Video: Multimodal Understanding
&
Guiding Robotic Manipulation

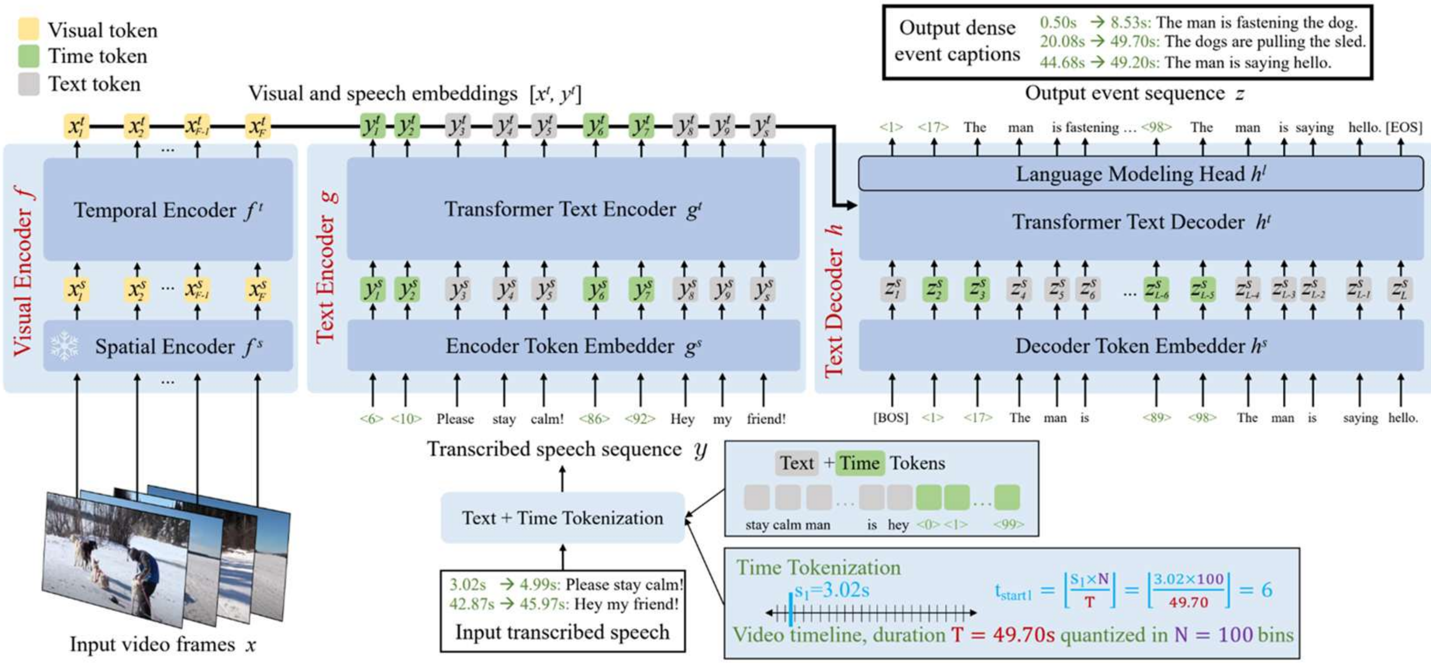
Cordelia Schmid

Video understanding

- Very rapid progress recently with VideoLLMs
- Video Captioning works surprisingly well
- Many benchmarks start to saturate, too easy?
- Definition of video understanding?

Dense Video Captioning: Vid2Seq

- Single target sequence consists of **Text + Time tokens combining localization + captioning**
- Large-scale pretraining from narrated untrimmed videos



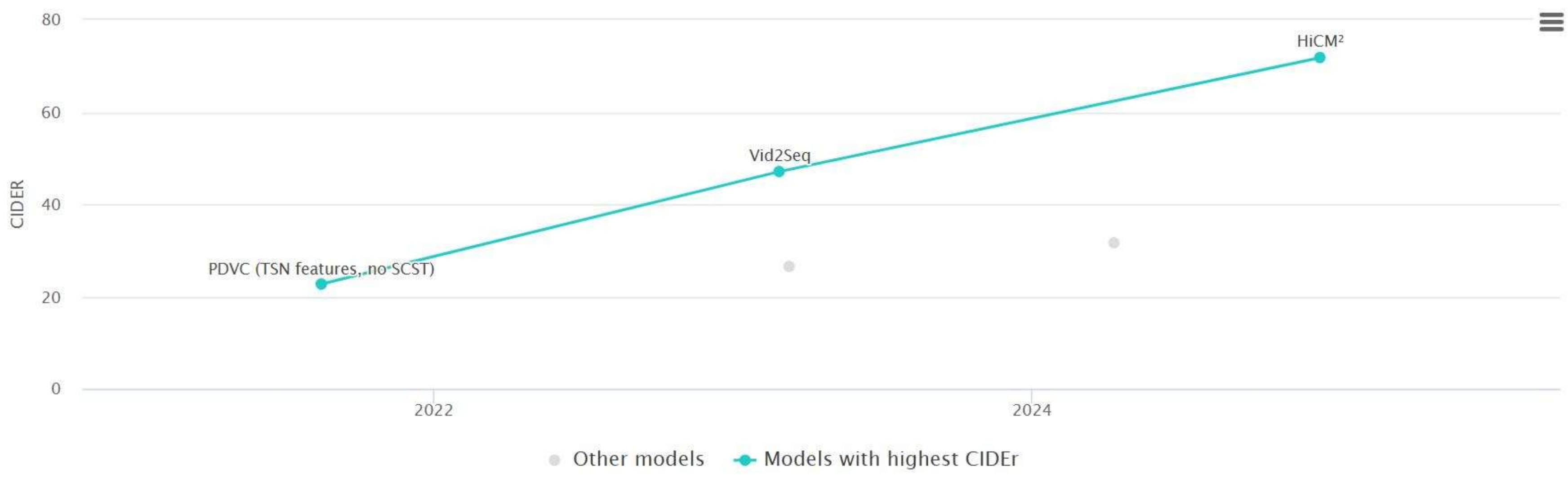
[Vid2Seq, A. Yang, A. Nagrani, P. Seo, A. Miech, J. Pont-Tuset, I. Laptev, J. Sivic, C. Schmid, CVPR 2023]

State of the art on dense video captioning

Dense Video Captioning on YouCook2

Leaderboard   Dataset

View CIDEr by Date for All models



Filter: untagged

Edit Leaderboard

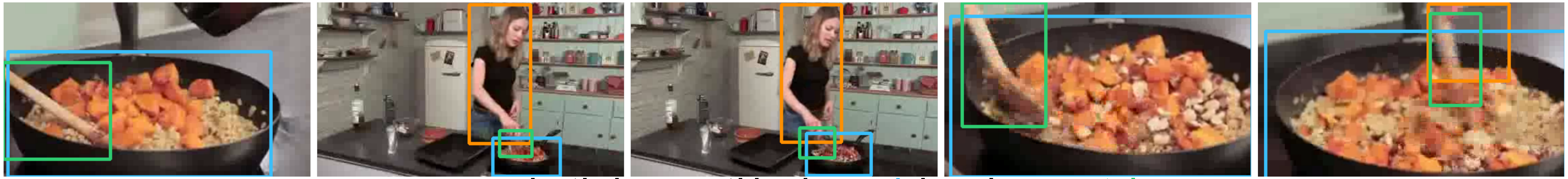
Open problems

- More difficult benchmarks and reasoning based approaches
  → Minerva dataset for VQA with reasoning traces (ICCV'25)
  → Spatiotemporal Hints for Video Understanding (CVPR'26)

- Video grounding for detailed, verifiable and scalable description
  → Grounded video caption generation (GROVE)
  → Temporal chain of thought (TCOT)

Grounded video caption generation

- Our approach
  - Large-scale pretraining
  - Spatio-temporal grounded caption generation
  - iGround dataset with training and test set

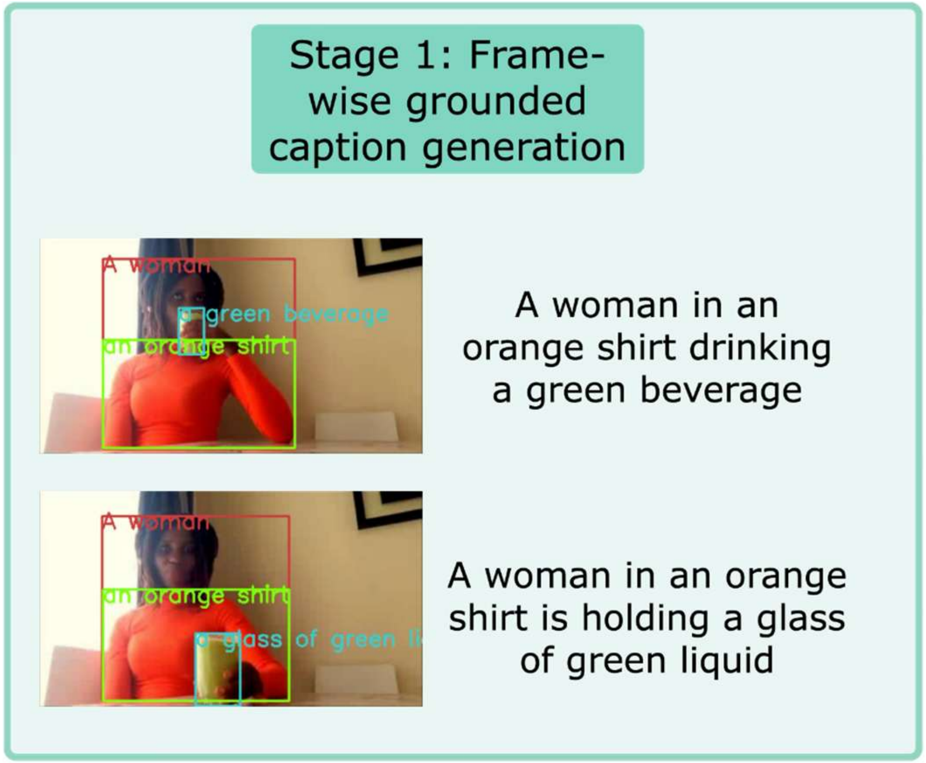


a woman is stirring something in a wok by using a spatula

[Large-scale pretraining for grounded video caption generation, E. Kazakos, C. Schmid, J. Sivic, ICCV, 2025]

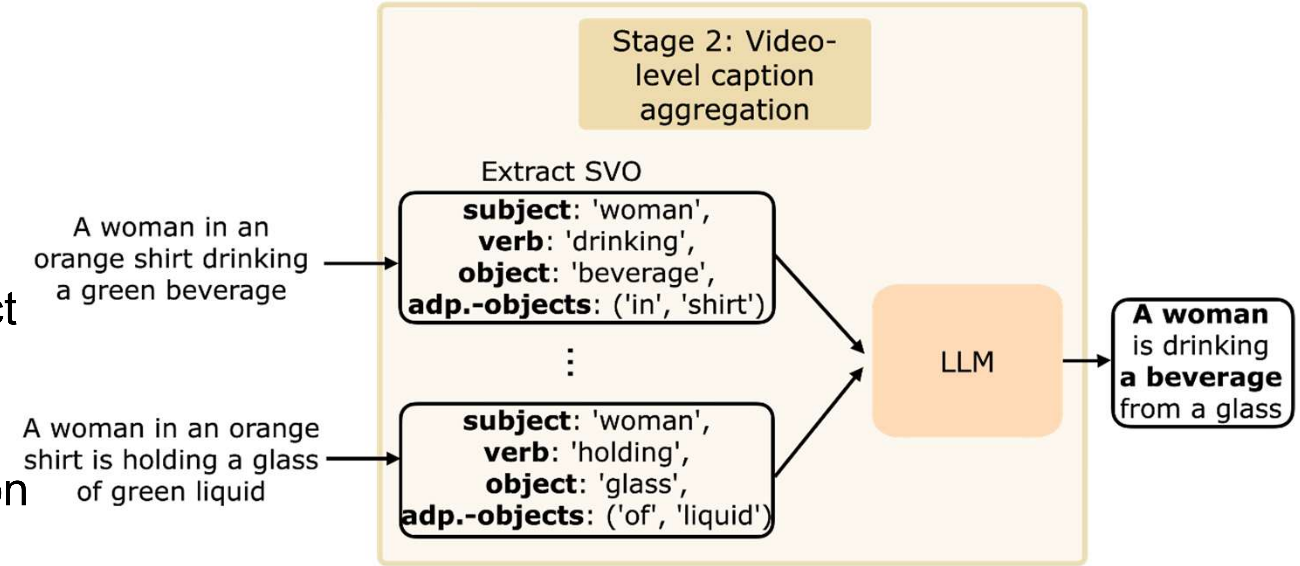
Large-scale generation of grounded captions

- Stage 1: Frame-wise grounded caption generation
  - GLaMM
  - Alternative: Open vocabulary detection (OWL-VIT)



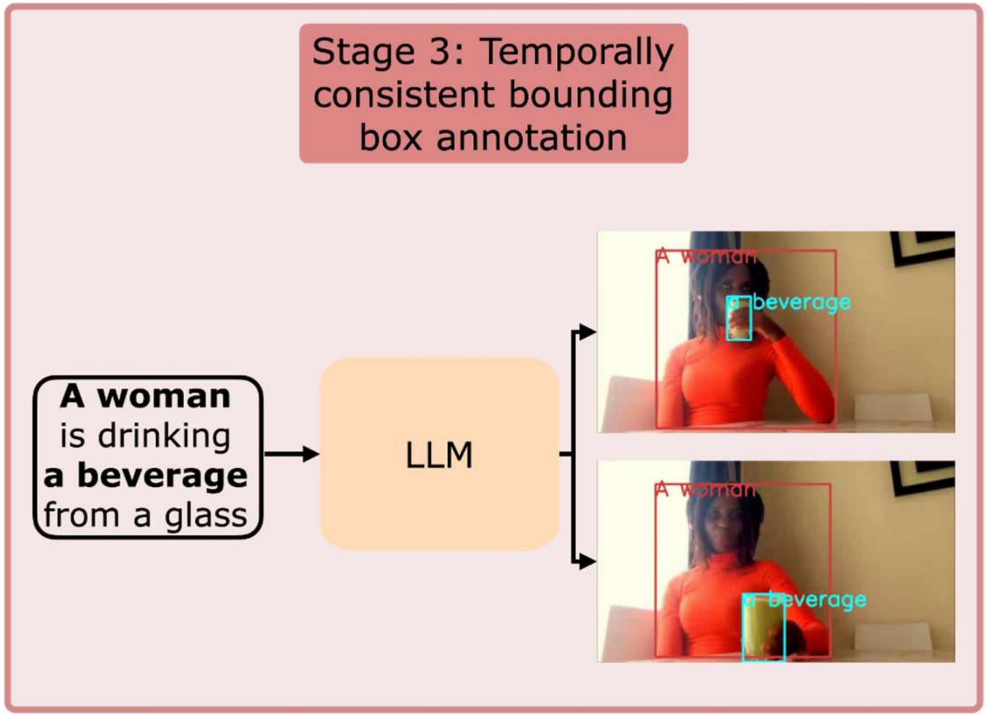
Large-scale generation of grounded captions

- Stage 1: Frame-wise grounded caption generation
- Stage 2: Video-level caption generation
  - Extract Subject-Verb-Object (SVO) triplets
  - LLM (Llama-2) to combine them into video level caption with tags
  - Alternative: applying LLM directly on captions



Large-scale generation of grounded captions

- Stage 1: Frame-wise grounded caption generation
- Stage 2: Video-level caption generation
- Stage 3: Temporal consistent bounding box annotation
  - Text classification with LLM
  - Alternative: tracking



HowToGround1M dataset

- Randomly sample 1M video clips from HowTo100M videos using start/end timestamps from HowToCaption [1]
- Downsample videos to 5fps
- Automatic annotation for 1M video clips with 80.1M bounding boxes in 43.6 frames

