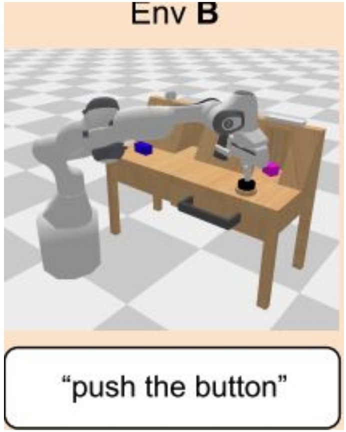
CALVIN Benchmark

Arya Miryala

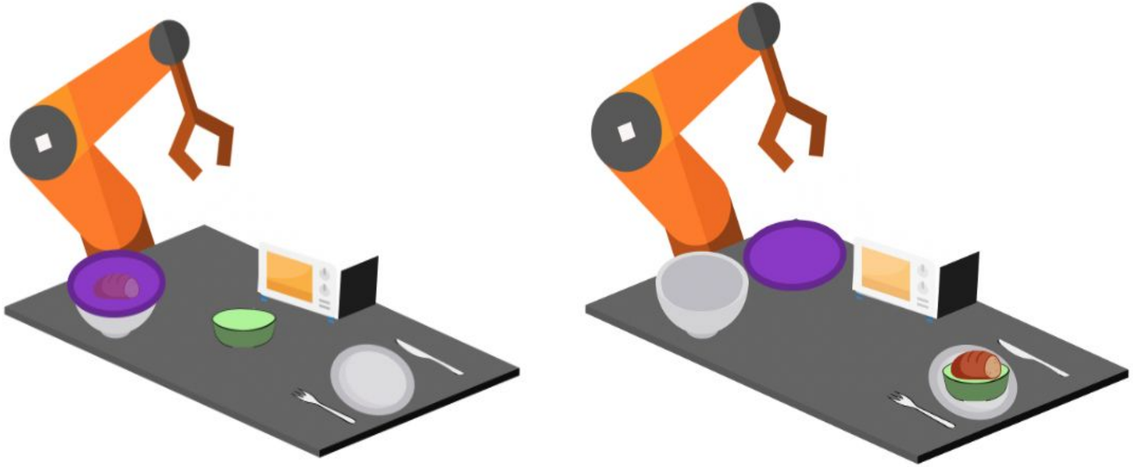
Why do we need a new benchmark?



Short-Horizon Task

- Progress in robot learning depends on **benchmarks** → lets researchers compare methods fairly.
- Most existing benchmarks focus on **short, simple tasks** (e.g., pushing or grasping).
- Real-world robotics needs **long-horizon tasks** (multi-step sequences).
- Language adds **flexibility and generalization**, but also makes tasks harder.
- CALVIN fills this gap: a **benchmark for language-conditioned, long-horizon manipulation**.

