *emergent capabilities:* emergence of more complex behavior not present in smaller models
  - *homogenization*: generalization to combinatorially many downstream use cases
  - More is Different for AI, Emergence in LLMs
- An “emergent capabilities” curve might be *required* for robotics to be useful

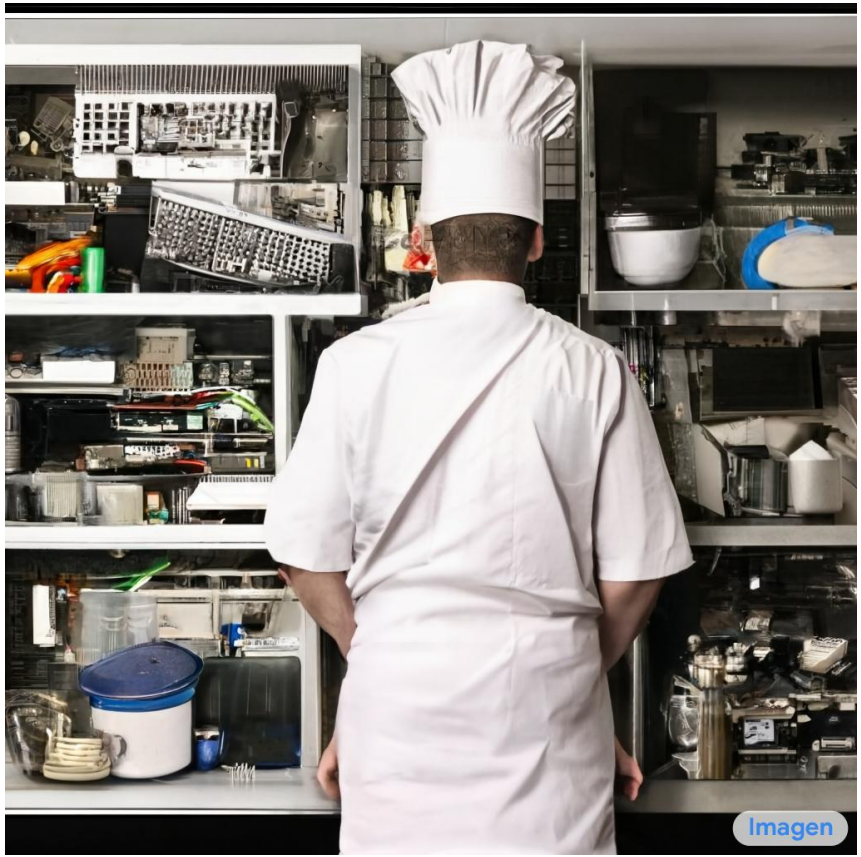




Why *not* a Foundation Model for Robotics?

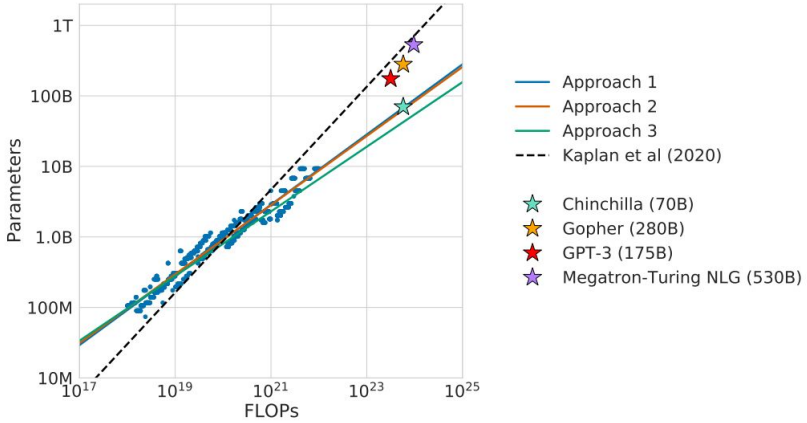


What are the ingredients for a Robotics Foundation Model?



Ingredient #1: Design Principles of ML Scaling

- High-capacity architectures, ie. self-attention
- Scaling params and compute and corpus size (tokens)
- Dataset size matters more than quality

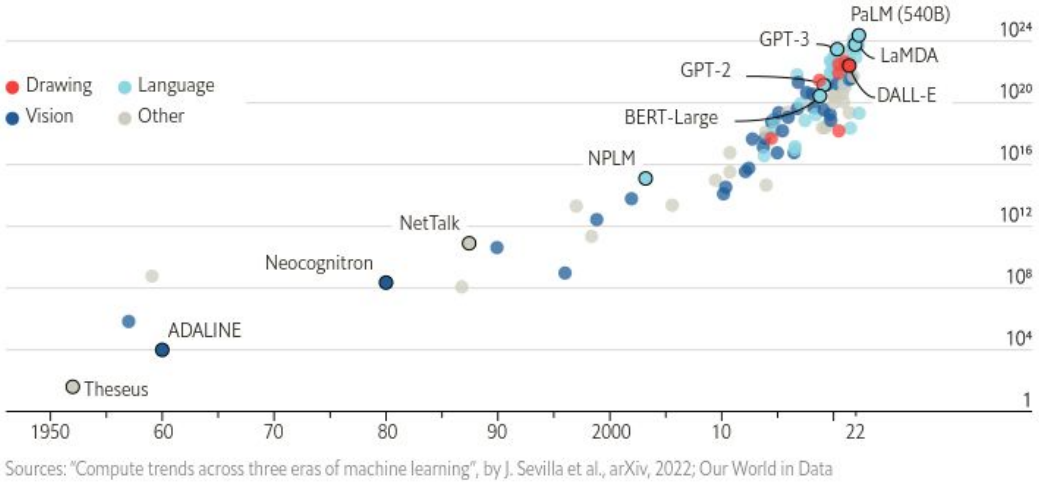


Andrej Karpa... @karpa... · Oct 19, 2022

The Transformer is a magnificient neural network architecture because it is a general-purpose differentiable computer. It is simultaneously:
1) expressive (in the forward pass)
2) optimizable (via backpropagation+gradient descent)
3) efficient (high parallelism compute graph)

Ingredient #2: Proliferation of Internet-scale Models

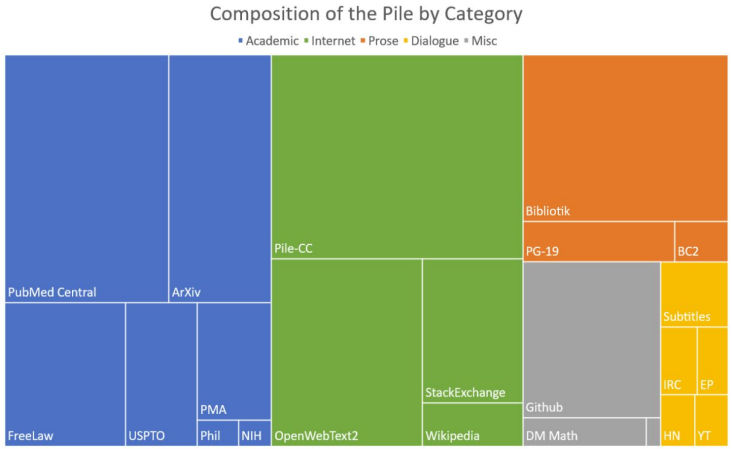
- Generative models in {language, coding, vision, audio, ...} experience *emergent capabilities*
- Proliferation + acceleration means these models will get better “on their own” over time



Sources: “Compute trends across three eras of machine learning”, by J. Sevilla et al., arXiv, 2022; Our World in Data

Ingredient #3: Robotics moves from Online to Offline

- Foundation models train on huge amounts of diverse offline datasets



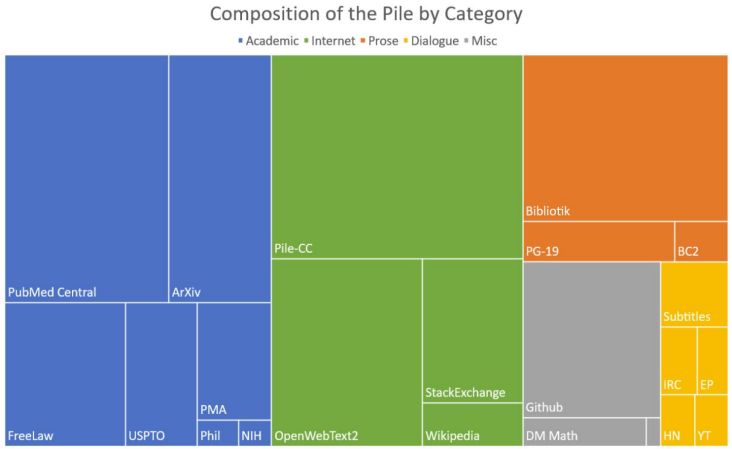
The Pile dataset (Eleuther)

Dataset	# English Img-Txt Pairs
Public Datasets	
MS-COCO	330K
CC3M	3M
Visual Genome	5.4M
WIT	5.5M
CC12M	12M
RedCaps	12M
YFCC100M	100M[2]
LAION-5B (Ours)	**2.3B**
Private Datasets	
CLIP WIT (OpenAI)	400M
ALIGN	1.8B
BASIC	6.6B

LAION-5B (Laion)

Ingredient #3: Robotics moves from Online to **Offline**

- Foundation models train on huge amounts of diverse offline datasets

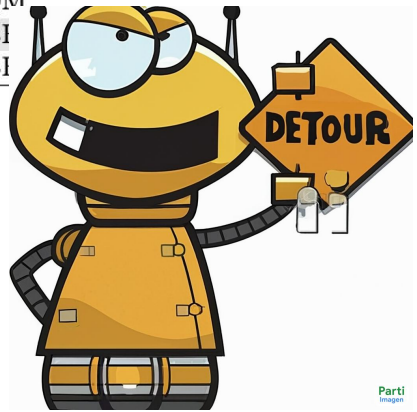


The Pile dataset (Eleuther)