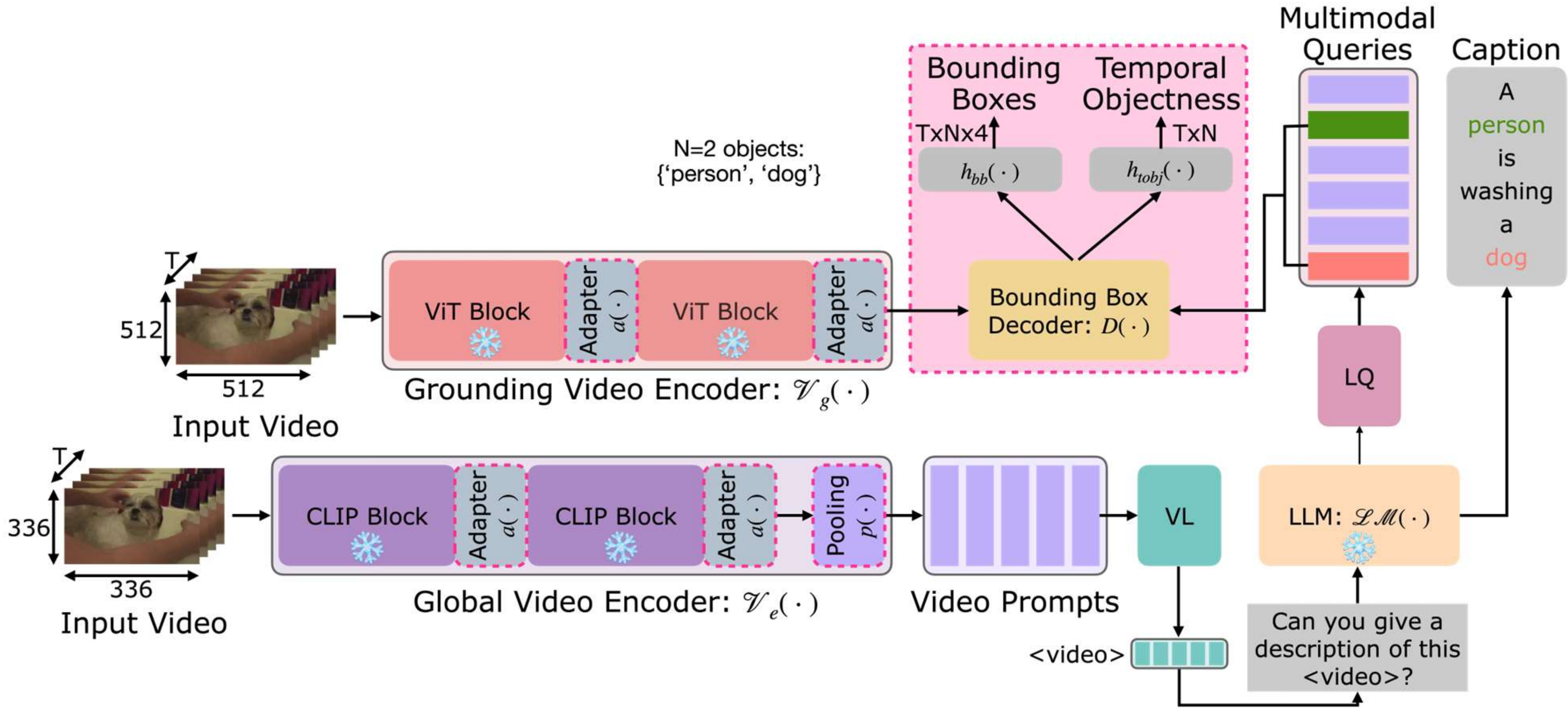
Dataset	Annot. type	Multiple frames	Multi-object grounding	Num. videos	Num. instances
VidSTG [43]	Manual	✔	✗	36.2K	9.9M
HC-STVG [34]	Manual	✔	✗	10.1K	1.5M
ActivityNet-Entities [45]	Manual	✗	✔	37.4K	93.6K
HowToGround1M (Ours)	Automatic	✔	✔	**1M**	**80.1M**
iGround (Ours)	Manual	✔	✔	2K	236.9K

Comparison of HowToGround1M and iGround with SOTA video grounding datasets

iGround dataset

- Manually annotated, no overlap with HowToGround1M dataset
- Annotation by multiple rater with precise annotation guidelines
- Train/val/test: 2000/500/1000
- Temporally dense annotations, multiple objects per frame

GROunded Video caption gEneration (GROVE)



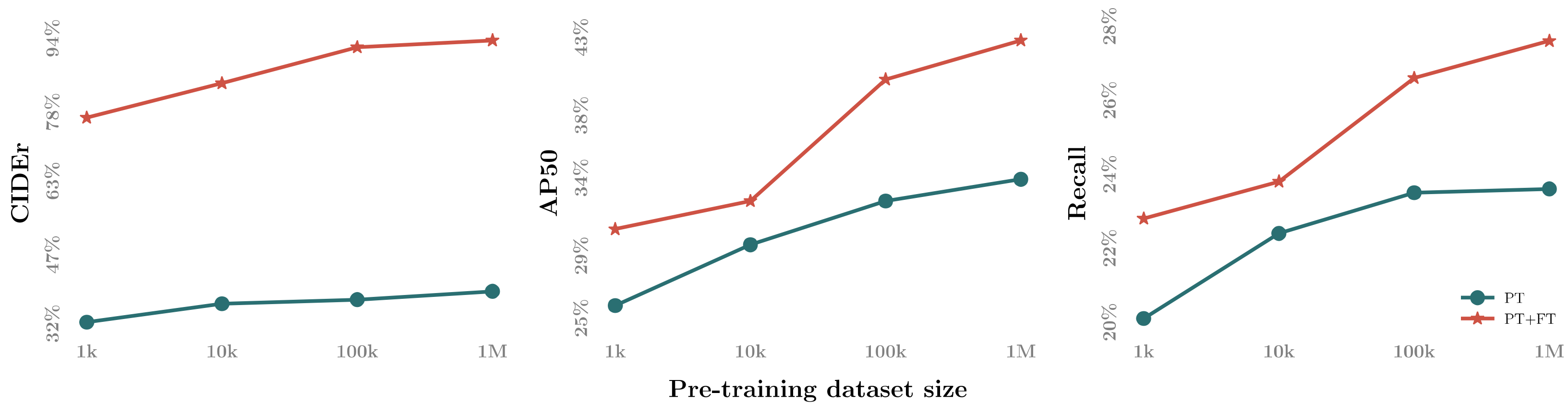
- 1 encoder for captioning + 1 for grounding
- Adapters for spatio-temporal modelling
- LLM predicts caption and noun phrases locations
- Temporal objectness predicts if object is present in a frame

Experimental results

	METEOR	CIDER	AP50	Recall
Automatic annotation	13.8	40.0	27.1	20.4
GROVE - PT (Ours)	14.3	50.6	33.6	24.3
GROVE - FT (Ours)	21.0	77.7	15.8	18.1
GROVE - PT+FT (Ours)	**21.4**	**83.5**	**40.0**	**28.7**

- Automatic annotation improves the results
- Pretraining improves performance

Experimental results



- Performance increases with increase in pre-training dataset size
- Behaviour is consistent during both pre-training (PT) as well as fine-tuning after pre-training (PT+FT)

Comparison to state of the art

Referring expressions localization

Method	m_{sIoU}
STVGBert [36]	47.3
TubeDETR [43]	59.0
STCAT [14]	61.7
DenseVOC [51]	61.9
GROVE FT (Ours)	61.3
GROVE PT+FT (Ours)	**63.7**

Results on VIDSTG test set, declarative sentences

Method	FT	m_{sIoU}
PG-V-L (13B) [25]	✗	35.1
GLaMM [31]		
+ SAM2 [32]	✗	38.6
GROVE	✗	**43.0**
VideoGLaMM[26]	✓	39.7
GROVE	✓	**55.5**

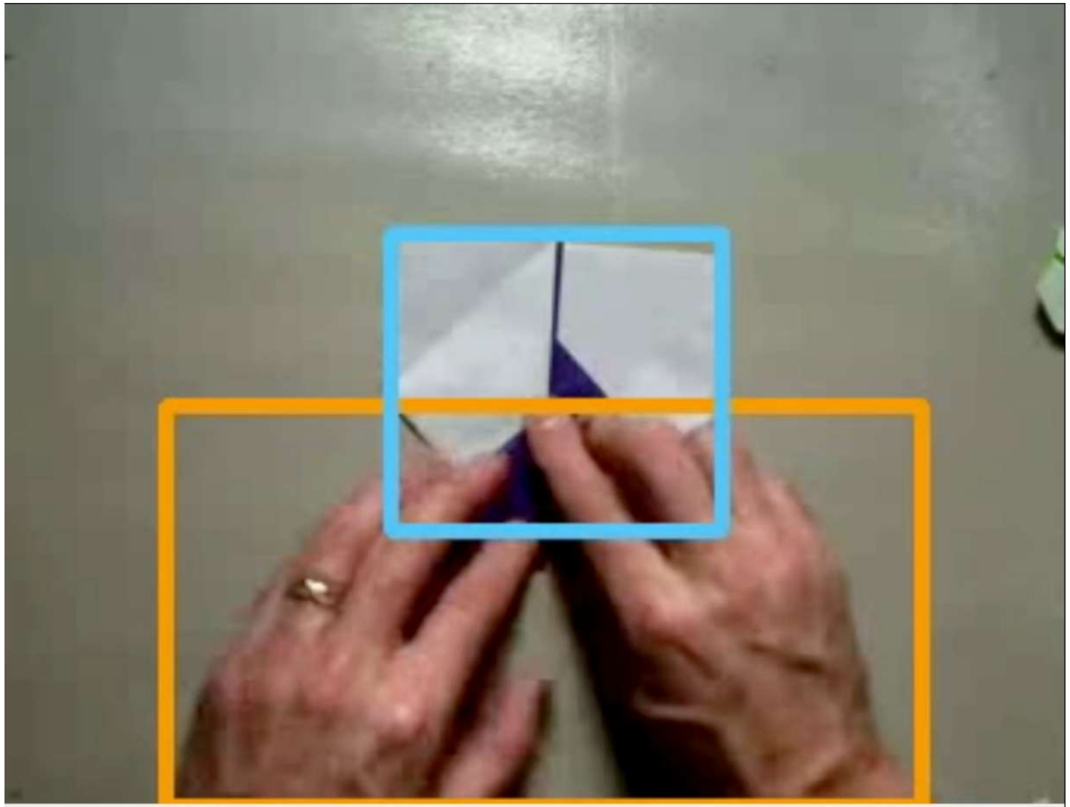
Results on VIDSTG test set, interrogative sentences

Spatio-temporal grounding

Method	$F1_{all}$	$F1_{all\_per\_sent}$	$F1_{loc}$	$F1_{loc\_per\_sent}$
GVD	07.10	17.30	23.80	59.20
GROVE FT (Ours)	09.51	21.15	30.96	68.79
GROVE PT+FT (Ours)	**13.57**	**24.23**	**43.07**	**76.99**

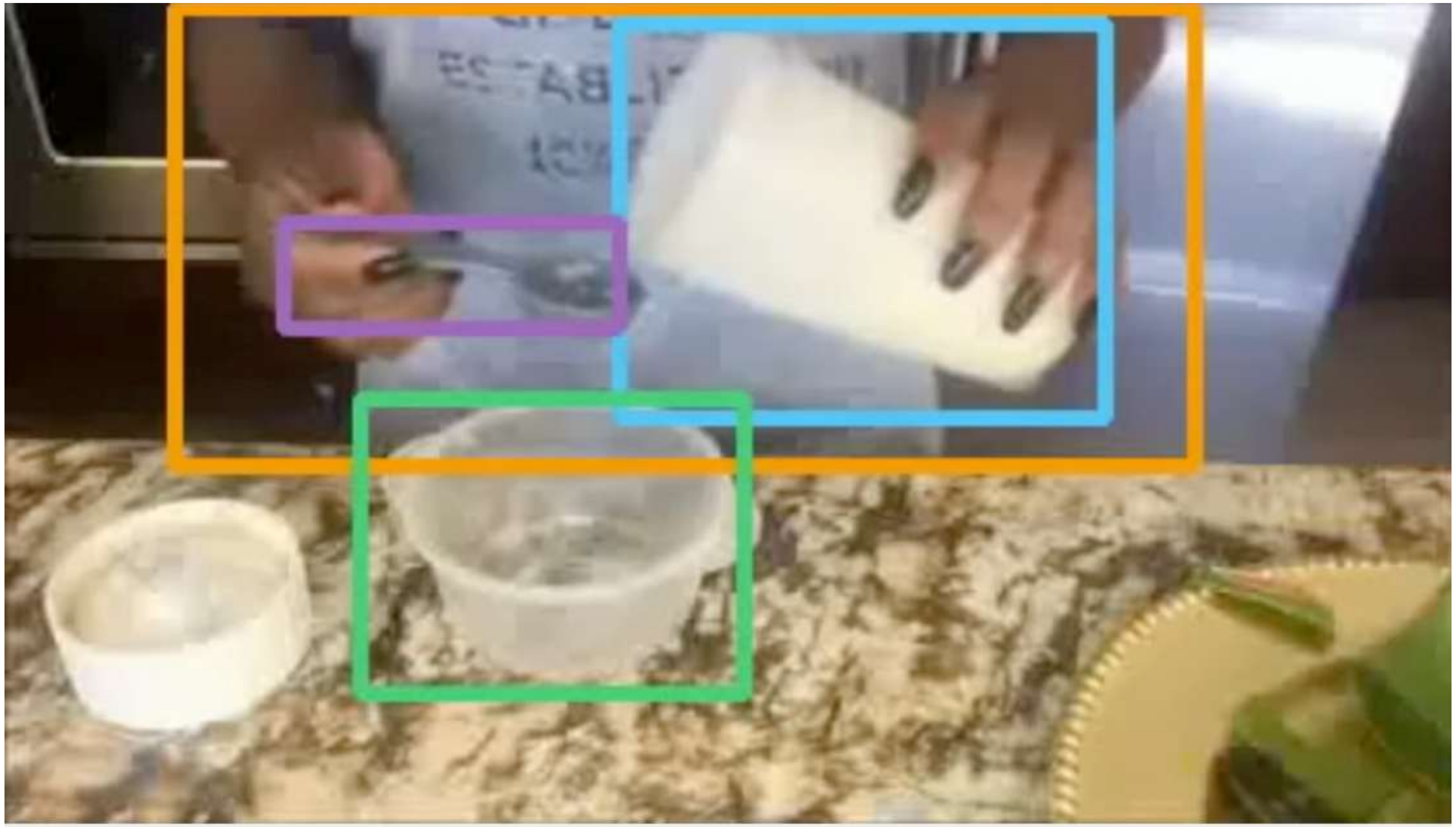
Results on the validation set of ActivityNet-Entities

Qualitative results



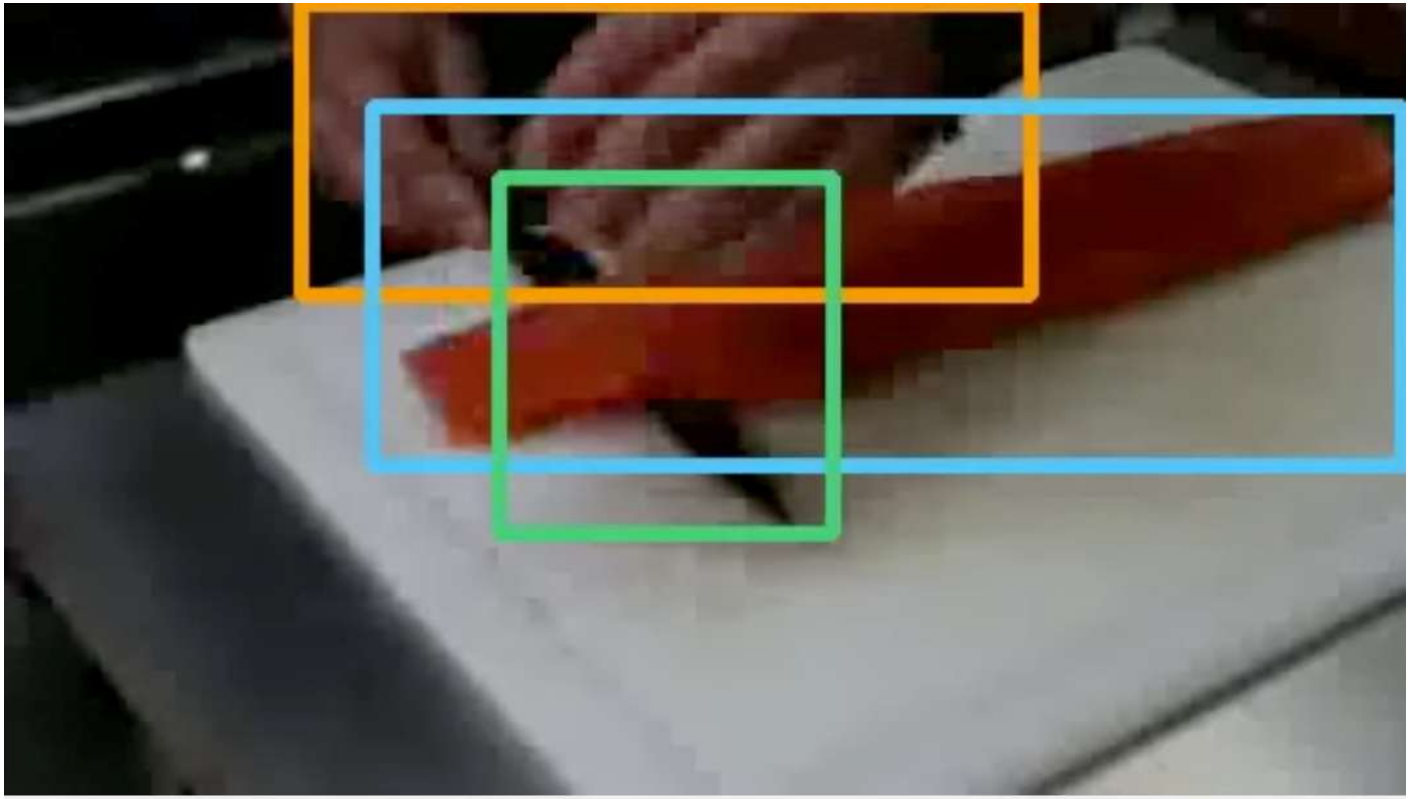
a person's hands are folding a paper craft