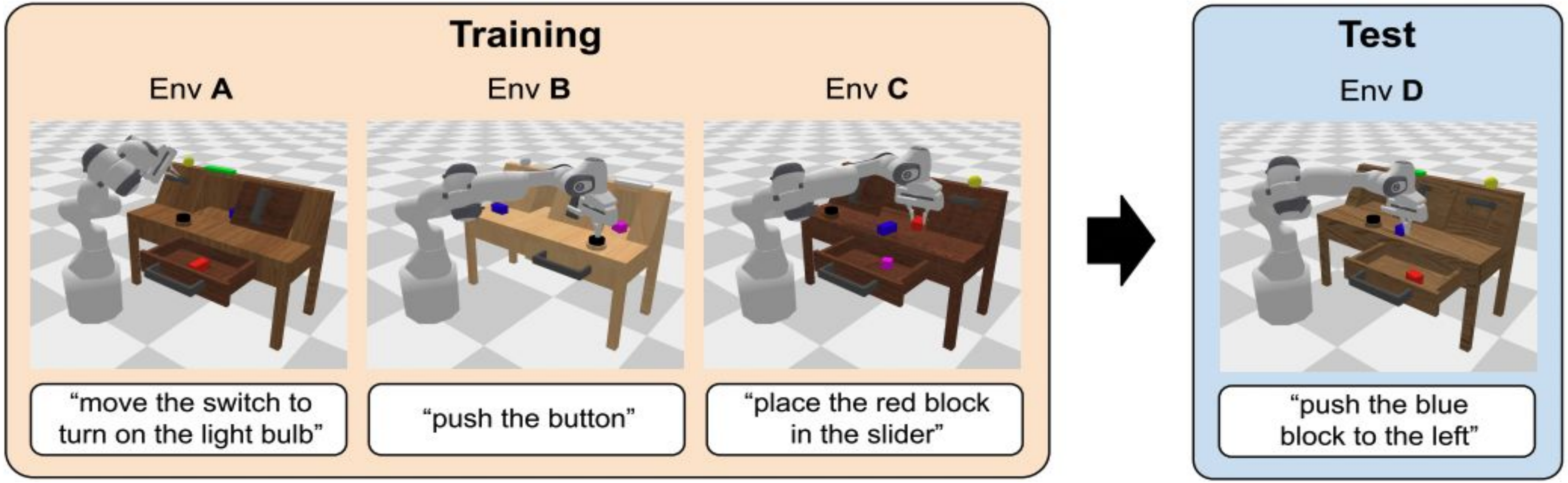
Long-Horizon Task



In the first task, the robot needs to retrieve the bread from the covered container and serve it on a plate.

What is CALVIN?

- Language-conditioned benchmark for **long-horizon manipulation**.
- Train on multiple environments, evaluate on **unseen tasks/environments**.
- Tests **generalization**: can policies handle new instructions/contexts?
- Supports imitation learning (IL) & reinforcement learning (RL).



Train on multiple environments, test on unseen tasks and settings

How are CALVIN Tasks Structured ?

- Each task is given as a **natural-language instruction**,
    - e.g. “open the drawer” or “place the red block in the slider.”
- The robot receives **multi-modal observations**:
    - RGB-D camera images
    - Language embedding of the instruction
    - Proprioceptive data (joint angles, gripper state, etc.)
- The policy must output **low-level control actions** (joint or end-effector commands).
- Tasks can be **composed sequentially** to form long-horizon goals
    - benchmark tests whether a single model can complete *multiple* instructions in a row.
- Training uses **demonstration data**; evaluation measures **success rate** on seen vs. unseen instructions and environments.

Observation Space	
RGB static camera	$200 \times 200 \times 3$
Depth static camera	200×200
RGB gripper camera	$84 \times 84 \times 3$
Depth gripper camera	84×84
Tactile image	$120 \times 160 \times 2$
Proprioceptive state	EE position (3) EE orientation (3) Gripper width (1) Joint positions (7) Gripper action (1)
Action Space	
Absolute cartesian pose (w.r.t. world frame)	EE position (3) EE orientation (3) Gripper action (1)
Relative cartesian displacement (w.r.t. gripper frame)	EE position (3) EE orientation (3) Gripper action (1)