Dataset	# English Img-Txt Pairs
Public Datasets	
MS-COCO	330K
CC3M	3M
Visual Genome	5.4M
WIT	5.5M
CC12M	12M
RedCaps	12M
YFCC100M	100M[2]
LAION-5B (Ours)	**2.3B**
Private Datasets	
CLIP WIT (OpenAI)	400M
ALIGN	1.8I
BASIC	6.6I

LAION-5B (Laion)

- **Robot learning classically does a lot of online on-policy learning (RL!) but winds have been shifting**



Detour: Brief History of Robotics at Google Brain

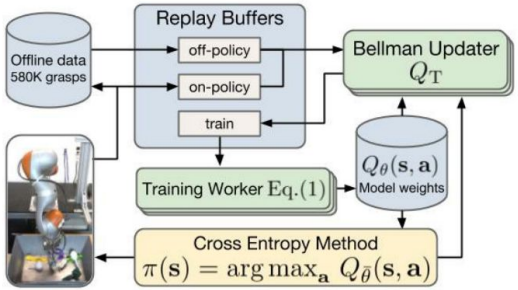
goal / task conditioning

off-policy evaluation

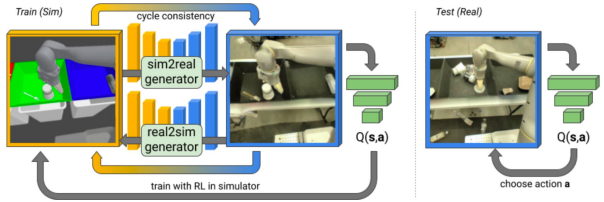
QT-Opt

real-world robotics

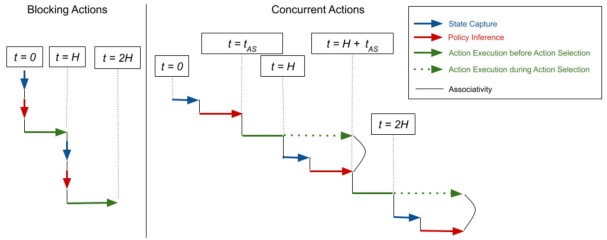
sim2real / CGAN



QT-Opt



RL-CycleGAN



Concurrent Control

2016 - 2020

“How do we do end-to-end robot learning in the real world?”

Detour: Brief History of Robotics at Google Brain

goal / task conditioning

off-policy evaluation

QT-Opt

real-world robotics

sim2real / CGAN

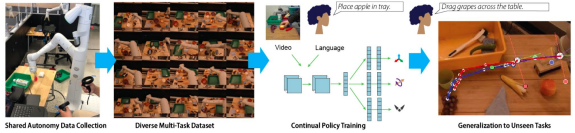
sim2real bootstrapping

BC-0

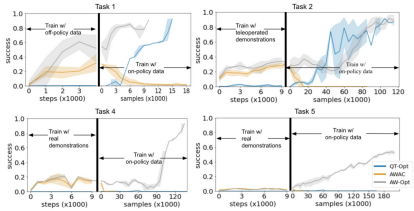
IL+RL

MT-Opt

Scaling real operations



BC-Z



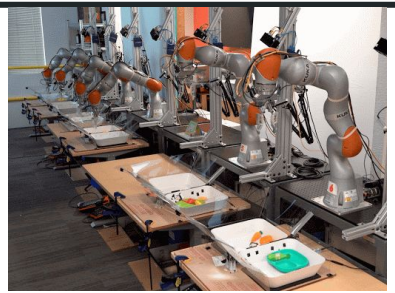
AW-Opt

2016 - 2020

"How do we do end-to-end robot learning in the real world?"

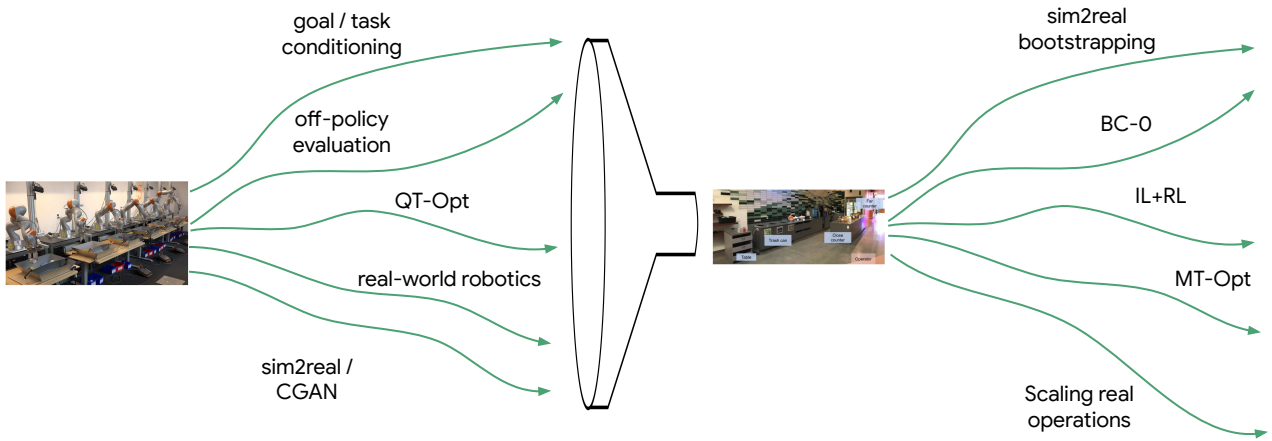
2020 - 2022

"How do we scale to many tasks in more complex scenarios?"



MT-Opt

Detour: Brief History of Robotics at Google Brain



#1: Some methods plateaued at 50-70% success rates
#2: Some methods required very specific data distributions

How do we satisfy both:
#1 Solve many tasks (>90%)
#2 Improve with “scalable” data

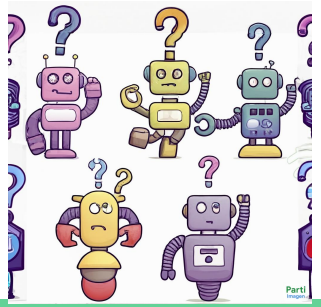
2016 - 2020

“How do we do end-to-end robot learning in the real world?”

2020 - 2022

“How do we scale to many tasks in more complex scenarios?”

Spring 2022



Detour: Brief History of Robotics at Google Brain

goal / task conditioning

off-policy evaluation

QT-Opt

real-world robotics

sim2real / CGAN

sim2real bootstrapping

BC-0

IL+RL

MT-Opt

Scaling real operations

Multi-task Imitation Learning

- It achieves >90%!
- It scales with teleoperated demonstrations!

2016 - 2020

"How do we do end-to-end robot learning in the real world?"

2020 - 2022

"How do we scale to many tasks in more complex scenarios?"

Spring 2022

Detour: Brief History of Robotics at Google Brain

goal / task conditioning

off-policy evaluation

QT-Opt

real-world robotics

sim2real / CGAN

sim2real bootstrapping

BC-0

IL+RL

MT-Opt

Scaling real operations

Multi-task Imitation Learning

- It achieves >90%!
- It scales with teleoperated demonstrations!

2016 - 2020

"How do we do end-to-end robot learning in the real world?"

2020 - 2022

"How do we scale to many tasks in more complex scenarios?"

Spring 2022

We went from focusing on online methods to focusing on offline methods. We decoupled data generation from data consumption.

Turning Ingredients into a Recipe

Ingredients

Lessons

Turning Ingredients into a Recipe

Ingredients

Design Principles of ML Scaling

Lessons

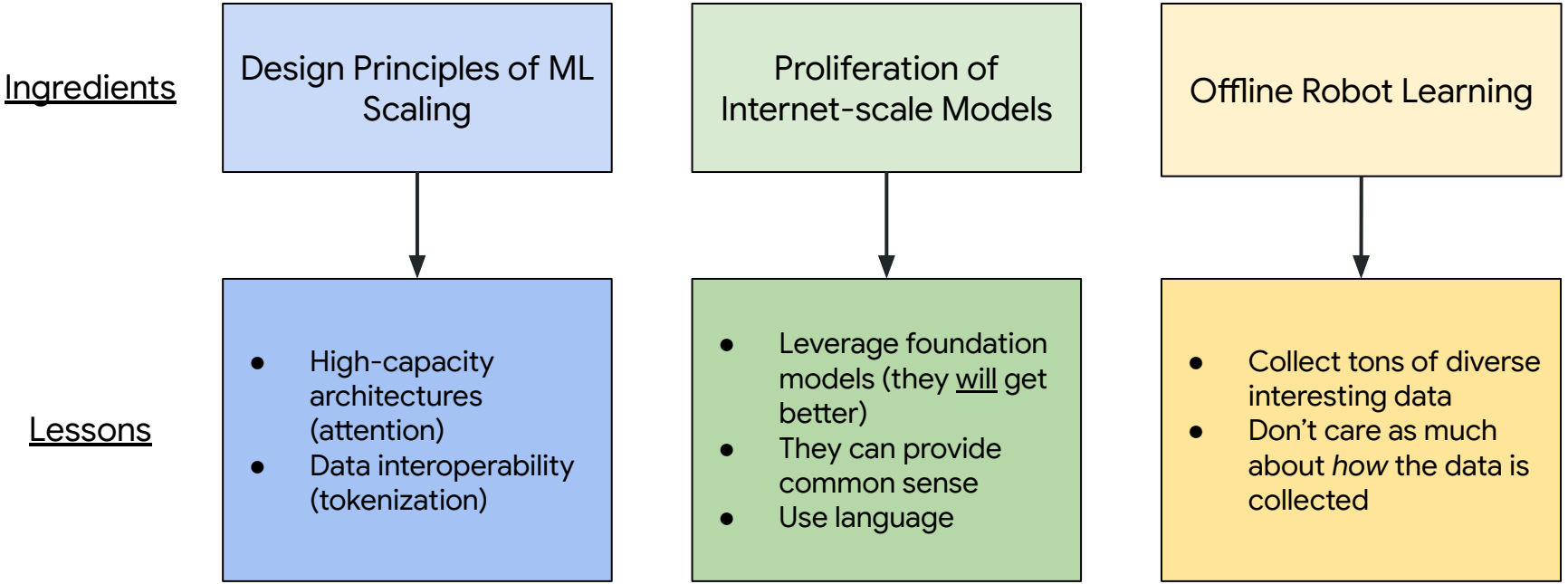
- High-capacity architectures (attention)
- Data interoperability (tokenization)