Qualitative results



a person is adding something from a container to a bowl using a spoon

Qualitative results



a person is cutting the meat into pieces by using a knife

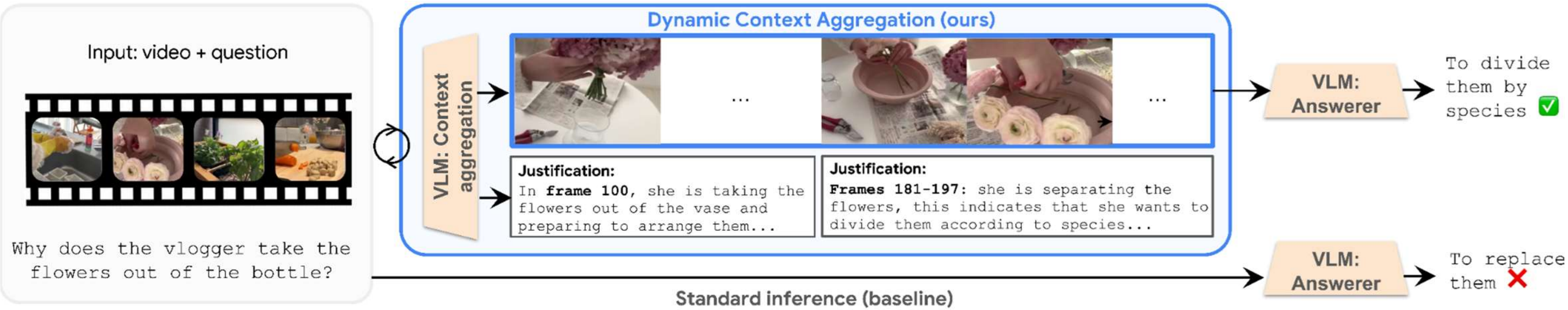
Overview

- Grounded video caption generation (GROVE)
- ***Temporal chain of thought (TCOT)***

Temporal Chain of Thought

Long Video Understanding by Thinking in Frames

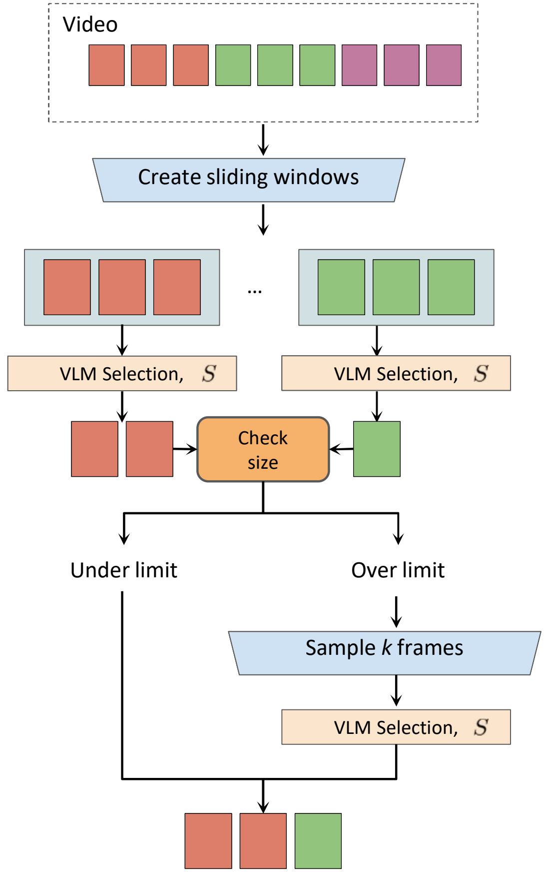
- Decompose a Video QA problem into
  - Finding relevant frames in the video
  - Answer the question
- Use the same VLM in both stages



[Temporal Chain of Thought: Long Video Understanding by Thinking in Frames. Arnab, Iscen, Caron, Fathi, Schmid. Neurips 2025.]

Temporal Chain of Thought (TCoT)

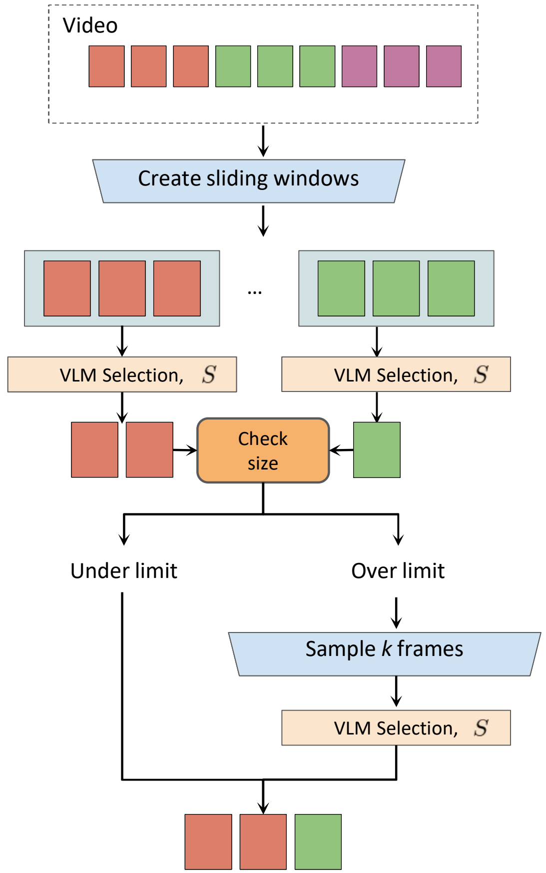
- Partition video into *l* smaller windows.
- Process each window independently.
- If total number of selected frames is above the context limit, rerun within selected frames.




```
You will be given a question about a video and five
possible answer options.

Frames: {frame1}, . . . , {frame N}
Question: {question}
Possible answer choices: {answer choices}

Return the frame ids which can answer the
given question.

Please use the following JSON format for your
output:
{
''frame_ids'': [List of integer frame IDs],
''justification'': Justification about your output
}
```

Temporal Chain of Thought (TCoT)

- Video length is decoupled from model's context limit.
- We need to only fit a single window within the context of the VLM.

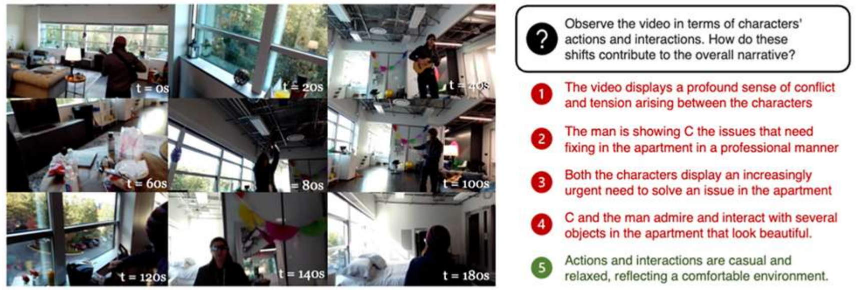


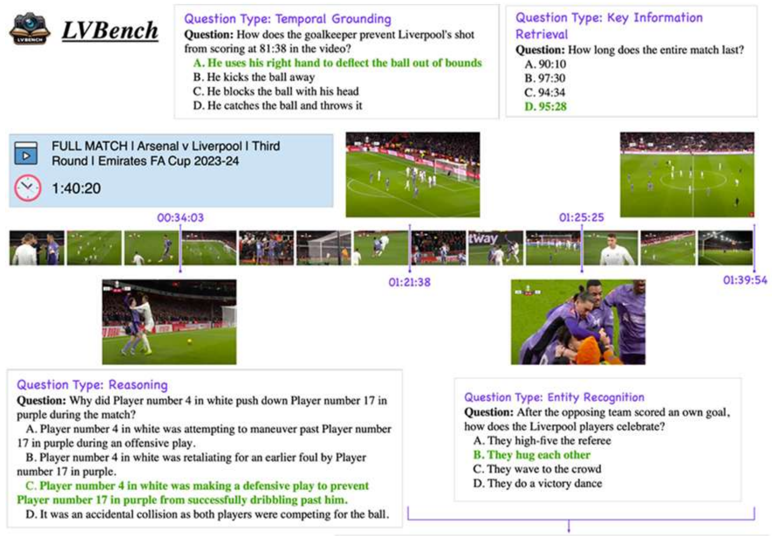
Long Video Example

- The TCoT approach can find relevant segments from long video, and then reason about it.
- Example from LVBench: [YouTube]

Experimental Setup

- Gemini 1.5 Flash as primary VLM
- Two primary datasets:
    - [Egoschema]
        - Videos are fairly short, all 3 minutes
        - Questions require looking at almost entire video
        - Egocentric videos from Ego4D.
    - [LVBench]
        - Videos are long, average 68 minutes.
        - Questions require looking at multiple segments of the video.
        - Videos from YouTube





Temporal Chain of Thought Analysis

- Baseline is standard inference with Gemini 1.5 Flash
- Improvements of various TCoT methods similar on Egoschema where videos are short and fit in ~32K context window

	Egoschema	LVBench
Baseline inference	72.6	50.3
Single-step	74.8	48.3
Hierarchical	74.0	53.3
Dynamic-segment	**75.2**	**61.7**

- Improvement of Dynamic-Segment CoT is substantial, because it can consider the whole video.

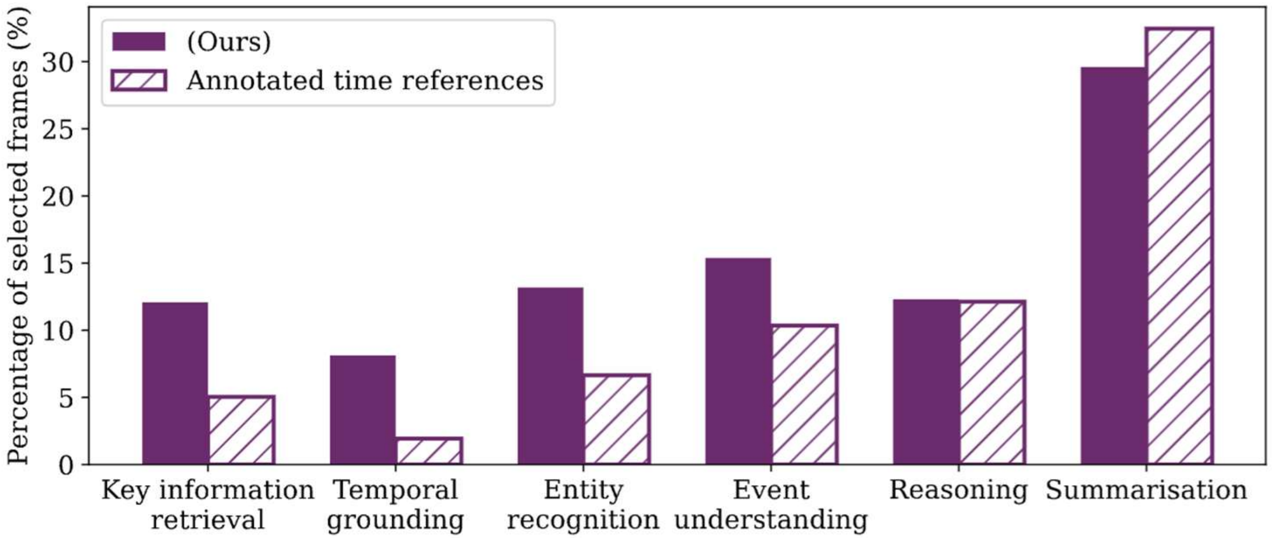
Generalises to other VLMs too

- TCoT generalises to Qwen-2.5 and GPT-4o-mini
- We compare with using both equal context limit (32K) or baseline inference with as many frames as it can support.

	Baseline inference		Our inference
	Equal context	Max. frames	(TCoT)
Qwen-2.5-VL-7B [6]	36.0	46.1	**49.1**
GPT-4o-mini [1]	43.4	48.0	**53.5**
Gemini 1.5 Flash [48]	50.3	58.9	**61.7**

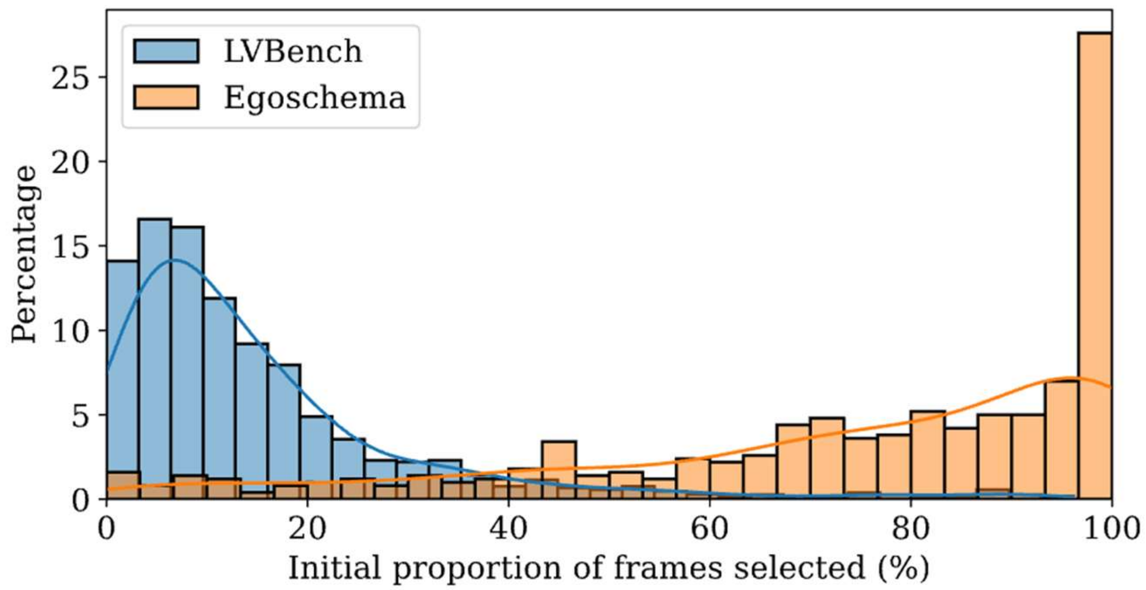
Adaptivity in Frame Selection

- We select a different proportion of frames depending on the question type.
- We can compare to labelled “time references” on LVBench.



Adaptivity in Frame Selection

- Select most of the frames in Egoschema, much fewer on LVBench
- Dataset constructed so that questions should require looking at least 55% of the video.



Other Frame Selection Strategies

- General idea of selecting relevant context is beneficial.
- Ranking based on feature similarity outperformed by using VLM itself to select relevant context.
- Selecting from pixels better than captions

	Selection strategy	Egoschema	LVBench
1	Uniform sampling	72.6	50.3
	Feature similarity		
2	Question $\rightarrow$ captions	73.8	52.1
3	Question $\rightarrow$ frames	73.4	54.4
	VLM-Based		
4	Select from "concise" captions	74.0	58.3
5	Select from "long" captions	72.8	60.4
6	Select directly from frames (Ours)	**75.2**	**61.7**
7	Oracle with annotated time references	–	67.4

State-of-the-Art Comparisons – Egoschema / Next-QA

Method	LLM / VLM	Egoschema		NeXT-QA
		Subset	Full set	Accuracy
LangRepo [24]	Mixtral	66.2	41.2	60.9
MoreVQA [37]	PaLM-2	–	51.7	69.2
Video Agent [55]	GPT-4	60.2	54.1	71.3
Video Agent [55]†	Gemini 1.5 Flash	65.6	–	–
LLoVi [66]	GPT-4	61.2	52.2	73.8
GPT-4V [3, 7]	GPT-4V	63.5	55.6	–
LVNet [39]	GPT-4o	68.2	61.1	72.9
MotionEpic [15]	Vicuna	–	–	76.0
VideoTree [56]	GPT-4	66.2	61.1	75.6
LongVU [42]	Qwen2-7B	–	67.6	–
Gemini 1.5 Flash	Gemini 1.5 Flash	72.9	67.8	80.0
TCoT (ours)	Gemini 1.5 Flash	**75.2**	**69.1**	**81.0**

State-of-the-Art Comparisons - LVBench

LVBench

Model	Context	Total	Accuracy
VideoAgent [15]†	–	–	37.6
InternVL-2.5-78B [11]	–	–	43.6
Qwen-2.5-VL-7B [6]	128K	128K	46.1
GPT-4o-mini [1]	22K	22K	48.0
Gemini 1.5 Flash [54]	32K	32K	50.3
Gemini 1.5 Flash [54]	700K	700K	58.9
TCoT (Qwen-2.5-VL-7B)	128K	320K	49.1
TCoT (GPT-4o-mini)	22K	86K	53.5
TCoT (Gemini 1.5 Flash)	32K	672K	**61.7**

State-of-the-Art Comparisons – Open-EQA

Open-EQA

Model	Context	Total	LLM Match
Claude 3 [4, 39]	6K	6K	36.3
GPT-4V [3, 39]	4.2K	4.2K	55.3
Gemini 1.5 Flash	77.4K	77.4K	68.0
TCoT (Gemini 1.5 Flash)	32K	76.4K	**69.2**

Qualitative examples

Question: Why is the refrigerator at the vlogger's hotel full? (A) Hotel offers a meal prep service (B) Hotel provides a complimentary minibar (C) The vlogger has lots of leftover food (D) The vlogger shares the refrigerator with other guests
Baseline answer: The vlogger has lots of leftover food ❌
Our answer: The vlogger shares the refrigerator with other guests ✅
Our frame selection $\mathcal{S}$: [1575, 1576, 1577, 1578, 1580, 1581, 1582, 1583]
Our VLM selection justifications:
Frames 1575-1578: The frames show a notice indicating that the refrigerator is for shared use among guests.
Frames 1580-1583: The frames show the refrigerator is full of various food items, suggesting that multiple guests are using it and storing their food.
Our answerer justification: The video shows a refrigerator in the vlogger's hostel. The refrigerator is full of various food items. There is a notice indicating that the refrigerator is for shared use among guests.