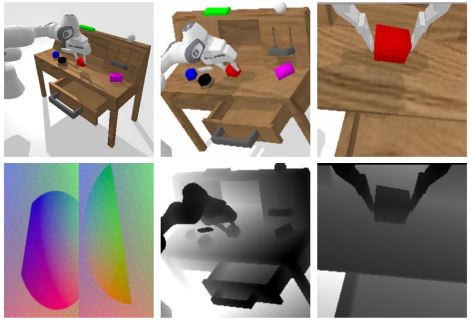


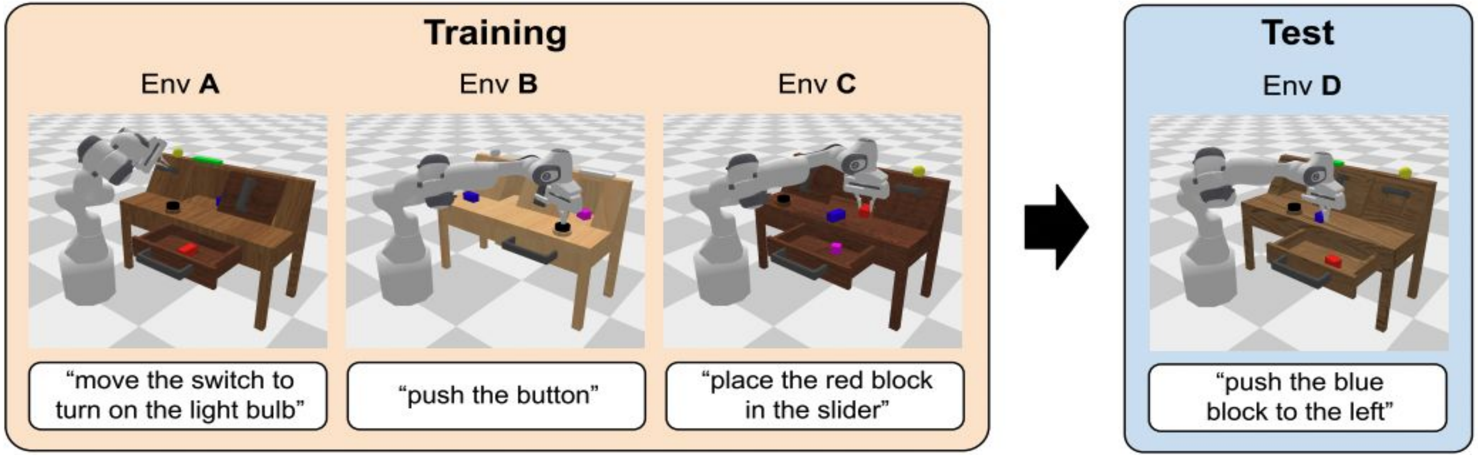
Fig. 3: CALVIN supports a range of sensors commonly utilized for visuomotor control: RGB-D images from both a static and a gripper camera, proprioceptive information, and vision-based tactile sensing (bottom-left).

Observation and action spaces in CALVIN: the robot receives visual, tactile, and proprioceptive inputs and outputs low-level control actions.

The CALVIN Simulation Environments

- CALVIN uses **four tabletop environments (A–D)** with variations in:
    a. *Object placements*
    b. *Lighting conditions*
    c. *Textures and background setup*
- Each environment includes **interactive elements** such as drawers, sliders, switches, and colored blocks
- These differences test **scene-level generalization** — the ability to perform the same instruction in new physical layouts
- The environments are built in simulation to allow for **consistent resets and reproducible evaluation**
- **Env D** is held out entirely for testing to evaluate transfer to unseen environments

Four tabletop environments with varying layouts and objects; Env D is held out for testing generalization



The CALVIN Dataset

- Contains **language-conditioned demonstrations** pairing visual observations, robot actions, and natural-language instructions.
- Covers **hundreds of thousands of trajectories** across 20 + manipulation tasks.
- Each trajectory can be short (e.g., “push the button”) or **long-horizon** (e.g., “open the drawer, then place the red block inside”).
- The dataset is divided into:
    a. **Seen tasks/environments** → used for training
    b. **Unseen tasks/environments** → used for testing generalization
- Enables study of **compositional generalization—**models must recombine learned sub-skills to solve new multi-step tasks

Task	Natural language instructions
rotate red block right	“rotate the red block 90 degrees to the right” “turn the red block right”
push blue block left	“go slide the blue block to the left” “push left the blue block”
move slider left	“grasp the door handle, then slide the door to the left” “slide the door to the left”
open drawer	“grasp the handle of the drawer and open it” “go open the drawer”
lift red block	“lift the red block from the table” “pick up the red block”
pick pink block from drawer	“pick up the pink block lying in the drawer”
place in slider	“put the grasped object in the slider”
stack blocks	“stack blocks on top of each other”
unstack blocks	“collapse the stacked blocks” “go to the tower of blocks and take off the top one”
turn on light bulb	“toggle the light switch to turn on the light bulb”
turn off green light	“push the button to turn off the green light”

Fig. 4: Example crowd-sourced natural language instructions to specify manipulation tasks in CALVIN.