Turning Ingredients into a Recipe

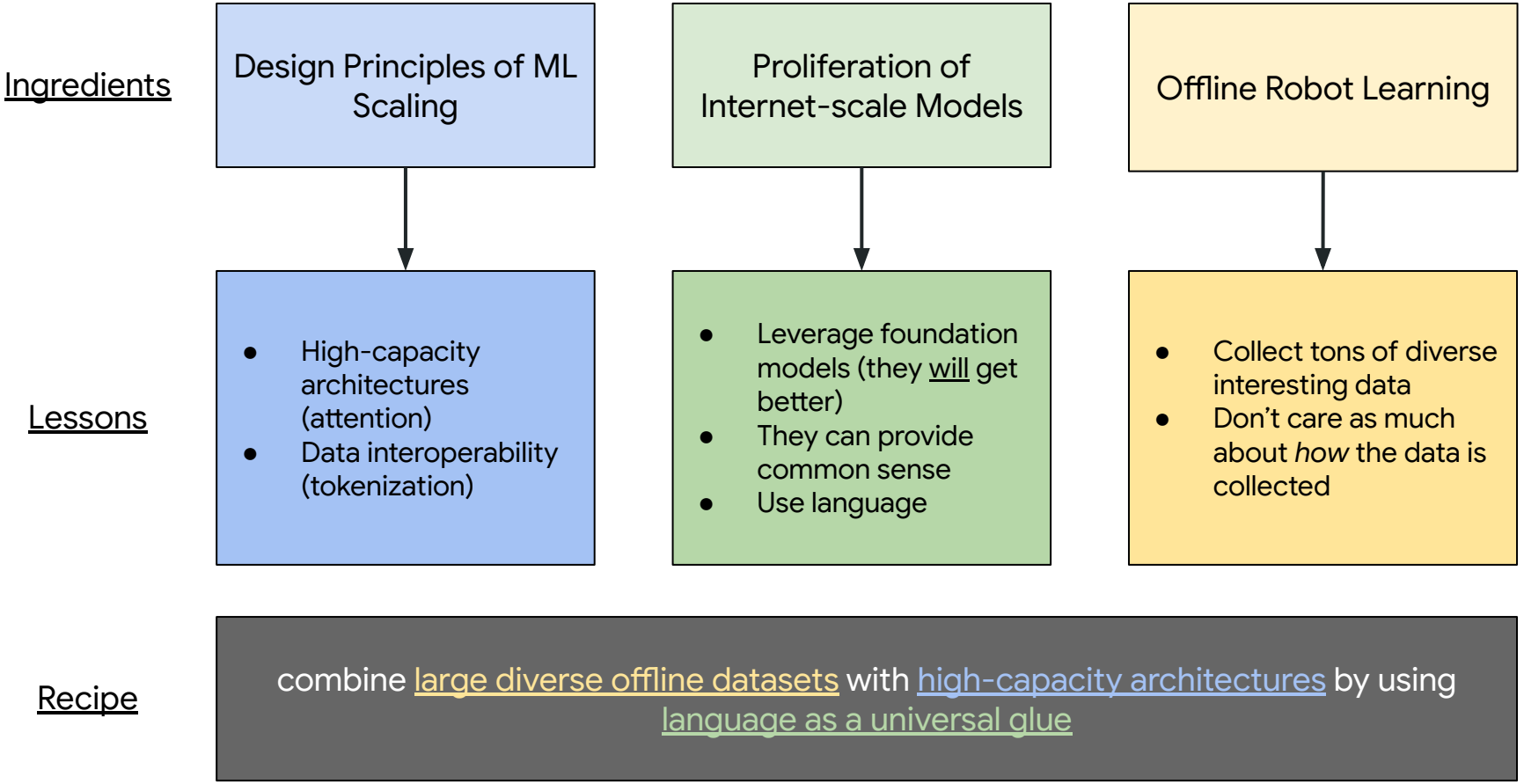
Ingredients	Design Principles of ML Scaling	Proliferation of Internet-scale Models
Lessons	• High-capacity architectures (attention) • Data interoperability (tokenization)	• Leverage foundation models (they will get better) • They can provide common sense • Use language

Bitter Lesson 2.0:
"Leverage methods that scale with better foundation models!"

Turning Ingredients into a Recipe



Turning Ingredients into a Recipe



Agenda

01 The Ingredients for a Robotics Foundation Model

02 **RT-1**

03 SayCan

04 Inner Monologue

05 NL Map Saycan

06 MOO

07 What's next?

Scaling Multi-task Imitation from First Principles

Existing Demonstration Dataset

- O(100k) demos from 13 robots, 17 months, 700 tasks

Limited throughput of new robotic data

- Autonomous collect is costly (engineering) [MT-Opt, RGB-Stacking]
- Human teleoperation is costly (operations) [BC-Z]

ResNet-50 Works Reasonably

- Still sensitive to training distributions [BC-Z]

Lesson #3: Offline Robot Learning

Models need to be robust and generalize

Scaling Multi-task Imitation from First Principles

Models need to be robust and generalize

Off-the-shelf models are too slow

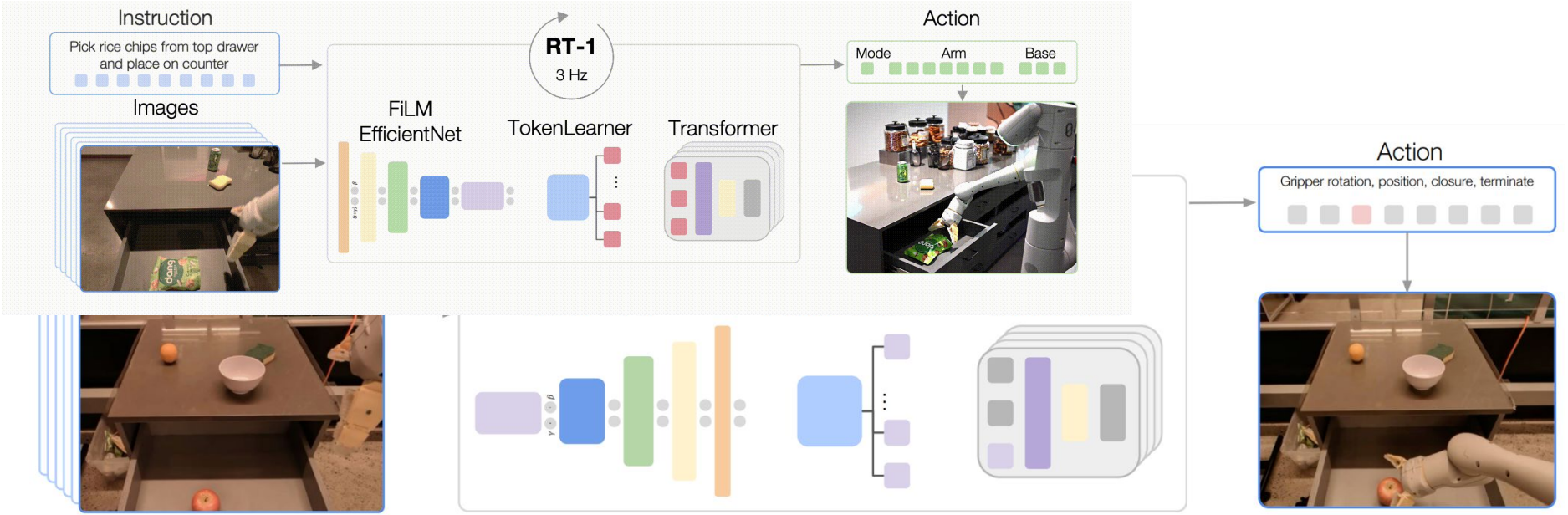
- We need to be able to run vision-based inference in the real world

Our data is language conditioned

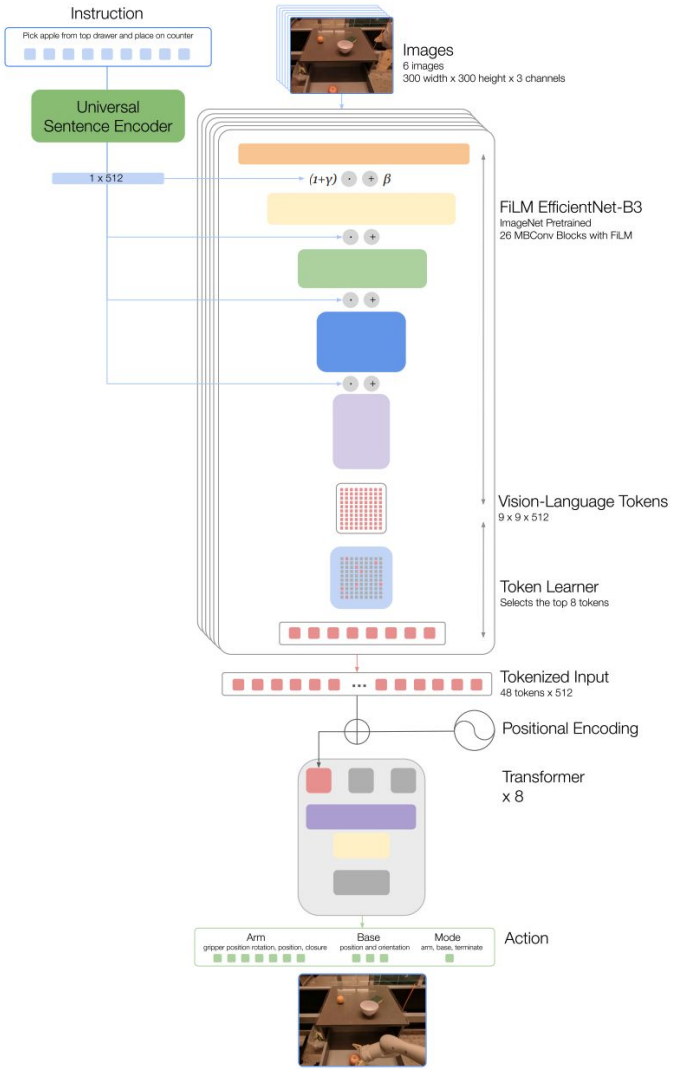
- We need to support multi-modality [SayCan]