Conclusion

- Grounded video caption works well
- Excellent results for scalable video understanding
- Further improve multihop reasoning
- Metrics for video captioning need improvement

Video-guided policies for robotic manipulation

Robot executes human instruction

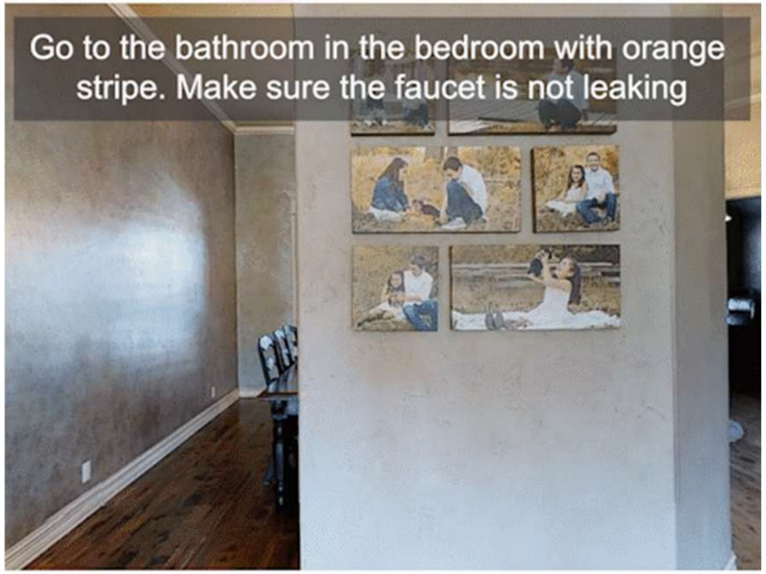
"go to the fridge and get a drink"

Requires

- Understanding language
- Modeling the visual environment
- Planning and executing the task

History Aware Multimodal Transformer (HAMT) for Vision-and-Language Navigation

- Train autonomous agent to follow natural language instructions to navigate in in-door environments

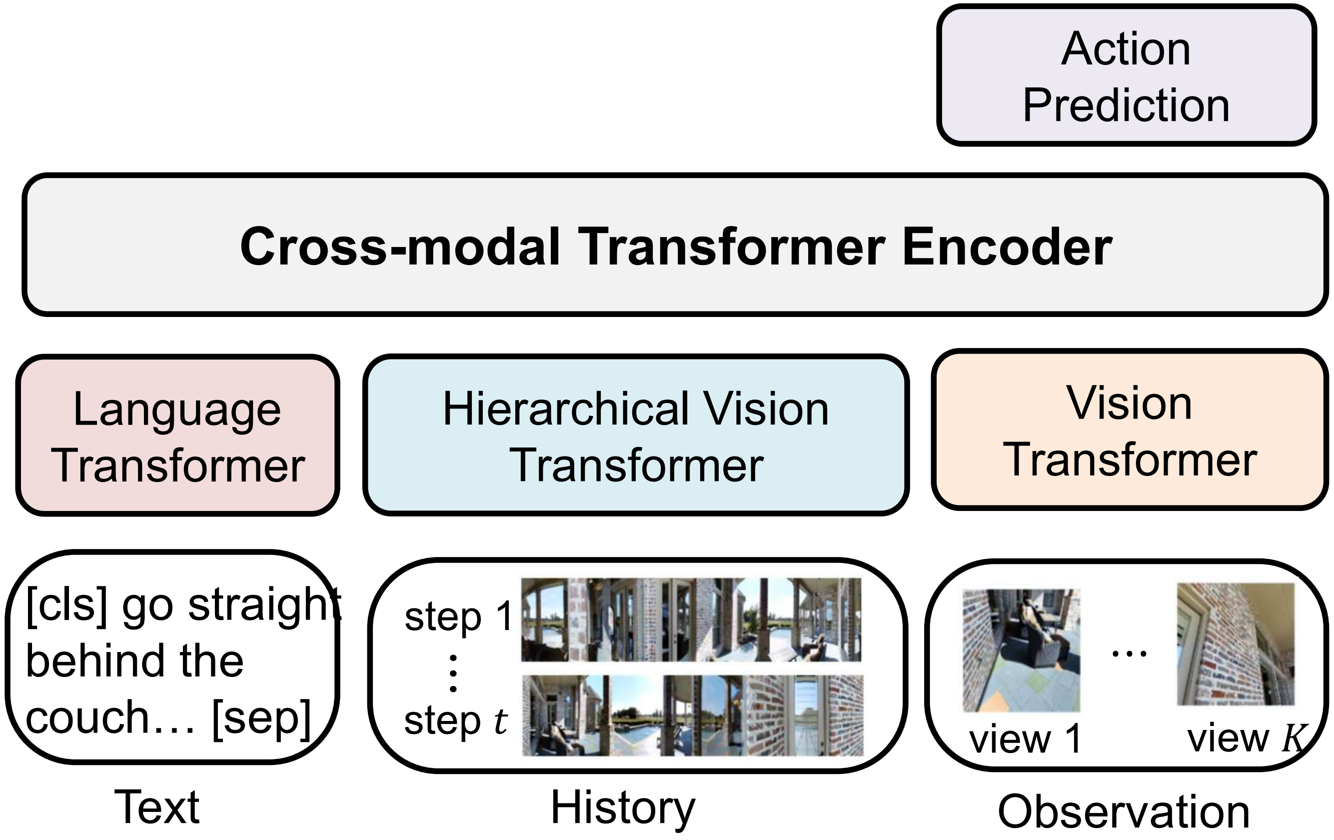


- Design of a fully transformer based model

[HAMT, S. Chen, PL. Guhur, C. Schmid, I. Laptev, Neurips'21]

History Aware Multimodal transformer (HAMT)

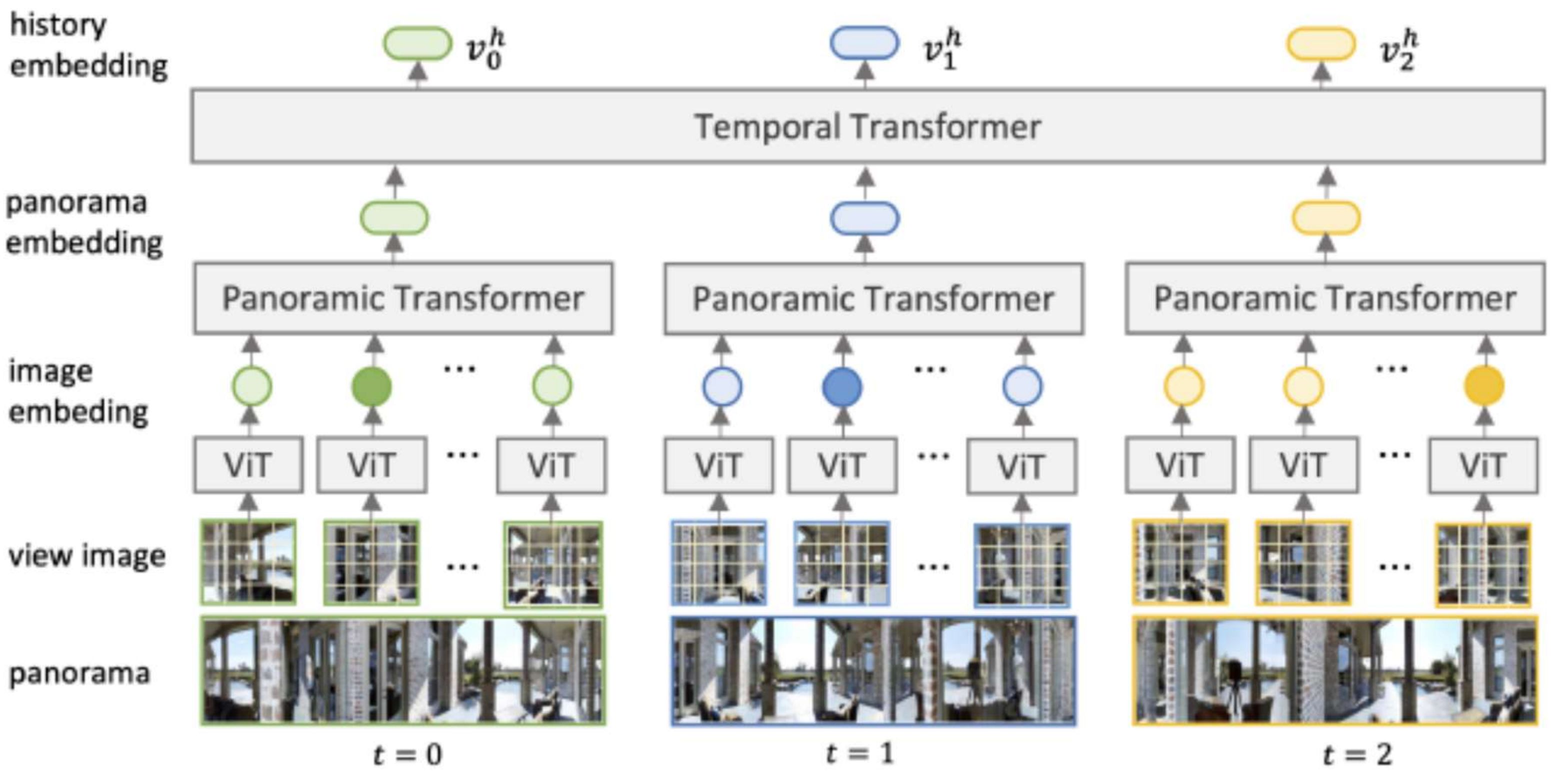
- Long-horizon history modeling for learning temporal dependency of observations and actions



[HAMT, S. Chen, PL. Guhur, C. Schmid, I. Laptev, Neurips'21]

Hierarchical history encoding

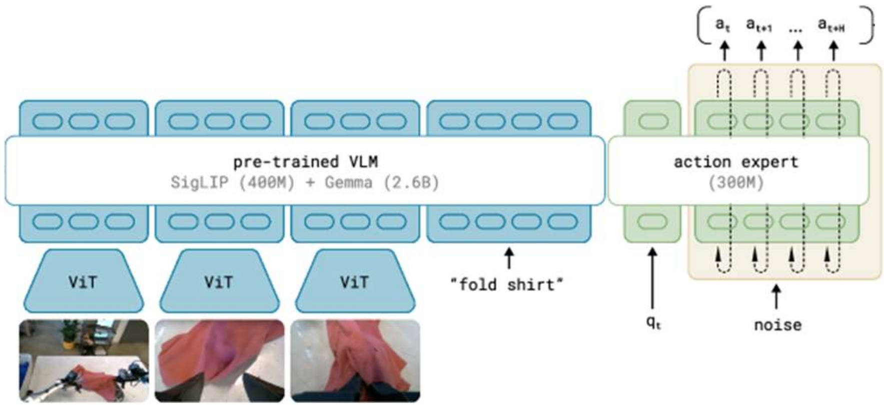
- ViT for single view image encoding
- Panoramic transformer for view encoding
- Temporal transformer for sequence encoding



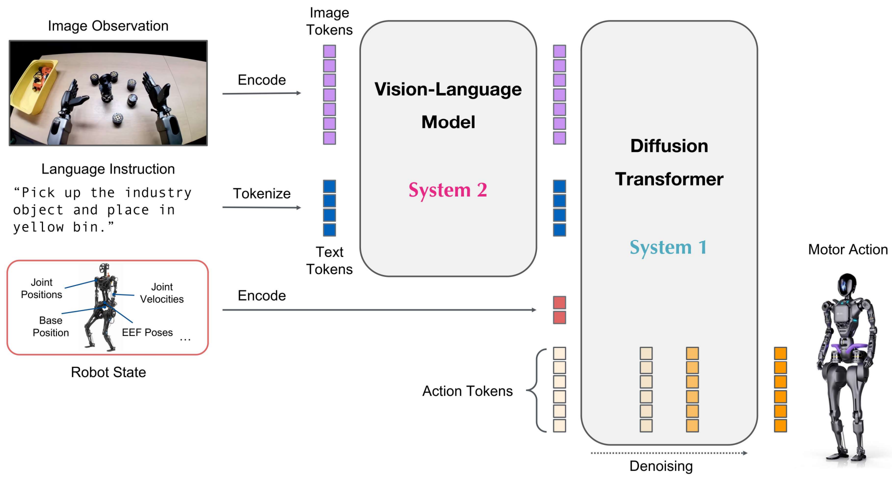
[HAMT, S. Chen, PL. Guhur, C. Schmid, I. Laptev, Neurips'21]

Vision-Language-Action Models (VLAs)

- Robot manipulation following language instructions
- VLAs: map observations directly to low-level actions



π_0 : A Vision-Language-Action Flow Model for General Robot Control. (Physical Intelligence, 2024)



Gr00t N1: An open foundation model for generalist humanoid robots. (Nvidia, 2025)

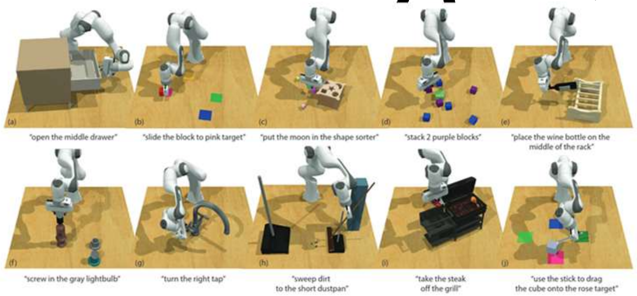
Overview

- *LLM-guided 3D policy for robotic manipulation*
- A critic for robotics - Guardian

LLM-guided 3D policy for robotic manipulation

- Approach with an LLM-guided 3D policy for manipulation
  - an LLM for task planning
  - an VLM for object localization
  - a 3D point cloud based policy for trajectory planning

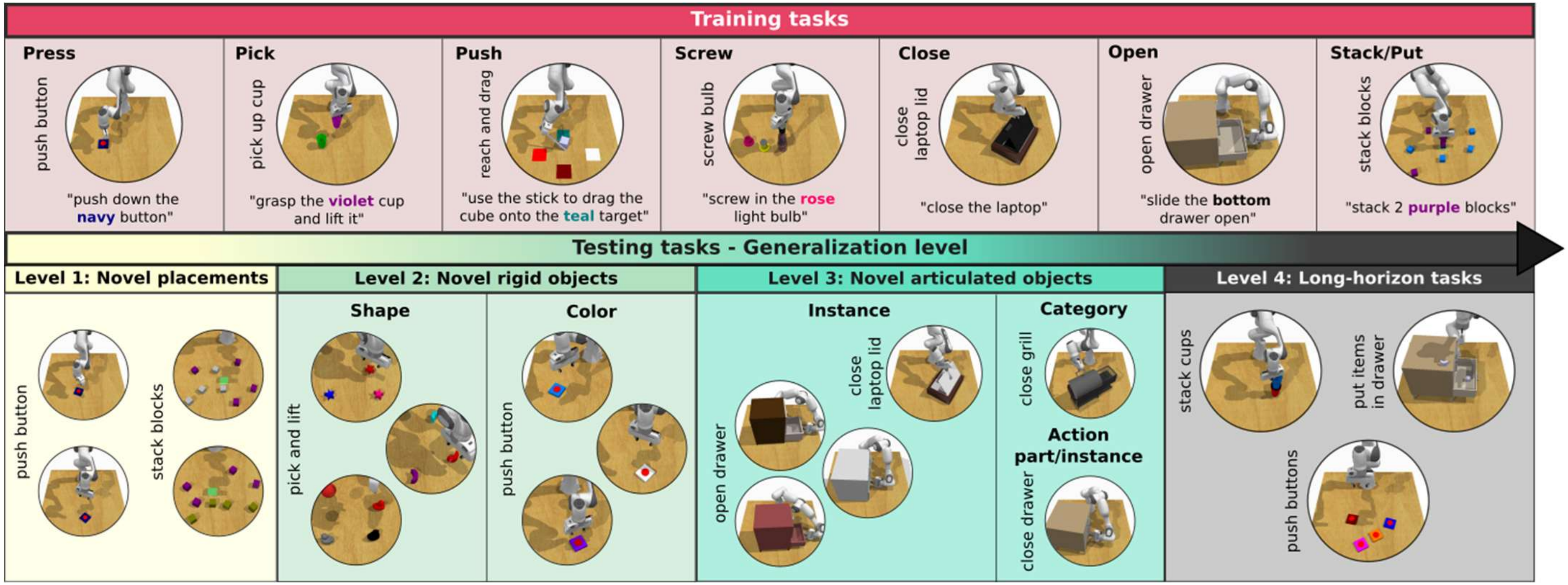
- Need for better benchmarks
  - RLBench-18Task framework [1] train, test on very similar setups



=> We introduce GEMBench to evaluate generalization

[1] Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation, CoRL 2022. Shridhar et al.