Learning Approaches in CALVIN

Task	Condition
Rotate red/blue/pink block right	The object has to be rotated clockwise more than 60° around the z-axis while not being rotated more than 30° around the x or y-axis.
Rotate red/blue/pink block left	The object has to be rotated counterclockwise more than 60° around z while not being rotated more than 30° around the x or y-axis.
Push red/blue/pink block right	The object has to move more than 10 cm to the right while having surface contact in both frames.
Push red/blue/pink block left	The object has to move more than 10 cm to the left while having surface contact in both frames.
Move slider left/right	The sliding door has to be pushed at least 12 cm to the left/right.
Open/close drawer	The drawer has to be pushed in/pulled out at least 10 cm.
Lift red/blue/pink block table	The object has to be grasped from the table surface and lifted at least 5 cm high. In the first frame the gripper may not touch the object.
Lift red/blue/pink block slider	The object has to be grasped from the sliding cabinet's surface and lifted at least 3 cm. In the first frame the gripper may not touch the object.
Lift red/blue/pink block drawer	The object has to be grasped from the drawer's surface and lifted at least 5 cm high. In the first frame the gripper may not touch the object.
Place in slider/drawer	The object has to be placed in the sliding cabinet/drawer. It must be lifted by the gripper in the first frame.
Push into drawer	The object has to be pushed into the drawer. It has to touch the table surface in the first frame.
Stack blocks	A block has to be placed on top of another block. It may not be in contact with the gripper in the final frame.
Unstack blocks	A block has to be removed from the top of another block. It may not be in contact with the gripper in the first frame.
Turn on/off light bulb	The switch has to be pushed up/down to turn on/off the yellow light bulb.
Turn on/off LED	The button has to be pressed to turn on/turn off the green LED light.

Fig. 6: List of all 34 tasks with their respective success criteria.

- CALVIN benchmarks a variety of **policy learning methods** for long-horizon, language-conditioned control.
- **Multi-Context Imitation Learning (MCIL):**
  a. Learns from demonstration data across multiple environments.
  b. Uses language embeddings to generalize to unseen instructions.
  c. Most effective and sample-efficient method in the benchmark.
- **Reinforcement Learning (RL) baselines (e.g., PPO):**
  a. Learn policies from trial-and-error reward feedback.
  b. Struggle with sparse rewards and long-horizon dependencies.
- **Language-conditioned baselines:**
  a. Combine pretrained vision-language encoders (like CLIP) with control networks.
  b. Test how well language understanding transfers to manipulation.
- focus is on **evaluating generalization**—how well each method handles unseen instructions and environments (Env D).

Understanding CALVIN's Task Complexity

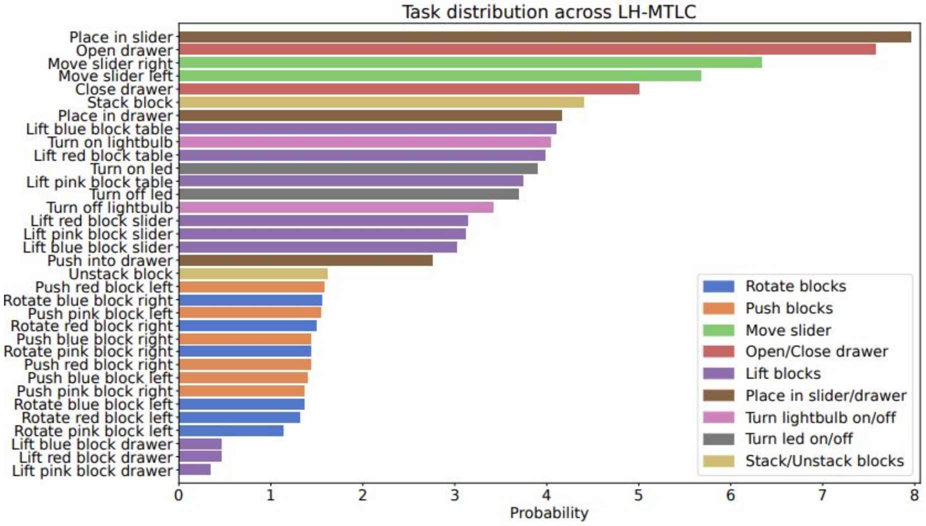


Fig. 7: Visualization of the subtask distribution across the 1000 instruction chains used for the Long Horizon MTLC evaluation. We show the percentage in which each subtask appears in the distribution.