Lesson #1: Design Principles of ML Scaling

We need to create a new architecture with attention and tokenization



- Tokenized input and outputs
- Decoder only transformer, sparse categorical entropy objective
- Image tokenizer: Pre-trained film efficient net backbone
- Token learner for compression/ faster inference

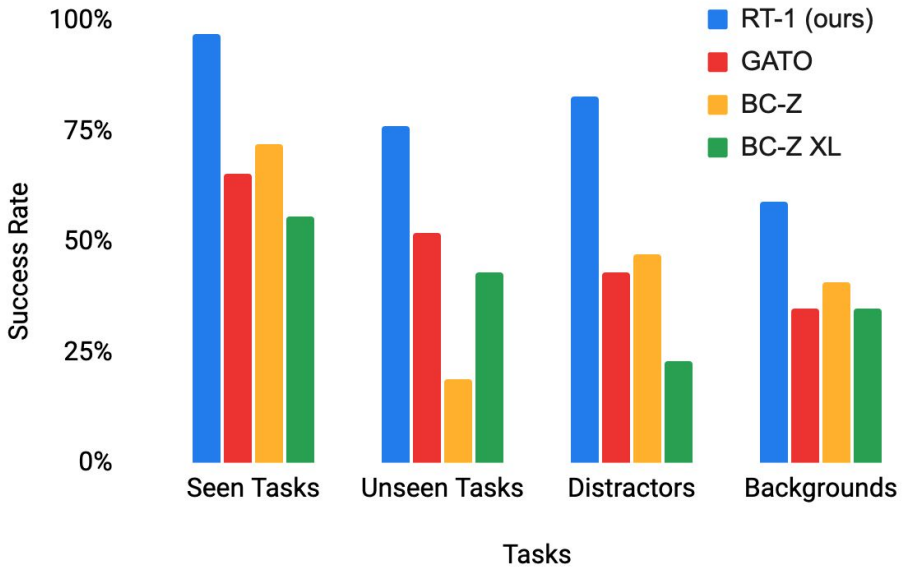


RT-1 Design Takeaways

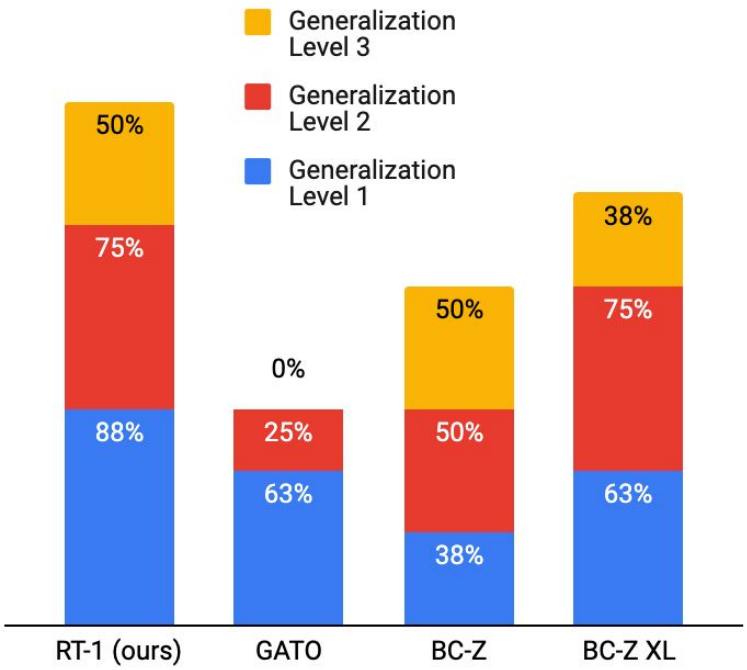
- Inference budget: 100ms (3Hz control)
- 6-image history
- TokenLearner subsamples 81 image patches into 8 image patches
- Action discretization to 256 bins
- Model Size: 35M parameters

Pushing the limit: adding visual and semantic diversity

- RT-1 has the best performance compared to baselines in seen/unseen
- RT-1 is robust to variations in backgrounds, distractors



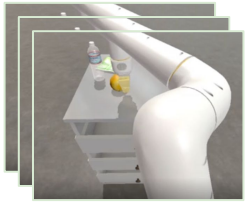
Pushing the limit: change multiple factors of variation



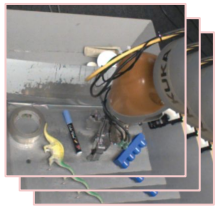
Pushing the limit: Training on Diverse Data Distributions

RT-1 data collected on Everyday Robots

RT-1 data collected in Sim and with Sim2Real

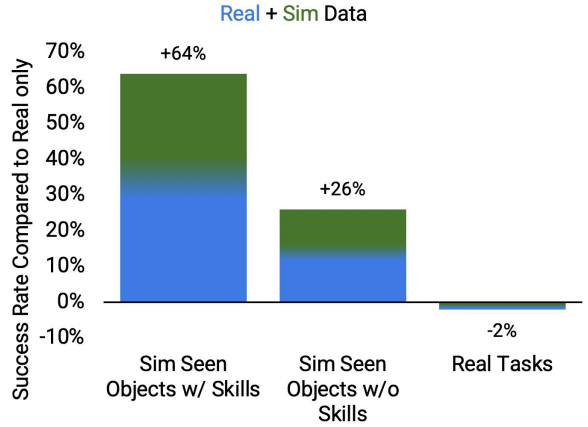


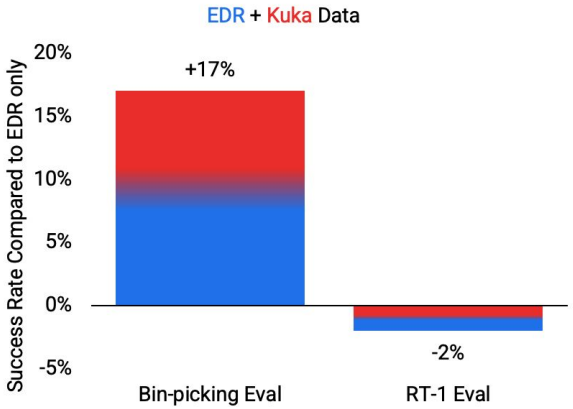
Bin-picking data collected on Kuka



RT-1

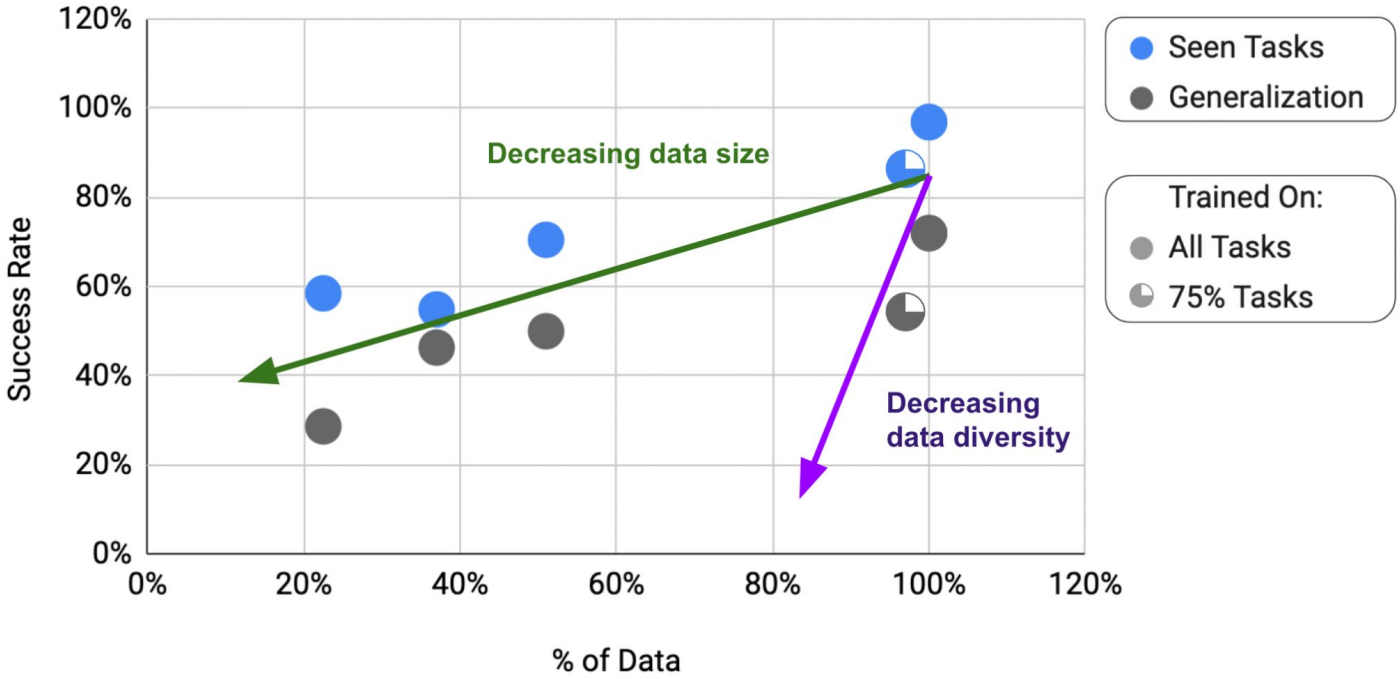
Real RT-1 eval





Scaling with data

- Reducing data size reduces performance and generalization
- Task diversity >> data size



Agenda

01 The Ingredients for a Robotics Foundation Model

02 RT-1

03 **SayCan**

04 Inner Monologue

05 NL Map Saycan

06 MOO

07 What's next

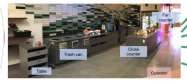
goal / task conditioning

off-policy evaluation

QT-Opt

multi-task RL

sim2real / CGAN



MT-Opt

BC-0

IL+RL

O(10^5) demos

sim2real bootstrapping

SayCan

Inner Monologue

DIAL

NLMap

Moo

Code as Policies

2016 - 2021

“How do we do end-to-end robot learning in the real world?”