GEMBench Benchmark

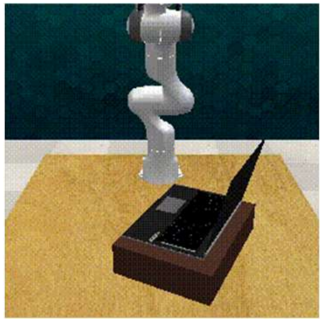


https://www.di.ens.fr/willow/research/gembench/

[Generalizable Robotic Manipulation: A Benchmark & LLM-guided 3D Policy, R. Garcia, S. Chen, C. Schmid, ICRA'25]

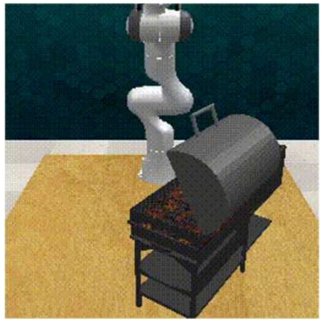
GEMBench – level 3

Level 3: Novel articulated objects



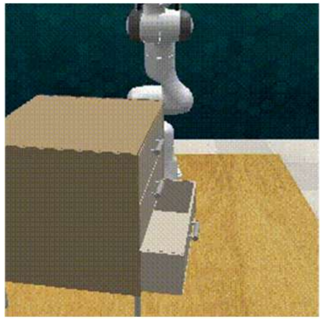
“close **laptop lid**”

Novel Category



“close **the grill**”

Novel Action/Object

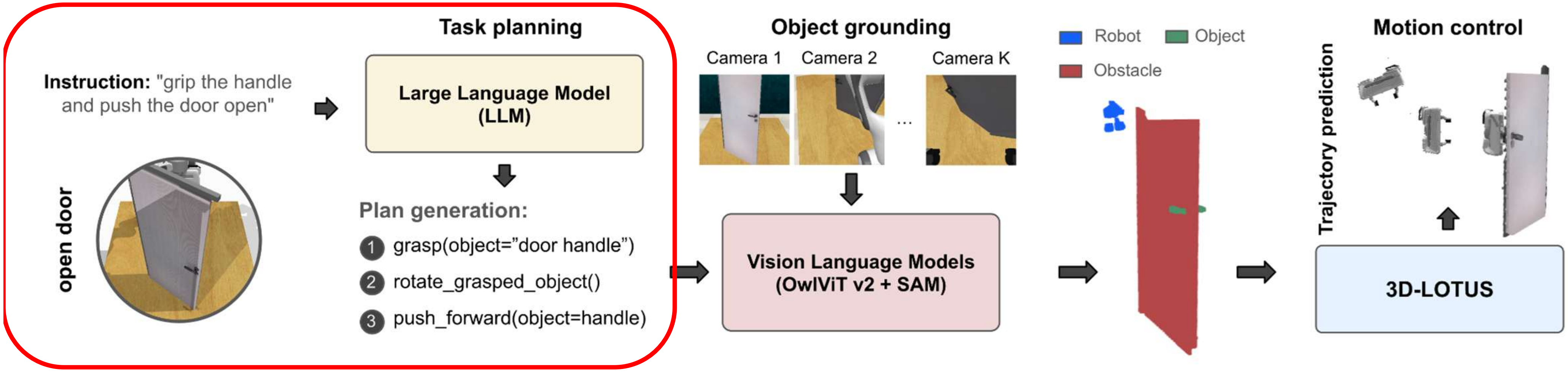


“close **bottom drawer**”

[Generalizable Robotic Manipulation: A Benchmark & LLM-guided 3D Policy, R. Garcia, S. Chen, C. Schmid, ICRA’25]

3D-Lotus++ for long horizon planning

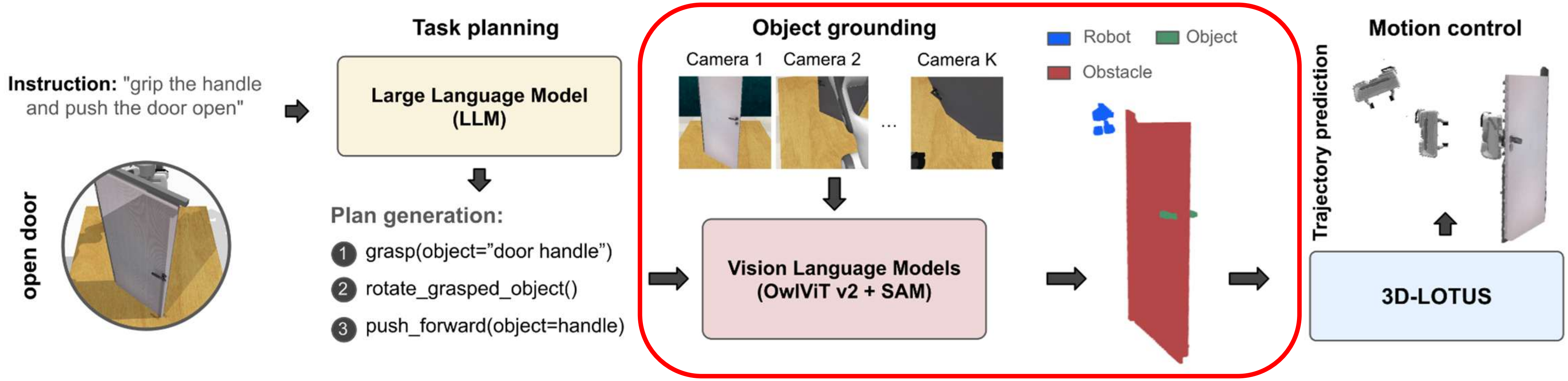
1) Task decomposition with an LLM into action primitives
   – LLaMa3-8B + prompt with task requirements & few-shot examples



[Generalizable Robotic Manipulation: A Benchmark & LLM-guided 3D Policy, R. Garcia, S. Chen, C. Schmid, ICRA'25]

3D-Lotus++ for long horizon planning

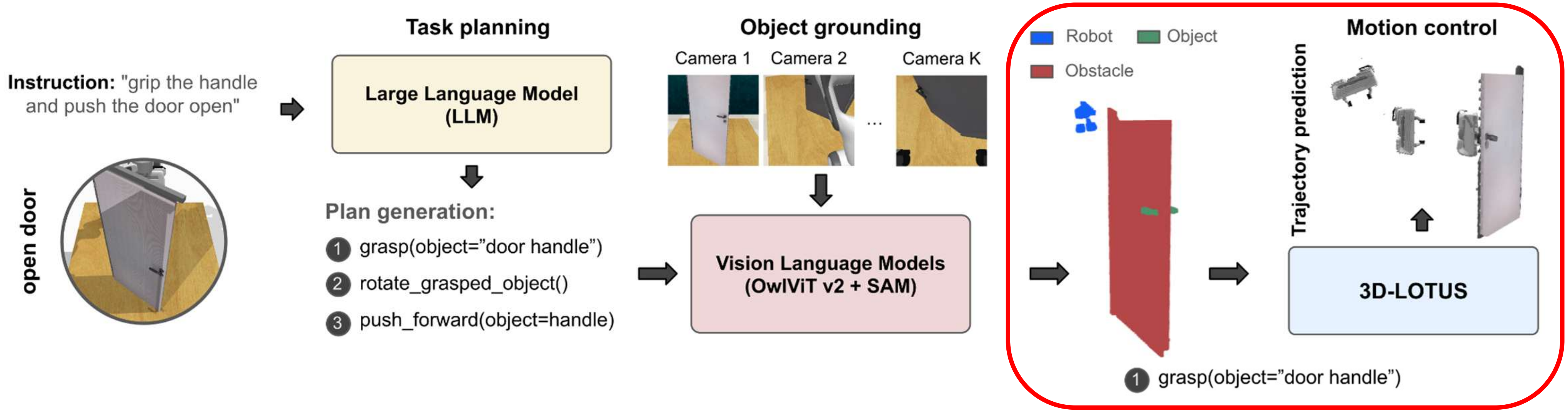
1) Task decomposition with an LLM into action primitives
2) Object grounding with an VLM
   – Open vocabulary object detector OwlViTv2 + SAM for segmentation



[Generalizable Robotic Manipulation: A Benchmark & LLM-guided 3D Policy, R. Garcia, S. Chen, C. Schmid, ICRA'25]

3D-Lotus++ for long horizon planning

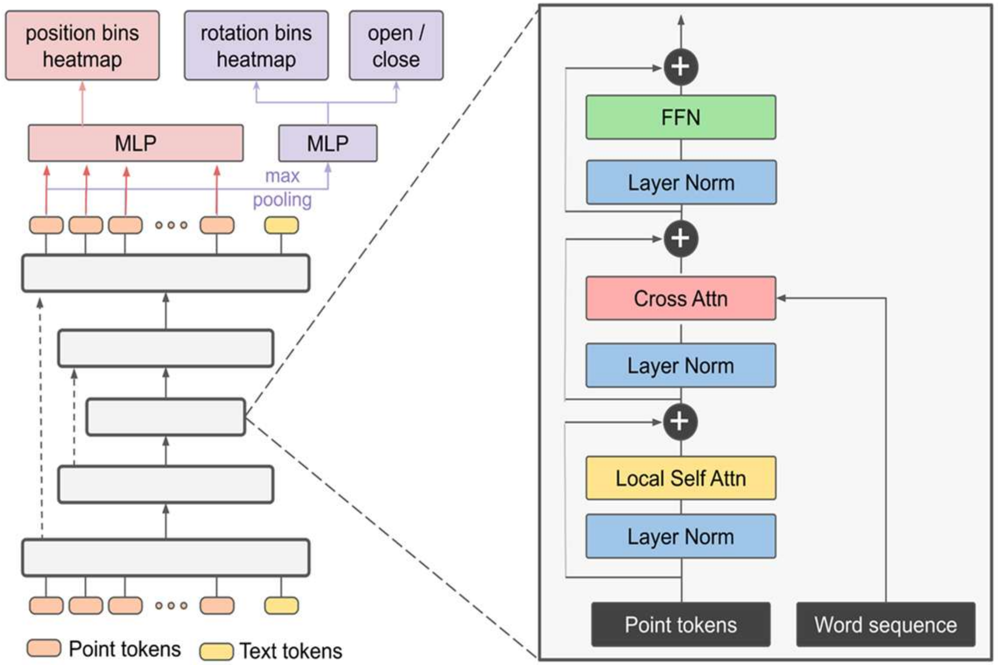
1) Task decomposition with an LLM into action primitives
2) Object grounding with an VLM
3) Novel efficient 3D robot manipulation policy, 3D-Lotus



[Generalizable Robotic Manipulation: A Benchmark & LLM-guided 3D Policy, R. Garcia, S. Chen, C. Schmid, ICRA'25]

3D-Lotus: 3D robot manipulation policy

- Language-conditioned 3D point cloud transformer



- Use of point cloud transformer v3 (PTV3) [5] as encoder
- Addition of language instruction

[5] Point Transformer V3: Simpler, Faster, Stronger. CVPR 2024. Wu et al.

SOTA comparison of 3D-Lotus on GemBench

Method	L1	L2	L3	L4
Hiveformer [17]	$60.3_{\pm 1.5}$	$26.1_{\pm 1.4}$	$35.1_{\pm 1.7}$	$0.0_{\pm 0.0}$
PolarNet [2]	$77.7_{\pm 0.9}$	$37.1_{\pm 1.4}$	$38.5_{\pm 1.7}$	$0.1_{\pm 0.2}$
3D diffuser actor [35]	$91.9_{\pm 0.8}$	$43.4_{\pm 2.8}$	$37.0_{\pm 2.2}$	$0.0_{\pm 0.0}$
RVT-2 [37]	$89.1_{\pm 0.8}$	$51.0_{\pm 2.3}$	$36.0_{\pm 2.2}$	$0.0_{\pm 0.0}$
3D-LOTUS	$\mathbf{94.3}_{\pm 1.4}$	$49.9_{\pm 2.2}$	$38.1_{\pm 1.1}$	$0.3_{\pm 0.3}$
3D-LOTUS++	$68.7_{\pm 0.6}$	$\mathbf{64.5}_{\pm 0.9}$	$\mathbf{41.5}_{\pm 1.8}$	$\mathbf{17.4}_{\pm 0.4}$

Comparison on 4 levels of GemBench (avg. SR)

Most SOT models fail to generalize

SOTA comparison of 3D-Lotus++ on GemBench

Method	L1	L2	L3	L4
Hiveformer [17]	$60.3_{\pm 1.5}$	$26.1_{\pm 1.4}$	$35.1_{\pm 1.7}$	$0.0_{\pm 0.0}$
PolarNet [2]	$77.7_{\pm 0.9}$	$37.1_{\pm 1.4}$	$38.5_{\pm 1.7}$	$0.1_{\pm 0.2}$
3D diffuser actor [35]	$91.9_{\pm 0.8}$	$43.4_{\pm 2.8}$	$37.0_{\pm 2.2}$	$0.0_{\pm 0.0}$
RVT-2 [37]	$89.1_{\pm 0.8}$	$51.0_{\pm 2.3}$	$36.0_{\pm 2.2}$	$0.0_{\pm 0.0}$
3D-LOTUS	$\mathbf{94.3}_{\pm 1.4}$	$49.9_{\pm 2.2}$	$38.1_{\pm 1.1}$	$0.3_{\pm 0.3}$
3D-LOTUS++	$68.7_{\pm 0.6}$	$\mathbf{64.5}_{\pm 0.9}$	$\mathbf{41.5}_{\pm 1.8}$	$\mathbf{17.4}_{\pm 0.4}$

Comparison on 4 levels of GemBench (avg. SR)

3D-Lotus++ with an additional planning module generalizes to novel objects, actions and long-horizon tasks

Ablation of 3D-Lotus++ modules

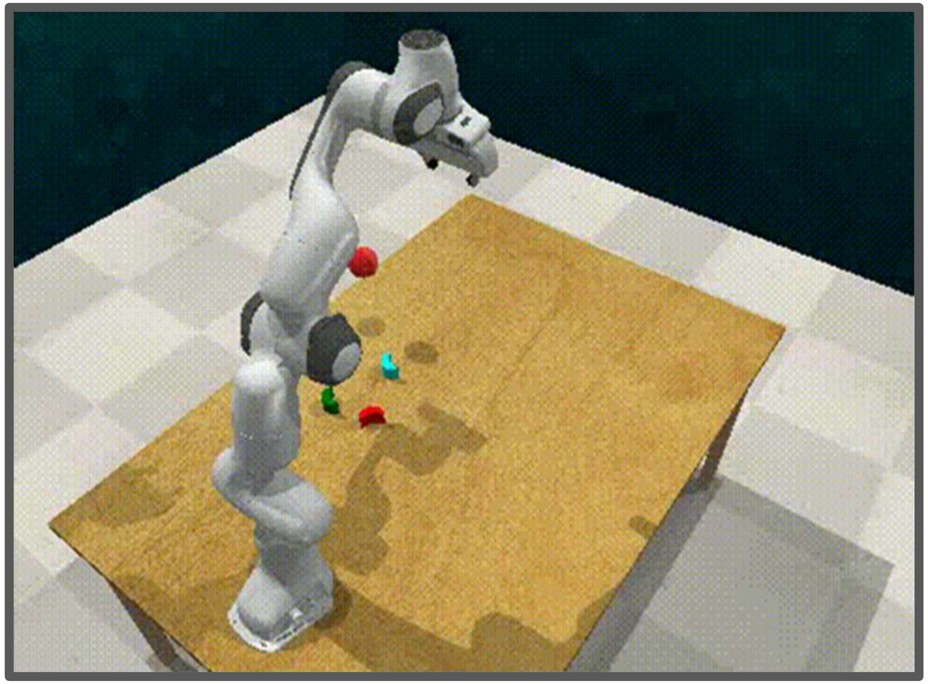
Method		L1	L2	L3	L4
3D-LOTUS		**94.3**$_{\pm 1.4}$	49.9$_{\pm 2.2}$	38.1$_{\pm 1.1}$	0.3$_{\pm 0.3}$
3D-LOTUS++		68.7$_{\pm 0.6}$	**64.5**$_{\pm 0.9}$	**41.5**$_{\pm 1.8}$	**17.4**$_{\pm 0.4}$

Task Planning	Object Grounding	L1	L2	L3	L4
GT	GT	**92.6**$_{\pm 0.7}$	**80.1**$_{\pm 0.5}$	**47.8**$_{\pm 1.4}$	**31.5**$_{\pm 1.1}$
GT	VLM	71.0$_{\pm 1.7}$	66.3$_{\pm 0.9}$	46.0$_{\pm 1.5}$	19.4$_{\pm 1.5}$
LLM	VLM	68.7$_{\pm 0.6}$	64.5$_{\pm 0.9}$	41.5$_{\pm 1.8}$	17.4$_{\pm 0.4}$