- CALVIN includes **a mix of short and long-horizon tasks**, covering diverse manipulation primitives (e.g., open drawer, push block, toggle switch)
- Tasks are **chained together** to form multi-step, language-conditioned instructions
- The **task distribution** is balanced enough to prevent bias toward any single action, but still complex enough to challenge compositional generalization
- **Performance drops** on long-horizon tasks highlight how chaining multiple skills remains an open research problem
- These insights explain **why imitation learning outperforms RL**—it leverages structured demonstrations for multi-step reasoning

Benchmark Results on CALVIN

- **Multi-Context Imitation Learning (MCIL)** achieved the **highest success rates**, showing strong generalization across unseen tasks.
- **Reinforcement Learning (RL)** baselines (e.g., PPO) performed significantly worse — they often failed to complete long-horizon tasks due to sparse rewards.
- **Language-conditioned models** using pre trained encoders (e.g., CLIP-based) showed moderate performance but lacked full compositional understanding
- Models trained on **multi-environment data** generalized better than those trained in a single context
- Performance drops notably from **seen tasks → unseen tasks**, highlighting the difficulty of long-horizon generalizationS

Input					Train → Test	MTLC	LH-MTLC				
Static Camera		Gripper Camera		Tactile		(34 tasks)	No. Instructions in a Row (1000 chains)				
RGB	D	RGB	D	RGB			1	2	3	4	5
✔	✗	✗	✗	✗	D → D	53.9%	48.9%	12.9%	2.6%	0.5%	0.08%
✔	✗	✗	✗	✗	A,B,C,D → D	35.6%	28.2%	2.5%	0.3%	0%	0%
✔	✗	✗	✗	✗	A,B,C → D	38.6%	20.2%	0.2%	0%	0%	0%
✔	✗	✔	✗	✗	D → D	51.8%	34.4%	5.8%	1.1%	0.2%	0.08%
✔	✗	✔	✗	✗	A,B,C,D → D	49.7%	37.3%	2.7%	0.17%	0%	0%
✔	✗	✔	✗	✗	A,B,C → D	38.0%	30.4%	1.3%	0.17%	0%	0%
✔	✗	✗	✗	✔	D → D	54.2%	28.5%	3.2%	0%	0%	0%
✔	✗	✗	✗	✔	A,B,C,D → D	47.9%	22.7%	2.3%	0.3%	0%	0%
✔	✗	✗	✗	✔	A,B,C → D	43.7%	17.3%	0.8%	0.08%	0%	0%
✔	✔	✔	✔	✗	D → D	46.1%	28.2%	4.6%	0.3%	0.08%	0%
✔	✔	✔	✔	✗	A,B,C,D → D	40.7%	14.4%	1.8%	0.08%	0.08%	0%
✔	✔	✔	✔	✗	A,B,C → D	30.8%	21.1%	1.3%	0%	0%	0%

Fig. 8: Baseline performance of MCIL [6] on the CALVIN Challenge for different combinations of training and test environments and sensor suites.

My Work / Key Takeaways

- Set up and ran complex robotics codebases (CALVIN / ManiSkill) involving Hydra configs, simulation rendering, and policy evaluation
    a. Logs in github repo: https://github.com/aryamiryala/CALVIN_Replication
- Tried running MCIL learning but ran into too many errors and ran out of time to debug

Benchmark Contribution:
CALVIN introduces a **unified benchmark** for *language-conditioned, long-horizon robot manipulation*, filling a gap between vision-language models and embodied control.

Task Design:
Focuses on **sequential, compositional tasks** (e.g., “open drawer, then move block”) that require **multi-step reasoning**—not just single-action imitation.

Dataset & Environment:
Provides **large-scale demonstrations** in **simulated tabletop scenes** with multiple objects and natural-language instructions, enabling reproducible research.

Model Insights:
Shows that **multi-context imitation learning (MCIL)** and **language-conditioned policies** can generalize across unseen tasks, scenes, and instructions—but still struggle with *long-term temporal credit assignment*.

Evaluation Framework:
Establishes **clear metrics** for generalization and compositionality, setting a baseline for future multimodal policy-learning approaches