2021 - 2022

“How do we scale to many tasks with demonstrations?”

2022 - 2023

“How can we leverage foundation models to accelerate robotics?”

Language models for robotics

Problem 1: Our robots can only do a fixed number of commands.

We need to get LLMs to speak “robot language”!

Problem 2: LLMs aren’t grounded in the real-world. They can’t “see”.

We need to ground LLMs in robotic affordances!

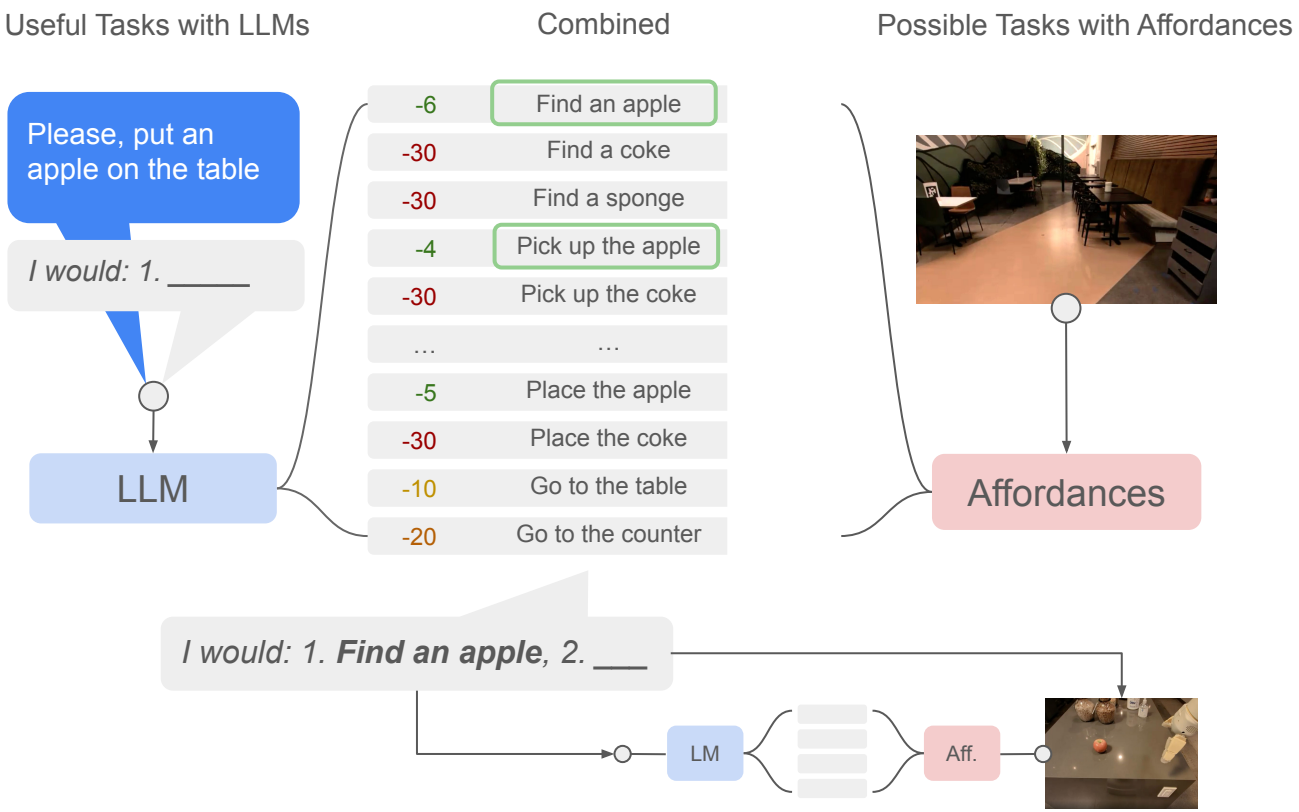
I spilled my drink, can you help?

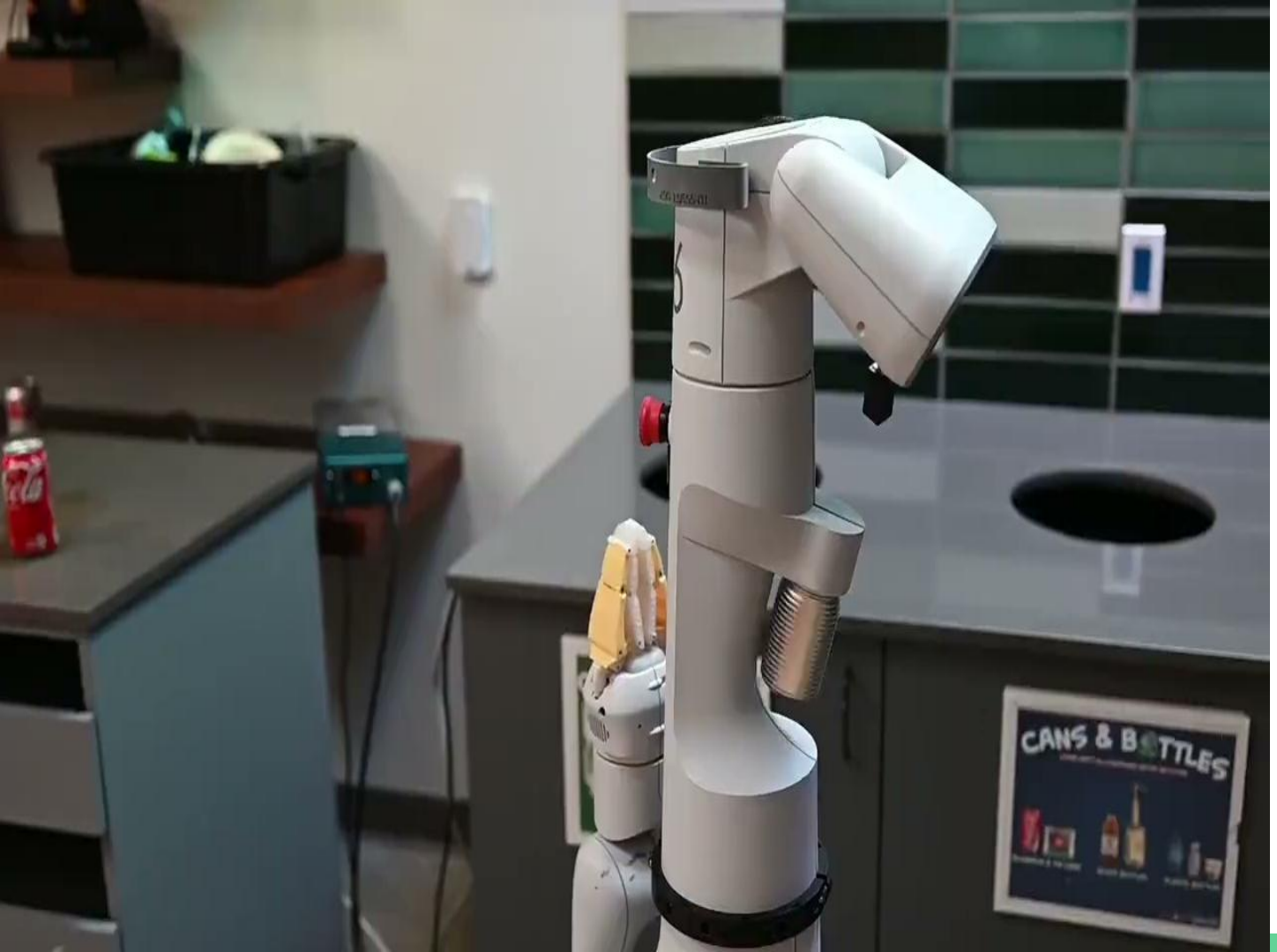
LLM1 You could try using a vacuum cleaner.

LLM2 Do you want me to find a cleaner?

LLM3 I'm sorry, I didn't mean to spill it.