Significant gain with GT task planning & object grounding

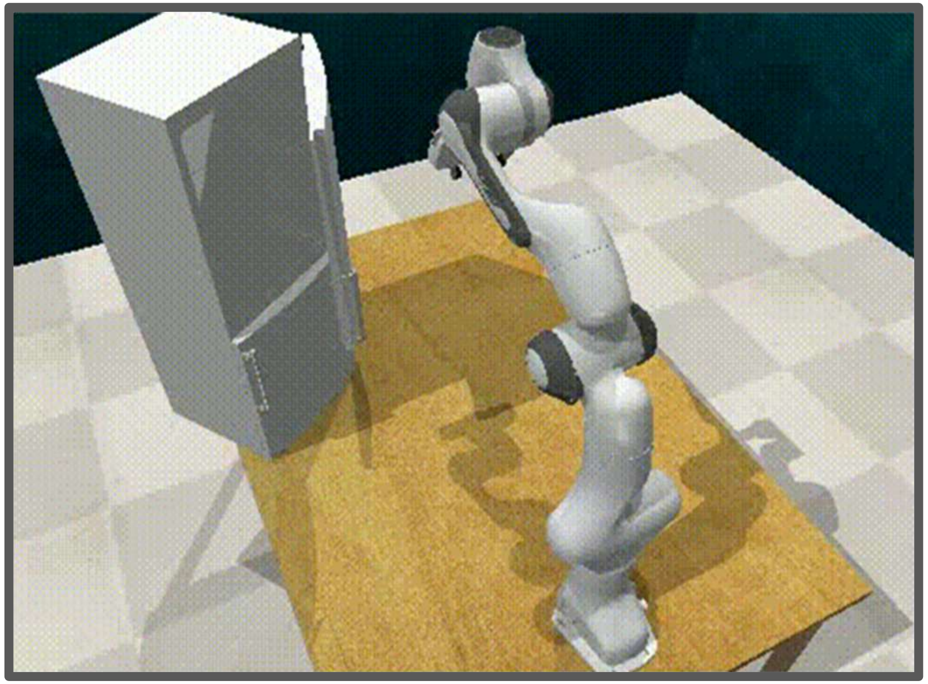
3D-LOTUS++: GEMBench success cases

Generalization to new shapes



“pick up the **red moon** and lift it up to the target”

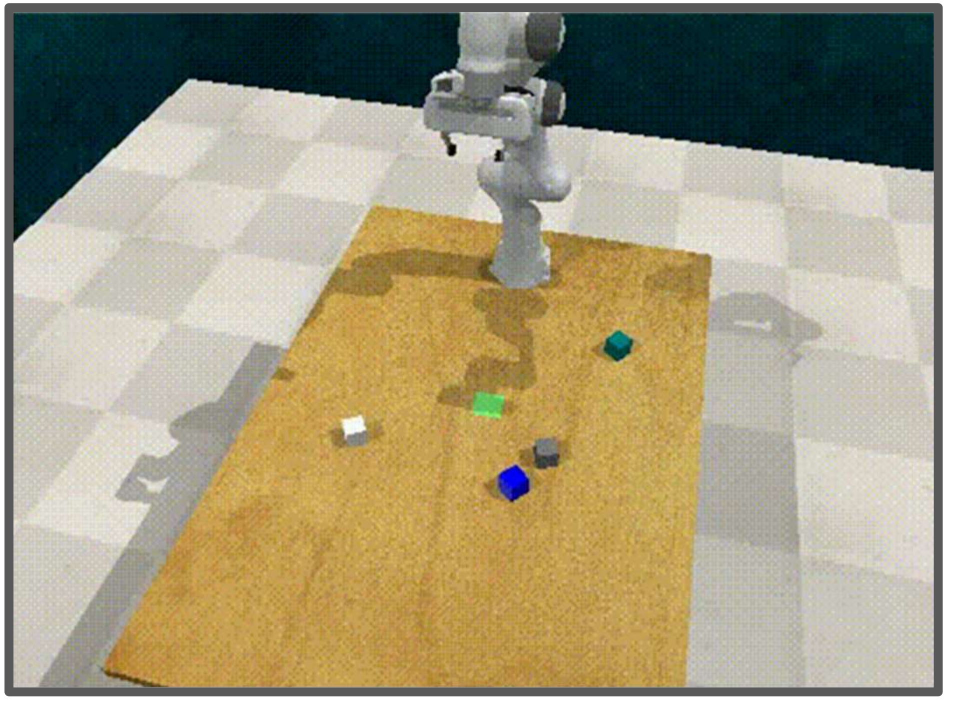
Generalization to new object instances



“close the fridge door”

3D-LOTUS++: GEMBench success cases

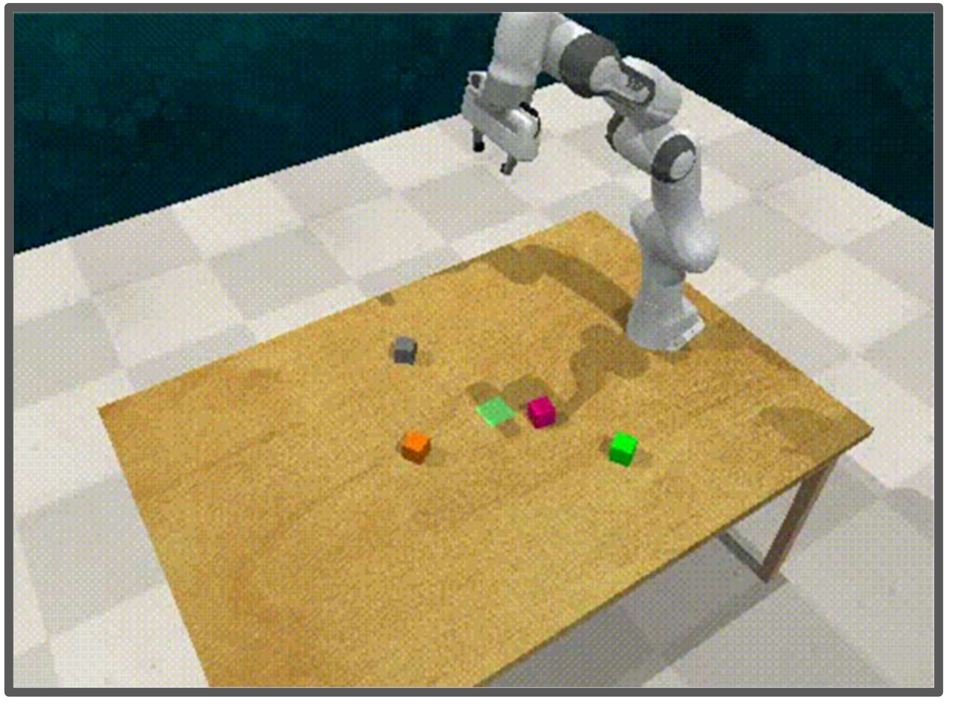
Long-horizon task generalization



“pick the **white** cube and put it on the **green** marker,
then stack the **teal** block on top of the **white**,
finally stack the **blue** block on top of the stacked cubes”

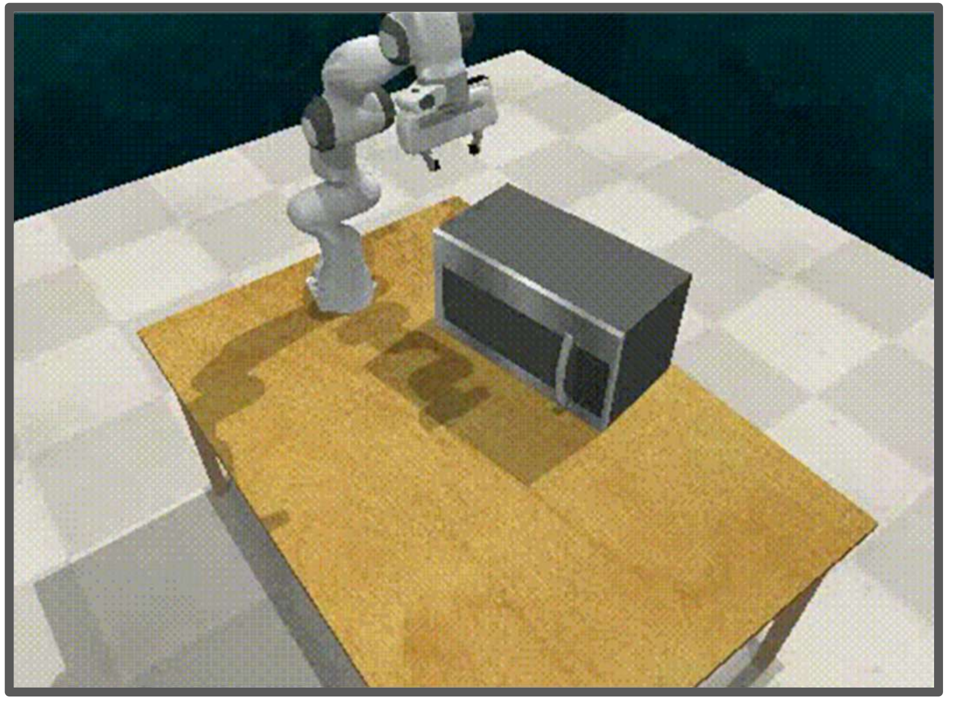
3D-LOTUS++: GEMBench failure cases

Confuse green cube with green marker
(object grounding failure)



“Pick the **orange** block and put it on the **green** marker,
then stack the **gray** block on top of the **orange**,
then stack the **lime** on top of the **gray**,
finally stack the **rose** block on top of the stacked cubes”

Open action in microwave very different from training data
(trajectory prediction failure)



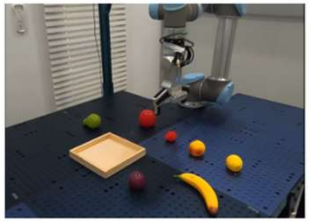
“**open** the **microwave door**”

Real world setup

Training set
7 task variations
20 episodes/taskvar

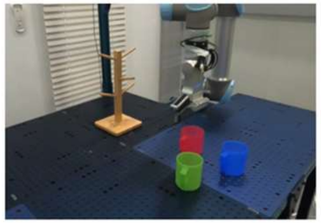
stack yellow cup in pink cup

stack **navy** cup in yellow cup

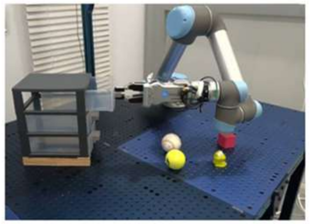


put **strawberry** in box

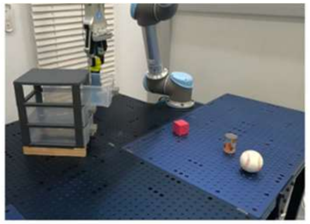
put peach in box



hang **pink** mug

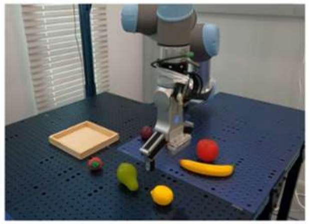


open the top drawer



put **frog** in the top drawer

Unseen test set
5 task variations

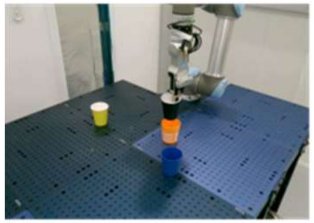


put **lemon** in box

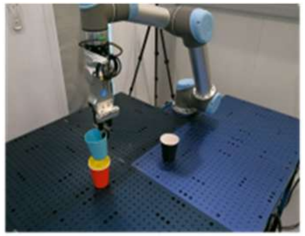
put banana in box

put plum in box, then **corn** in box

put **grapes** in yellow plate,
then banana in **pink** plate



stack the (black/red) cup
on top of the (orange/black) cup



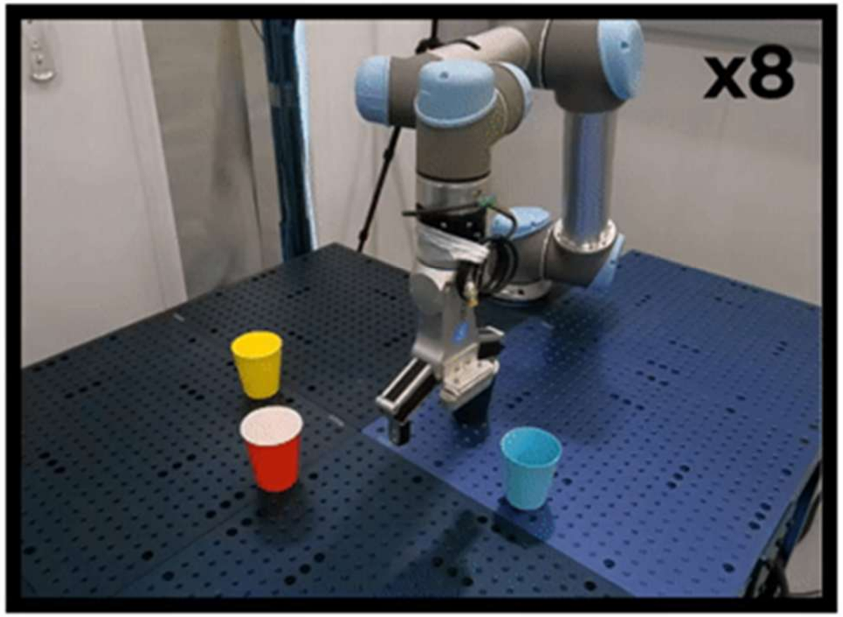
place the yellow cup inside the
red cup, then the cyan cup on top

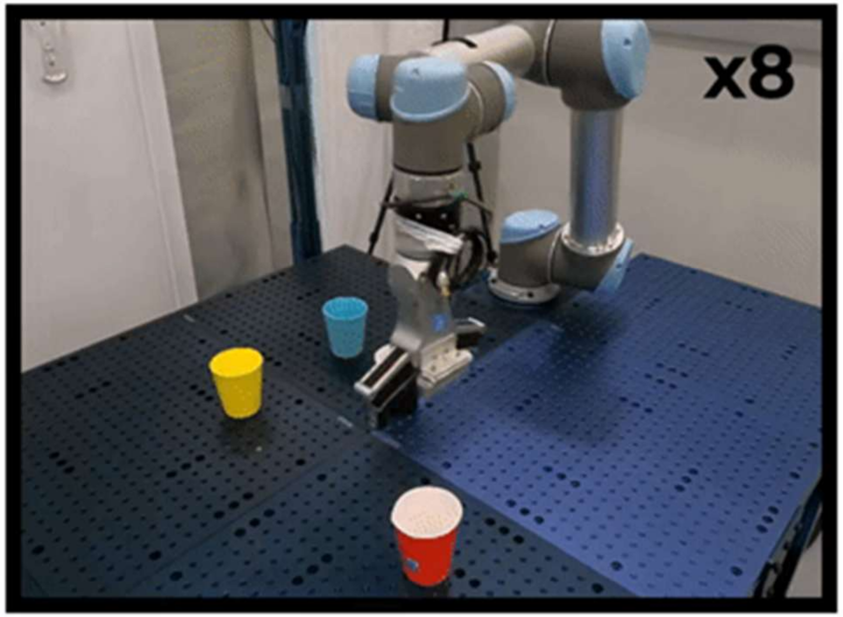
Comparison with state of the art for “unseen” task variations

	3D-LOTUS	3D-LOTUS++
stack_cup (black → orange)	0/10	**8/10**
stack_cup (red → yellow)	0/10	**7/10**
stack_cups (cyan → yellow → red)	0/10	**7/10**
put_fruit_in_box (banana)	0/10	**9/10**
put_fruit_in_box (lemon)	0/10	**7/10**
put_food_in_box (tuna, corn)	0/10	**8/10**
put_fruits_in_plates (banana)	0/10	**9/10**
Average	0/10	**7.9/10**

3D-LOTUS++ outperforms PolarNet

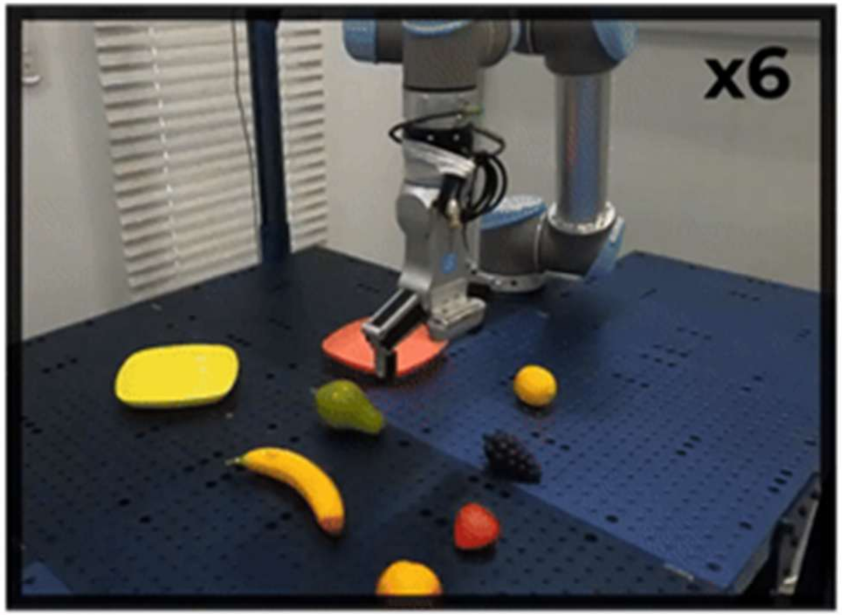
3D-LOTUS++: Stack cups (cyan → yellow → red)

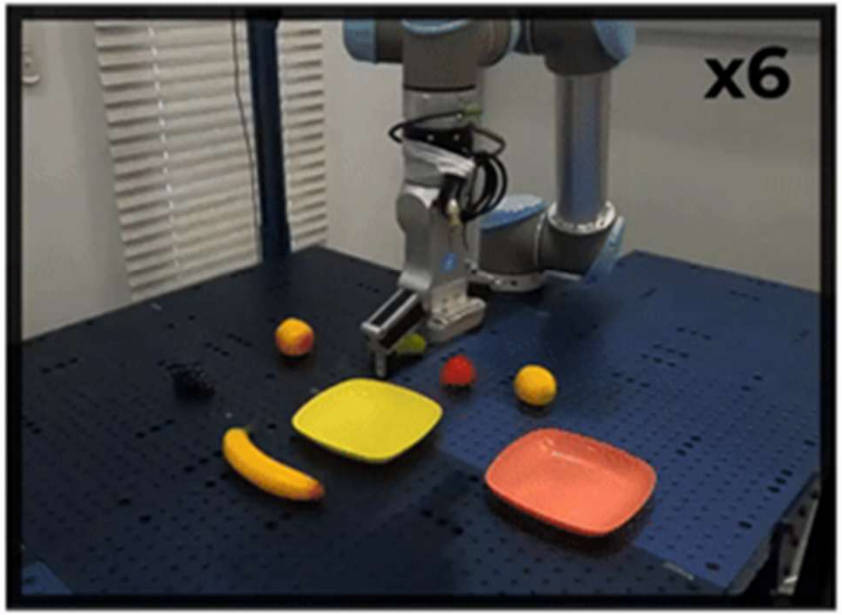




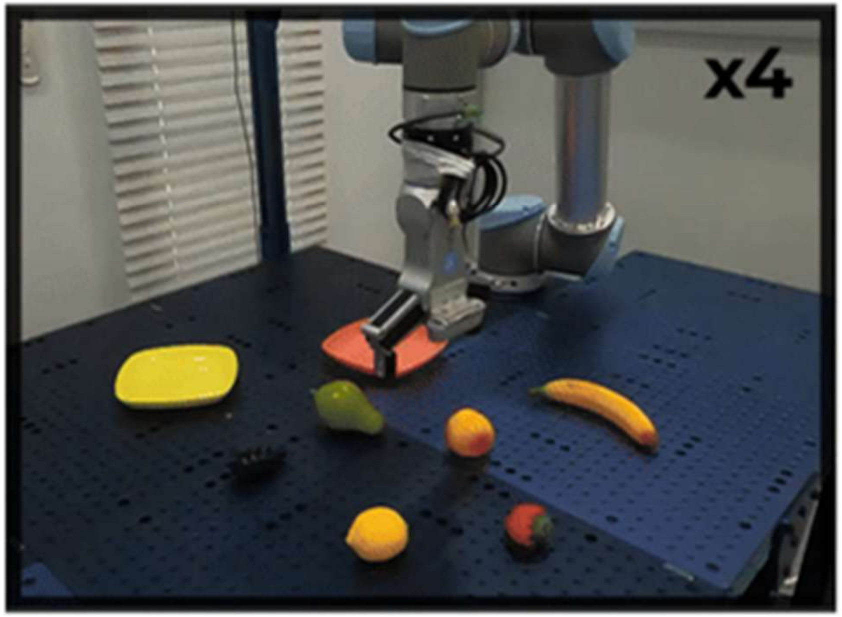
"place the yellow cup inside the red cup, then the cyan cup on top"

3D-LOTUS++: Put food in plates (grapes → yellow plate, banana → pink plate)



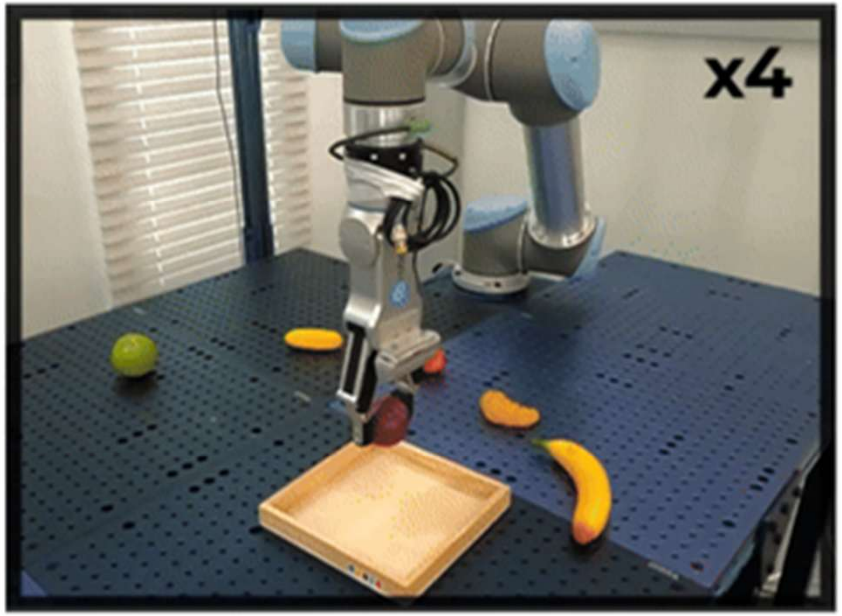


3D-LOTUS++: Failure cases



Imprecise grasping for difficult shape

“put the **grape** in the yellow plate”

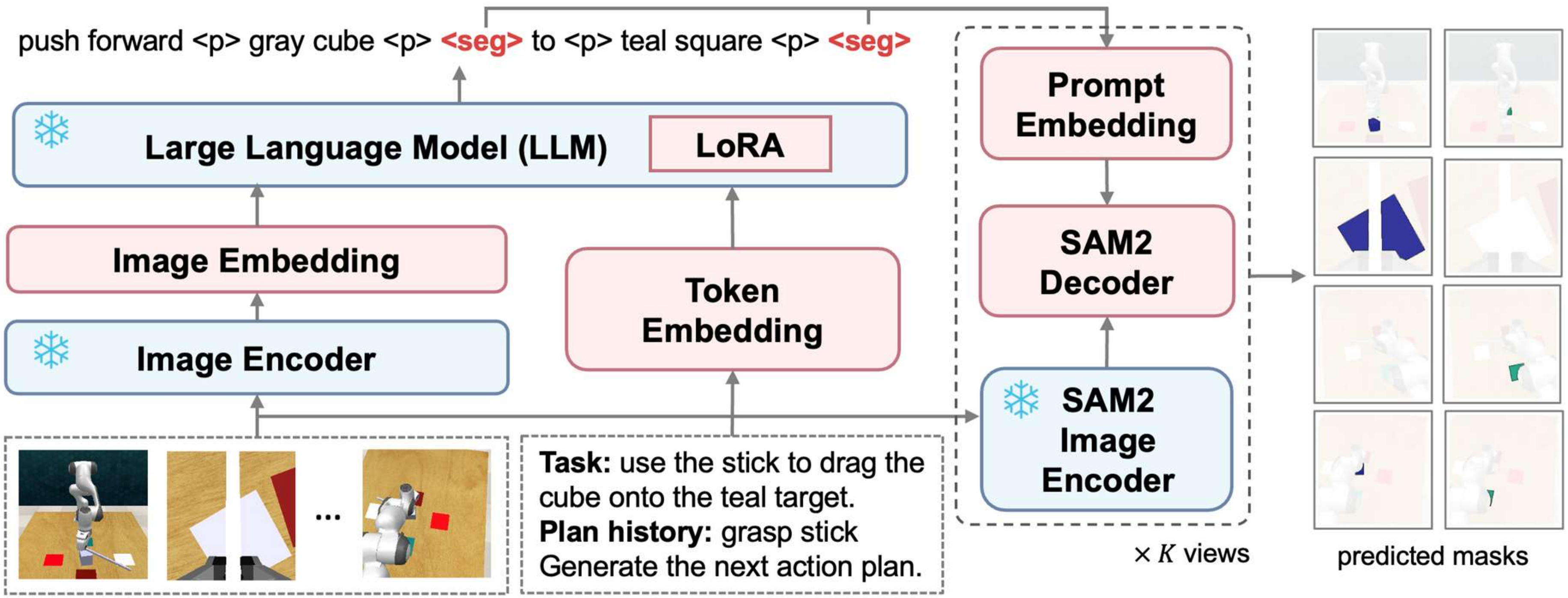


Confuse objects with similar color-shape

“put the plum in the box, then put the **corn** in the box”

Gondola: Grounded Task Planning

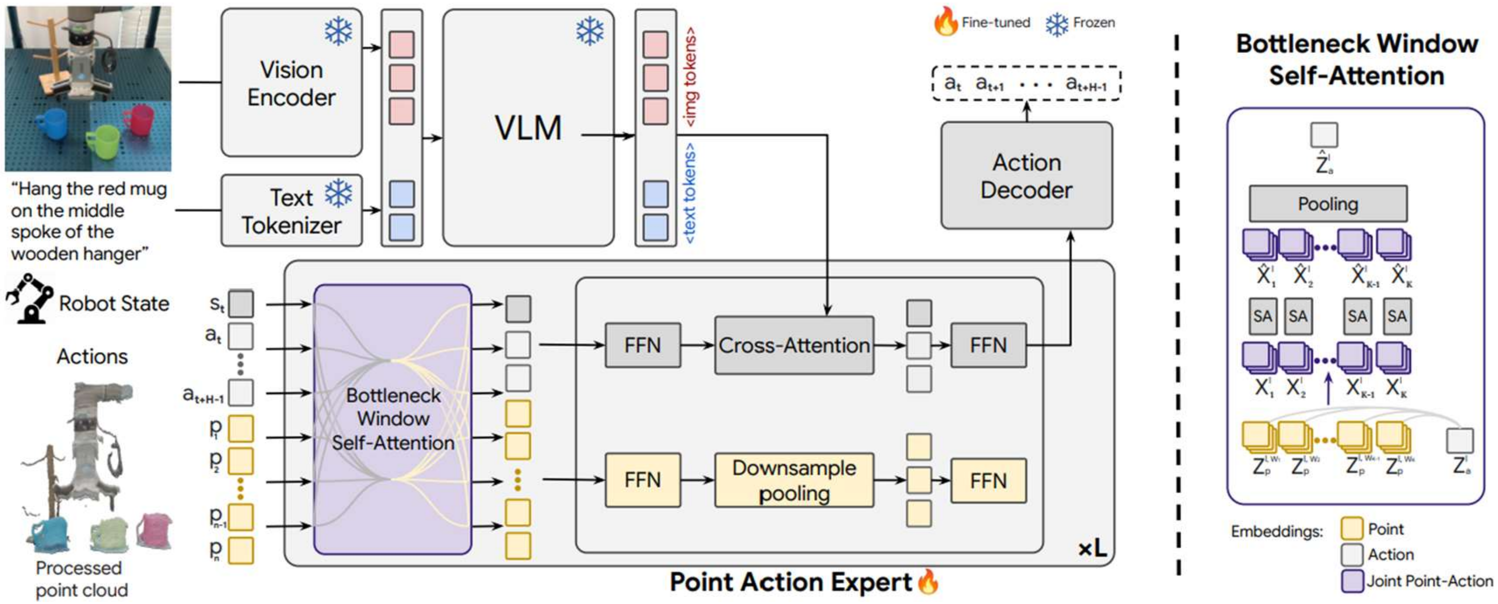
- Unified textual plan generation and image segmentation



[Gondola: Grounded Vision Language Planning for Robotic Manipulation. Chen, Garcia, Pacaud, Schmid. arXiv'26]

VLA with Multi-Scale Point-Action Interaction

- Model based on 3D point cloud input
- Multi-scale point-action interaction



[PointACT: Vision-Language-Action Models with Multi-Scale Point-Action Interaction. Chen, Pacaud, Schmid. RSS'26.]

VLA with Multi-Scale Point-Action Interaction

TABLE II: **Success rate (%) on RLBench 10 tasks.** We reproduce EO1 [52] and evaluate both EO1 and our PointACT using 100 episodes per task. The results for the other methods are directly taken from [41], where each method is evaluated over 20 episodes.

	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean
ManipLLM (7B) [36]	50	80	40	20	80	35	10	25	15	20	38
OpenVLA (7B) [31]	65	40	75	60	80	20	35	15	10	10	41
π_0 [6] (2.6B)	90	60	100	30	90	25	35	75	5	45	55
CogACT (7B) [35]	80	85	90	65	90	50	60	35	25	25	60
HybridVLA (7B) [41]	85	95	100	90	100	50	50	70	50	50	74
EO1 (3B) [52]	97	99	100	95	83	46	76	40	61	35	73.2
ACT3D [17]	94	62	99	80	87	53	97	31	31	11	64.5
PointACT (3B)	91	99	96	59	81	99	99	69	90	40	**82.3**

[PointACT: Vision-Language-Action Models with Multi-Scale Point-Action Interaction. Chen, Pacaud, Schmid. RSS'26.]

Overview

- LLM-guided 3D policy for robotic manipulation
- *A critic for robotics - Guardian*

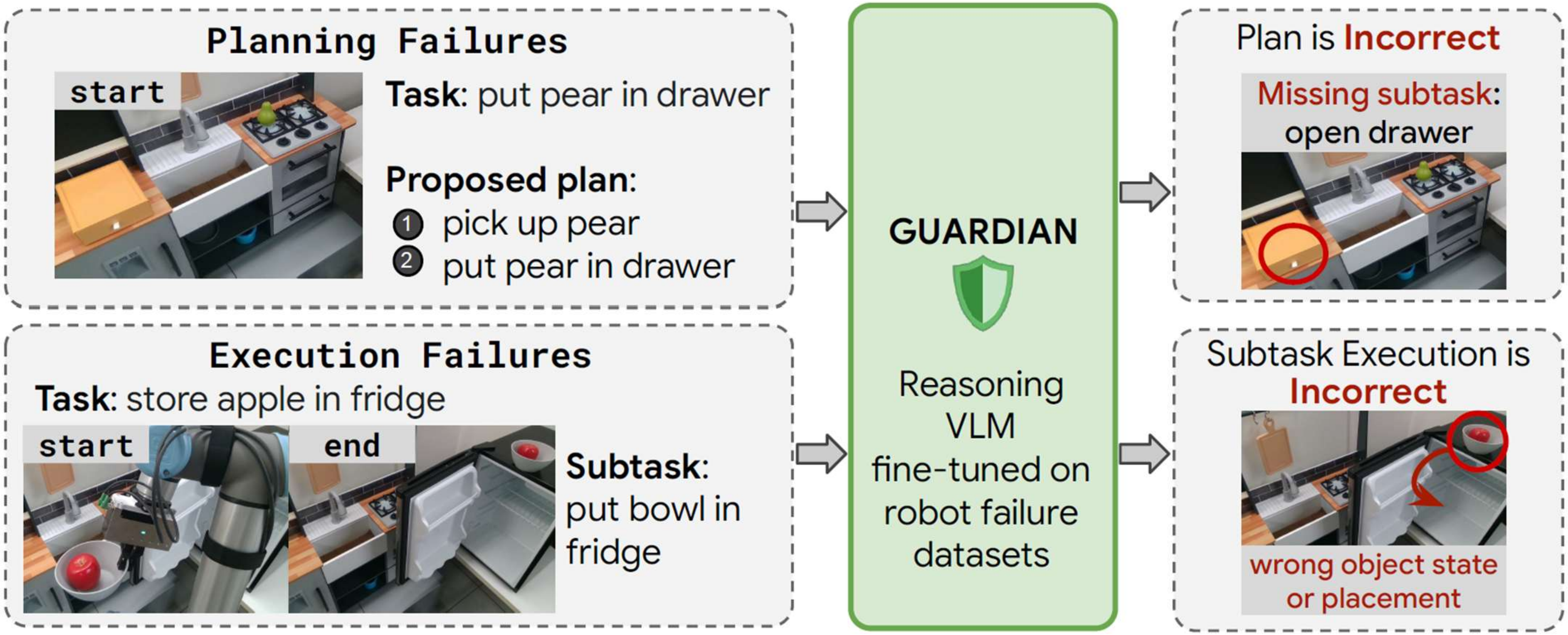
A critic for robotics - Guardian

- Robot execution can fail, but difficult to detect automatically
- Failure detection is necessary to re-plan
- Critics have shown to successful for LLMs in general
- We introduce a critic (VLM) to detect planning and execution errors

[Guardian: Detecting Robotic Planning & Execution Errors with VLMs. P. Pacaud, R. Garcia, S. Chen, C. Schmid, arXiv]

A critic for robotics - Guardian

- A critic to detect planning and execution errors



Planning verification

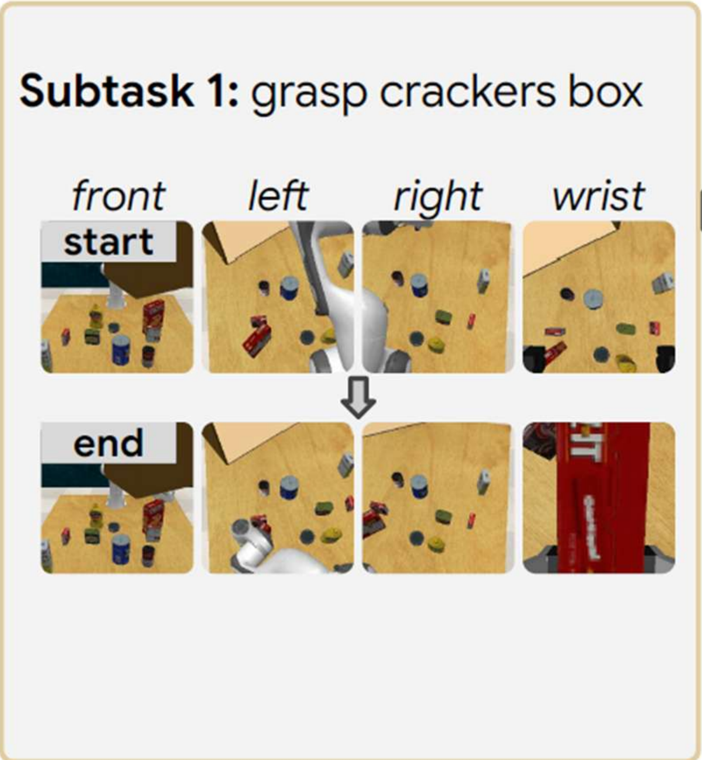
Task: Pick up the crackers box and place it in the cupboard, then place the tuna can next to it, and then move the spam can inside the cupboard