SayCan: LLMs for robotics and robotics for LLMs





say-can.github.io

Experiment Overview

84% planning rate

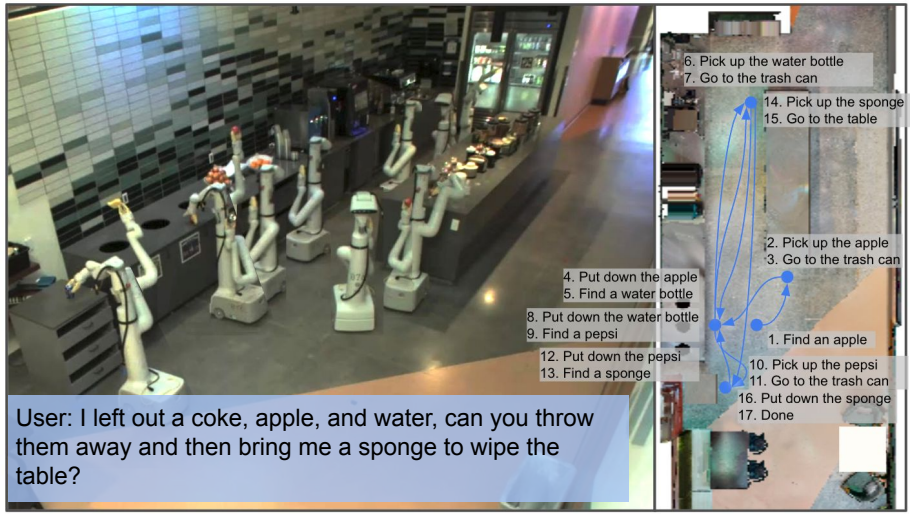
74% execution rate

101 long-horizon instructions

10+ navigation and manipulation skills in a row

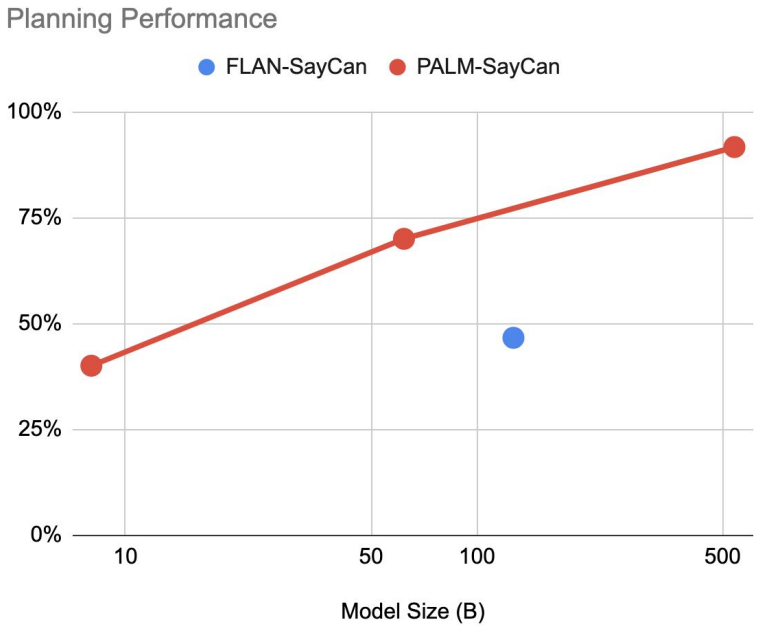
Without grounding nearly halves performance

Instruction Family	Num	Plan	Execution
Natural Language Single Primitive	15	100%	100%
Natural Language Nouns	15	67%	47%
Natural Language Verbs	15	100%	93%
Structured Language	15	93%	87%
Embodiment	11	64%	55%
Crowd Sourced	15	87%	87%
Long-Horizon	15	73%	47%
Total	**101**	**84%**	**74%**



PaLM-SayCan

Robotics performance scales with better LLMs!



Proliferation of Internet-scale Models

- Leverage foundation models (they will get better)
- They can provide common sense
- Use language

PaLM-SayCan

Robotics performance scales with better LLMs!

- Chain-of-thought prompting
- Better prompts

Solves all kinds of queries:

- I'm thirsty
- Bring me two different sodas
- I left out a coke, apple, and water, can you throw them away and then bring me a sponge to wipe the table?
- 你能给我拿杯可乐吗？

Human Input: How would you bring me a fruit that is not an apple?
Model Output:
Explanation: The user has requested me to bring a fruit that is not an apple. The banana is a fruit that is not an apple, I will bring the user a banana.
Robot: I would
1. find a banana
2. pick up the banana
3. go to the table
4. put down the banana
5. done.

Agenda

01 The Ingredients for a Robotics Foundation Model

02 RT-1

03 SayCan

04 **Inner Monologue**

05 NL Map Saycan

06 MOO

07 What's next

Language as connective tissue for robotics

Task

“Bring me a snack”

“Bring me a snack”

World States

“Apple in drawer”
“Orange in drawer”
“Coke in fridge”
“Top drawer open”
“Nothing on table”

LLM Planner

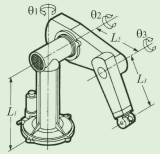
“Go to the user”

Policy

π_ϕ

Actions

$\mathbf{a}$



“Banana in gripper”
“Near the cabinet”

Robot States:

innermonologue.github.io

Inner Monologue

- Close the loop further with the scene
- Scene understanding
- And human feedback for free

Kitchen Mobile Manipulation (Real)

Human: Bring me a drink from the table
Robot: go to the table
Scene: I see a coke and lime soda
Robot: What kind of drink would you like?
Human: Something with caffeine
Robot: Pick up coke
Success Detector: False
Robot: Pick up coke
Robot: Bring it to you
Robot: Put down the coke
Robot: Done

Object Recognition

Success Detection

Agenda

01 The Ingredients for a Robotics Foundation Model

02 RT-1

03 SayCan

04 Inner Monologue

05 **NL Map Saycan**

06 MOO

07 What's next

Language as connective tissue for robotics

Task

“Bring me a snack”

“Bring me a snack”

World States

LLM Planner

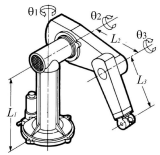
“Go to the user”

Policy

π_ϕ

Actions

$\mathbf{a}$



“Banana in gripper”

“Near the cabinet”

Robot States:

NLMap

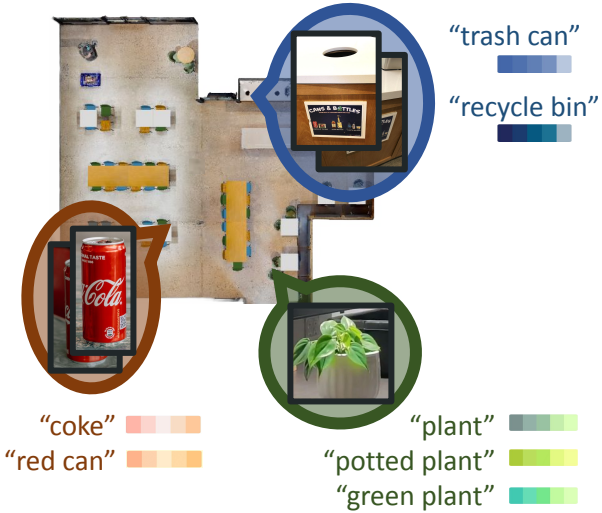
nlmap-saycan.github.io

Queryable, natural language map provides world context to the planner.

Scene Exploration

Region Proposal + VLM Feature
+ Multiview Fusion

Open-vocabulary Queryable Representation



Instruction

“Recycle the coke can”

Object Proposal

“coke can”, “recycle bin”

Query Context

“coke can” found at (x1, y1)
“recycle bin” found at (x2, y2)

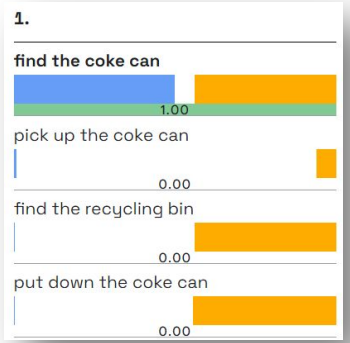
Generate Executable Options

“go to coke can”, “pick up coke can”,
“put down coke can”, ….

Context Aware Planning

Scene: coke can, recycle bin
Robot: I should

1. find the coke can
2. pick up coke can
3. go to recycle bin
4. put down coke can



Use Foundation Models for Importing Common Sense

“The bottleneck for robotics is high level semantic planning”
- Robotics Professor in 2018

“No.”
- LLMs in 2022

Lesson #2: Leverage Foundation Models

If language is the universal API of the system, then foundation models can inject common sense

Agenda

01 The Ingredients for a Robotics Foundation Model

02 RT-1

03 SayCan

04 Inner Monologue

05 NL Map Saycan

06 **MOO**

07 What's next?

Two Axis for Generalization

- Tasks

Examples:

Pick up X

Move X near Y

Open the lid on X

Place X into the Y

- Objects

Examples:

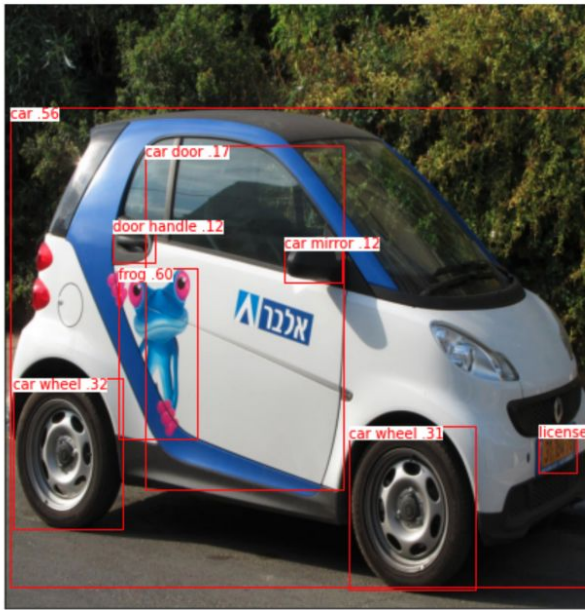
Pick up the apple core

Place the screws into the box

Move the coffee near the human

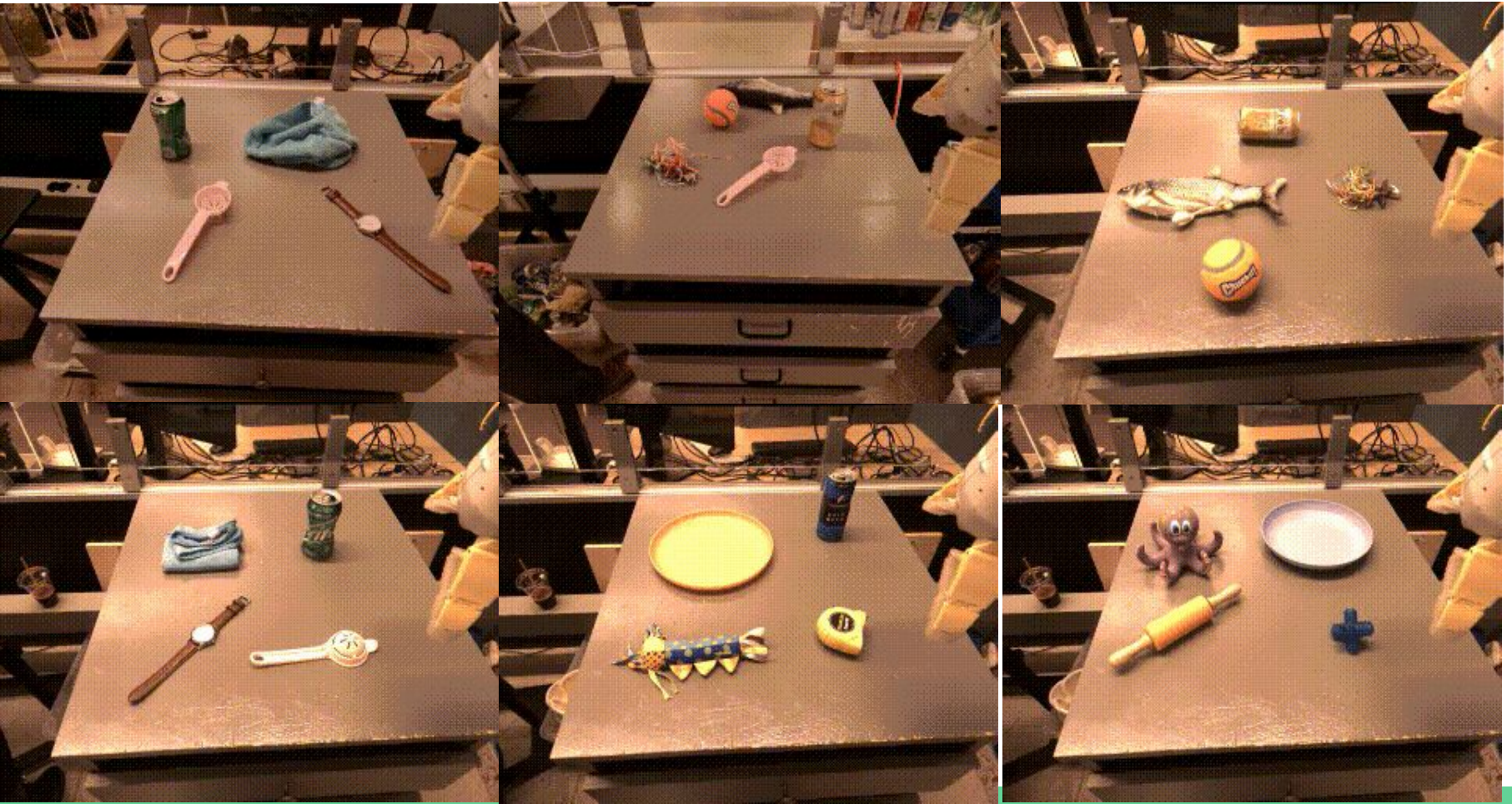
Open the lid of the large box.

Vision Language Models can zero-shot detect almost any object.



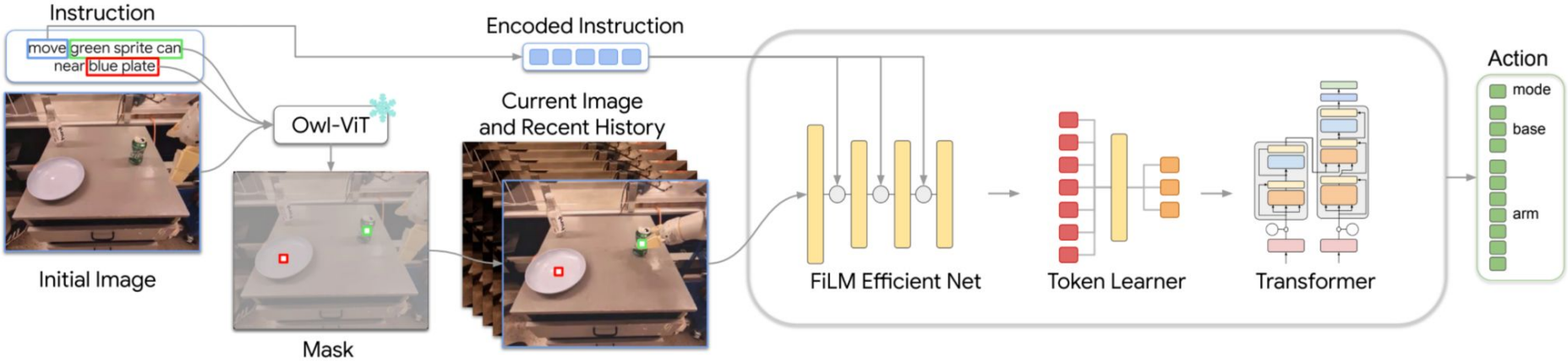
(Real detections from OWL-ViT)

Generalizing to any object type

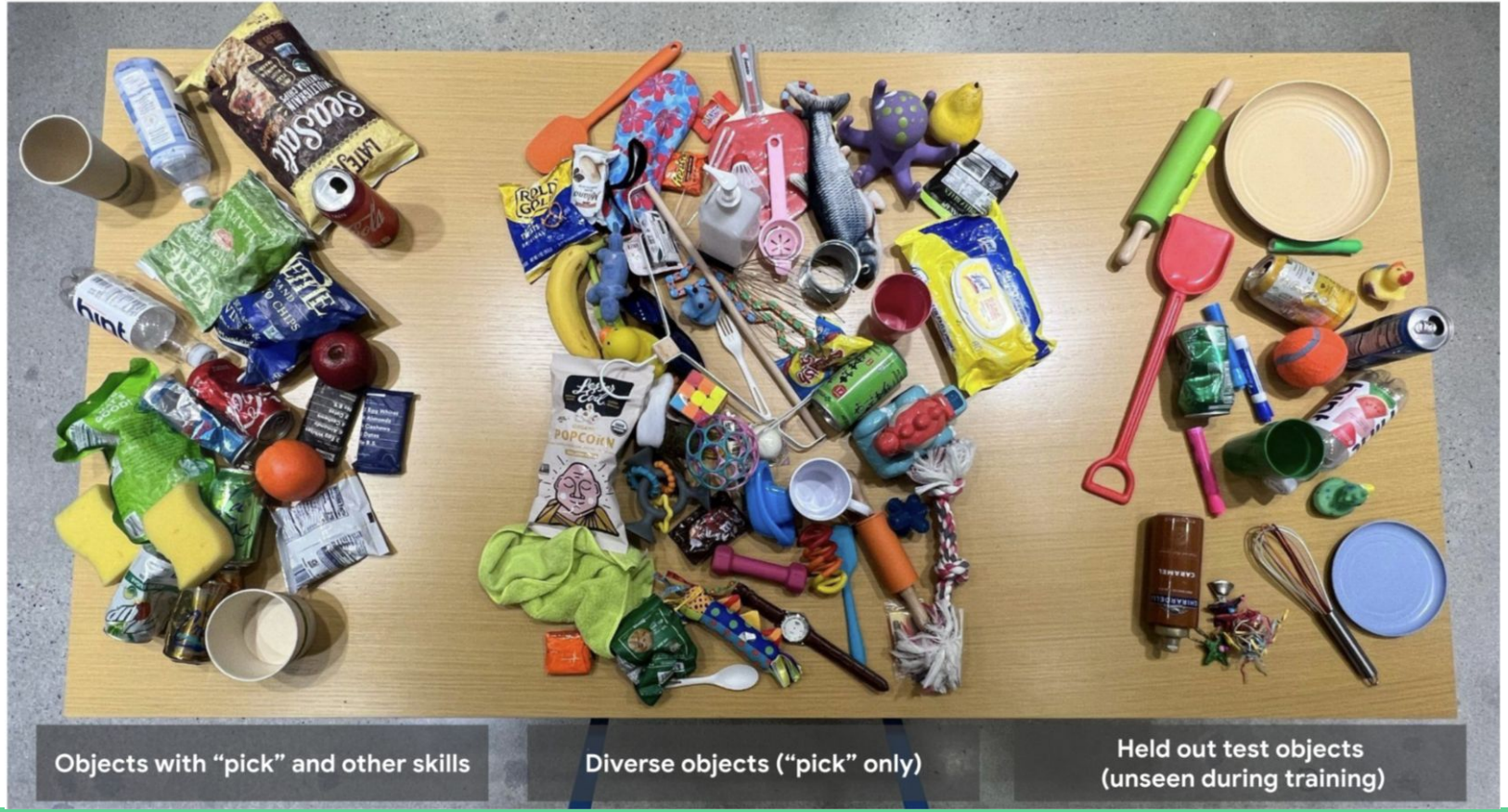


Interface is natural language

Extract objects from phrase and feed them to OWL-ViT, feed detections as input along with "task embedding."

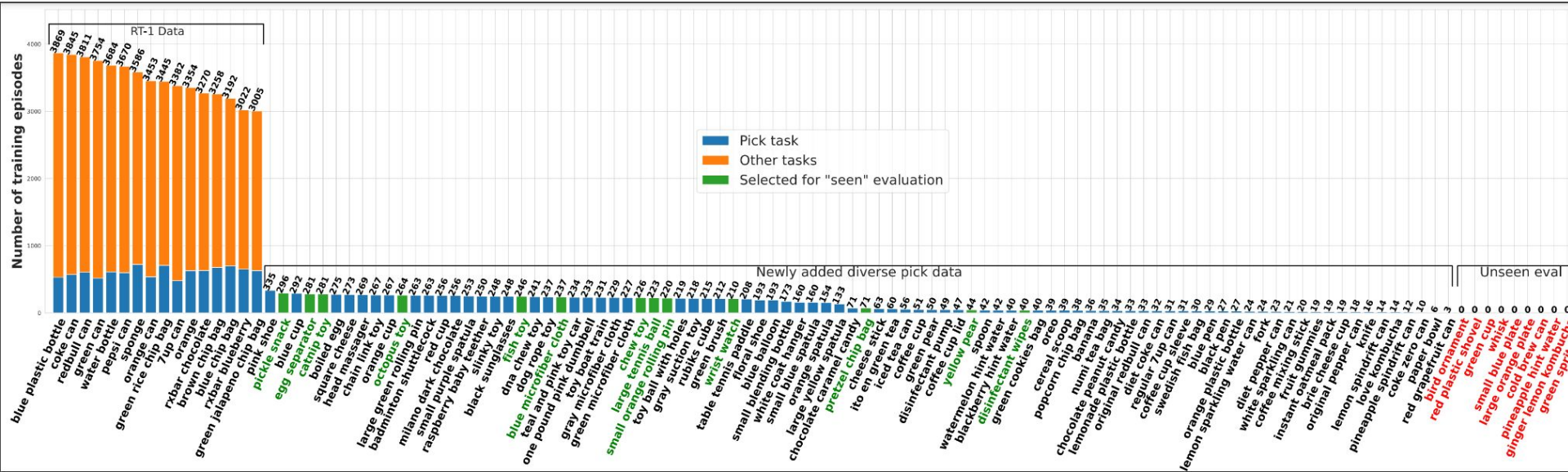


Train on enough objects the policy learns generalizable skills



Distribution of Objects

We added small amounts of "pick" skill data for 100 additional, diverse objects.



Main Results

Method	Pick		Other skills	
	Seen objects	Unseen objects	Seen objects	Unseen objects
RT-1 (our data) [2]	54	25	50	50
RT-1 (original data)	31[1]	38	17[1]	13
VIMA-like [13]	62	50	50	25
MOO (ours)	**92**	**75**	**83**	**75**

Bonus: Better Environment Generalization

Method	Open-World Objects	Challenging Textures	New Environments
RT-1 (our data) [2]	17	7	29
VIMA-like [13]	50	7	7
MOO (ours)	**67**	**50**	**43**

TABLE II: Robustness evaluations for novel use cases. MOO is able to handle new objects, textures, and environments with substantially greater success than prior methods.

Limitations

- Still struggles with objects far outside training distribution.
- Failures often do plausible things and exhibit retry behavior
- Learning new action primitives is still very hard.

Takeaways

Lesson #2:
Leverage Foundation Models

Lesson #3:
Offline Robot Learning

- **Use VLMs** for generalization
- Import **internet-scale priors** into **offline datasets**
- When datasets are large enough, **some label noise is okay**

Agenda

01 The Ingredients for a Robotics Foundation Model

02 RT-1

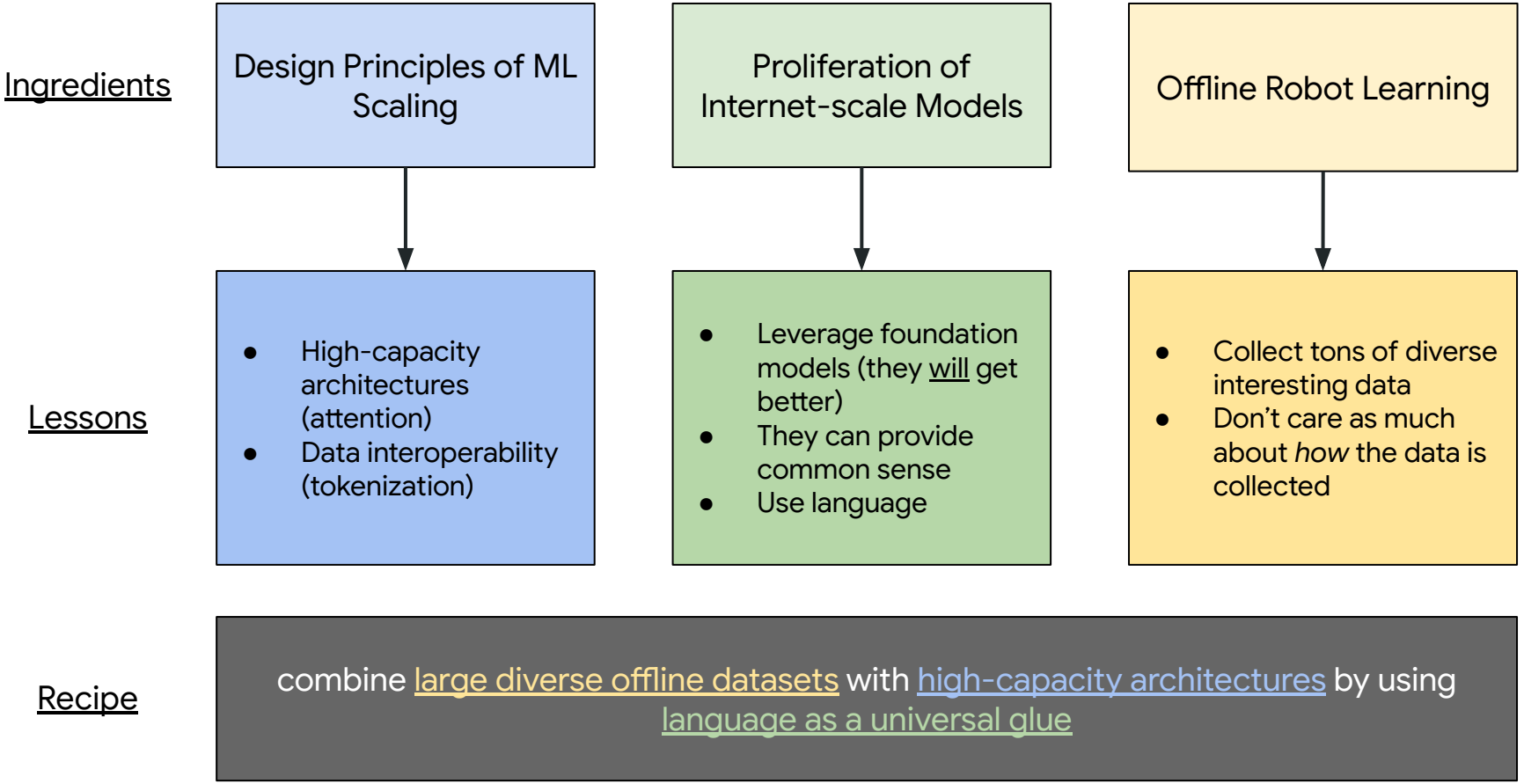
03 SayCan

04 Inner Monologue

05 DIAL

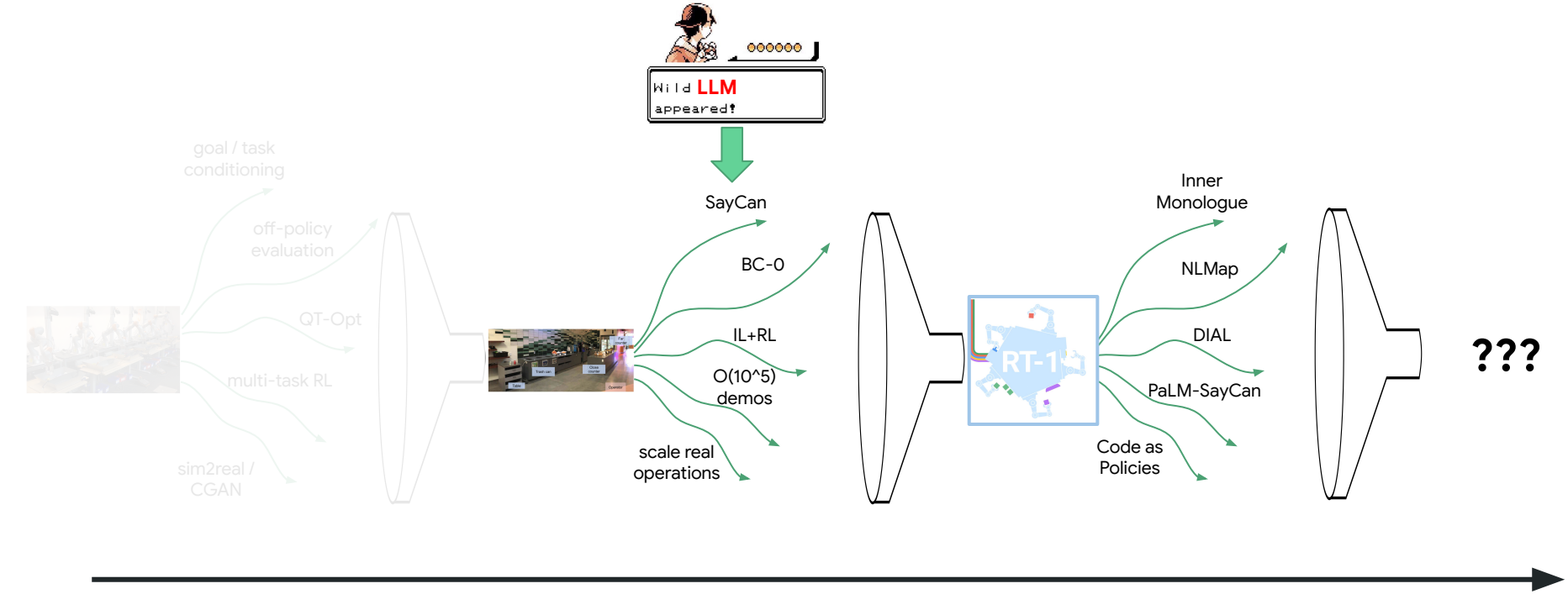
06 **What's Next?**

Turning Ingredients into a Recipe



combine large diverse offline datasets with high-capacity architectures by using language as a universal glue

Component	Method
Skill Learning	RT-1
Planning	SayCan, Inner Monologue
Low-level Control	Code as Policies
Data Augmentation	DIAL, ROSIE
Object-centric Representations	NLMap, MOO



2016 - 2021

"How do we do end-to-end robot learning in the real world?"

2021 - 2022

"How do we scale to many tasks with demonstrations?"

2022 - 2023

"How can we leverage foundation models to accelerate robotics?"

Open Research Directions

- Bottleneck is still on skill learning
- Leveraging Bitter Lesson 2.0 is non-trivial
- How do we collect diverse and useful data more scalably?
- What algorithms are absorbent “data sponges”?
- Transfer from human embodiment

Thank you!

say-can.github.io
innermonologue.github.io
robotics-transformer.github.io
robot-moo.github.io
nlmap-saycan.github.io