1. grasp crackers box
2. move grasped crackers box to cupboard
3. release crackers box
4. grasp tuna can
5. release tuna can
6. move grasped spam can to inside cupboard
7. release spam can

Planning Verification

<think> **1.** **subtask 1**: This step is logical **2.** **subtask 2**: This step is logical **3.** **subtask 3**: This step is logical **4.** **subtask 4**: This step is logical **5.** **subtask 5**: This step is not logical because it lacks the subtask of moving the grasped tuna can next to the crackers box before releasing it. **6.** **subtask 6**: This step is not logical because it lacks the subtask of grasping the spam can before moving it to the cupboard. **7.** **subtask 7**: This step is logical only after ensuring the spam can is grasped and moved to the cupboard. **Analysis:** The plan is missing subtasks for moving the tuna can next to the crackers box and for grasping the spam can before moving it to the cupboard. </think>
<answer> False </answer>
<category> missing subtasks: one or several subtasks are missing </category>

Execution verification

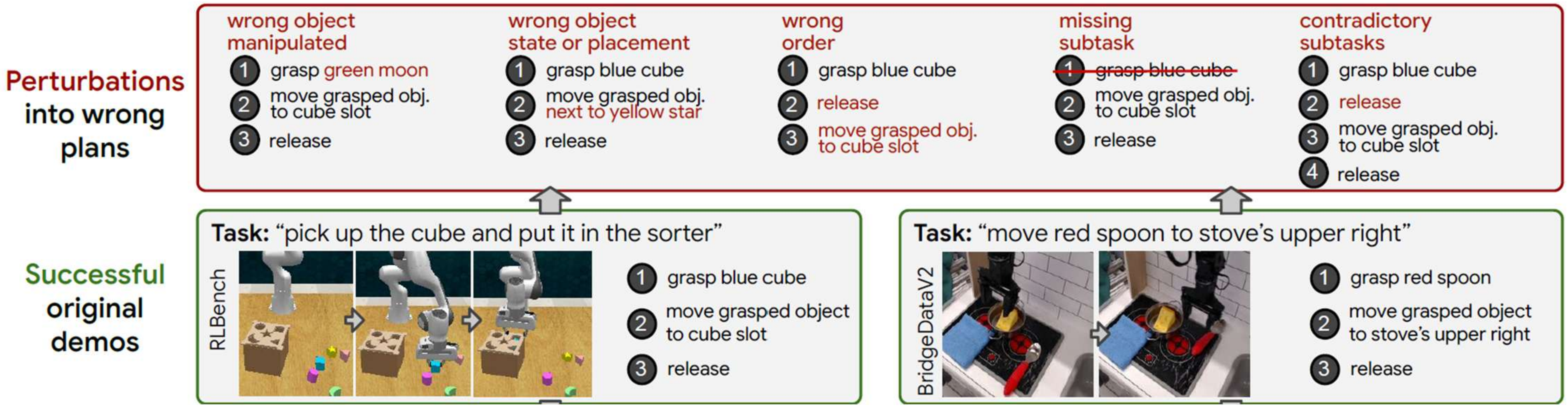


Execution Verification

<think> **1.** **Start Image Description:** The scene shows a wooden tabletop with various items including a chocolate jello box, strawberry jello box, soup can, tuna can, spam can, sugar box, coffee can, mustard bottle, and crackers box, with the robot arm positioned above the table. **2.** **End Image Description:** The scene remains similar with the same items on the table, but the robot gripper is now closed around the crackers box, indicating it has grasped the object. **3.** **Analysis:** The robot gripper successfully moved to the crackers box and is holding it, demonstrating the successful completion of the subtask. </think>
<answer> True </answer>
<category> success </category>

Guardian – training data

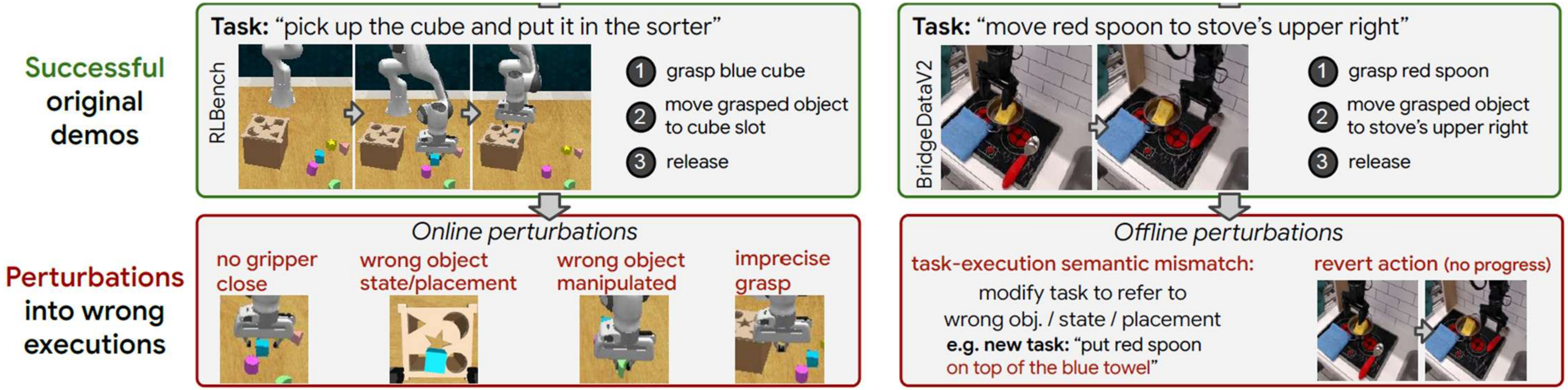
- Training data for planning



  - Start from successful plans + generate wrong ones
  - For RL Bench (simulator) and BrideDataV2 (real robot data)

Guardian – training data

- Training data for execution



  - Start from successful executions + generate incorrect ones
  - For RL Bench (simulator → online) and BrideDataV2 (real robot data → offline by changing text instruction)

Dataset statistics

		Training		Validation		Test	
Dataset	**Env.**	Exec	Plan	Exec	Plan	Exec	Plan
FailCoT (Ours)							
RLBench-Fail	Sim	12358	5808	1000	500	1000	500
BridgeDataV2-Fail	Real	7830	4880	1000	500	1000	500
Real-World Robot Failure Detection Benchmarks							
UR5-Fail (Ours)	Real	-	-	-	-	140	140
RoboFail [7]	Real	-	-	-	-	153	30
RoboVQA [27]	Real	-	-	-	-	357	-

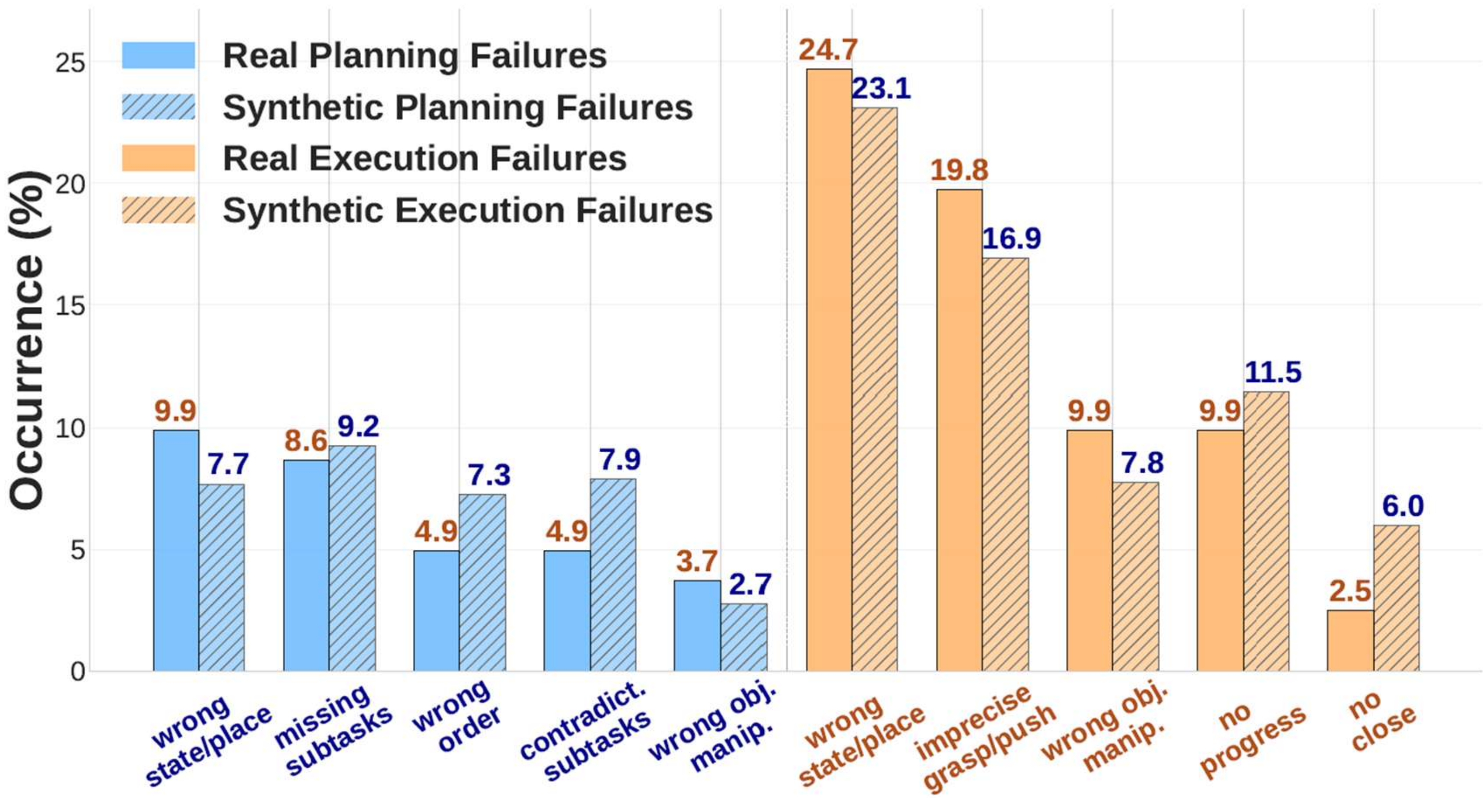
- Large-scale synthetic dataset FailCoT with RLBench + BridgeData
  - Balanced execution and planning failures

Dataset statistics

Dataset	Env.	Training		Validation		Test	
		Exec	Plan	Exec	Plan	Exec	Plan
FailCoT (Ours)							
RLBench-Fail	Sim	12358	5808	1000	500	1000	500
BridgeDataV2-Fail	Real	7830	4880	1000	500	1000	500
Real-World Robot Failure Detection Benchmarks							
UR5-Fail (Ours)	Real	-	-	-	-	140	140
RoboFail [7]	Real	-	-	-	-	153	30
RoboVQA [27]	Real	-	-	-	-	357	-

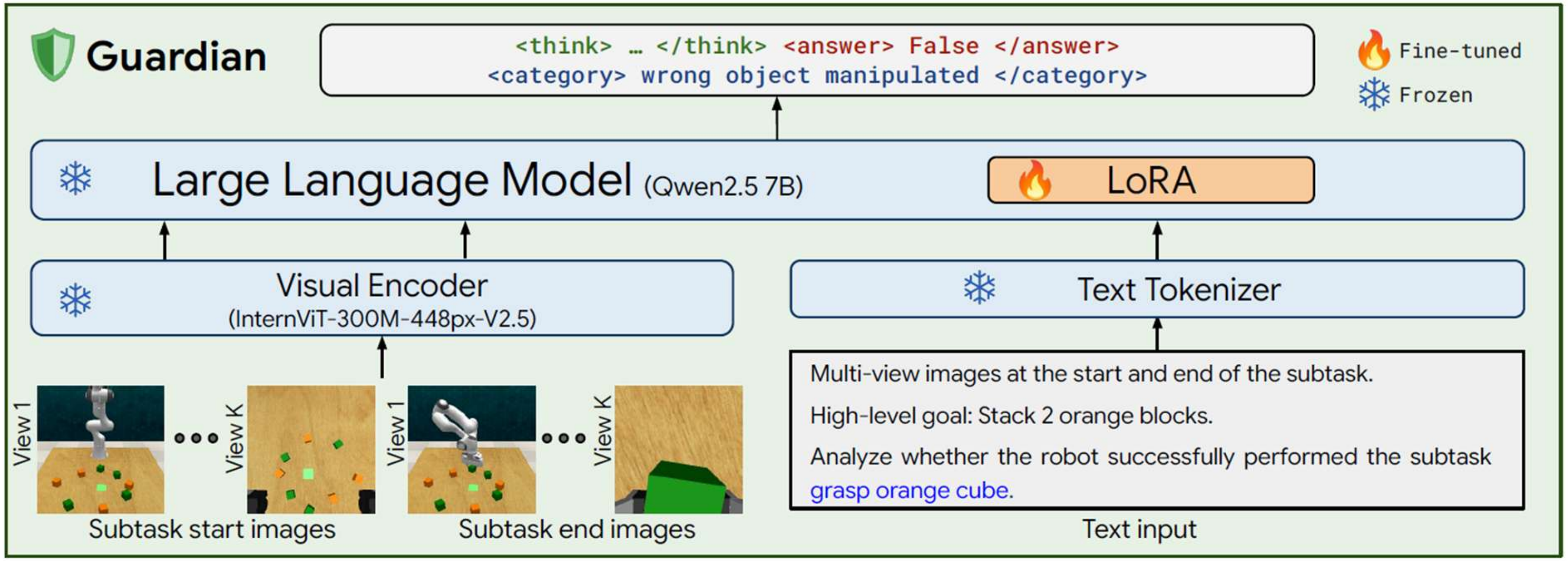
- Large-scale synthetic dataset FailCoT with RLBench + BridgeData
  - Balanced execution and planning failures
- UR5-Fail with balanced real-world failures

Guardian – data distribution



Guardian – model + training data

- InternVL3-8B model with Lora adapters
  - Input multiple views for start and end images, text prompt
  - Output: true/false + category label

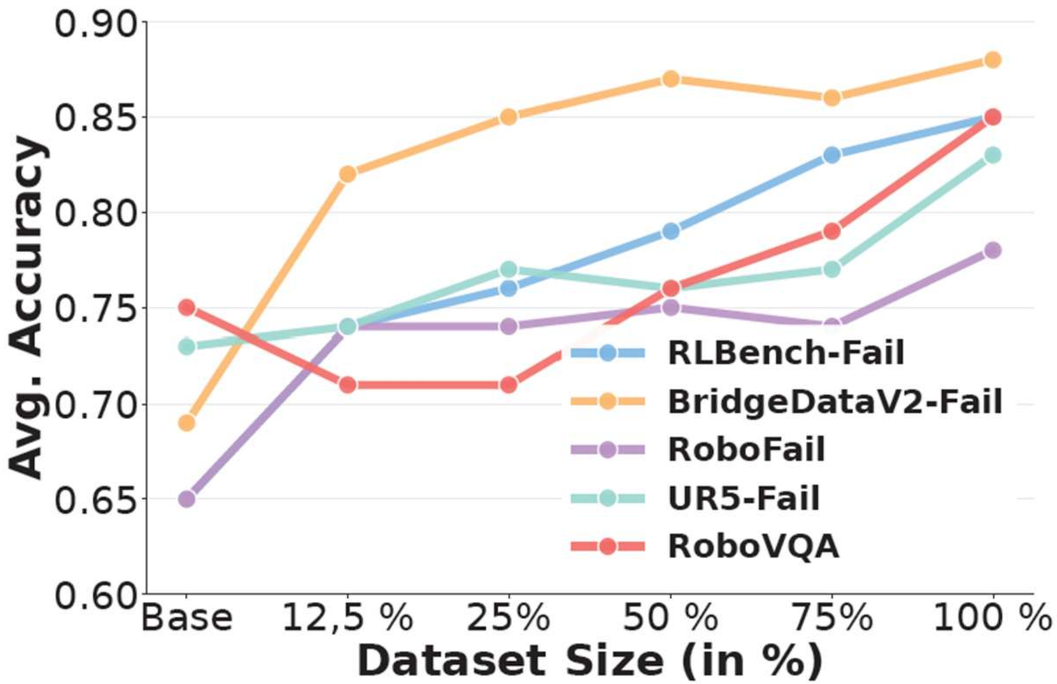


Guardian - impact of the training data

Training Data: RLBench	Training Data: BDV2	RLBench -Fail	BDV2 -Fail	Robo -Fail	UR5 -Fail	Robo -VQA
✗	✗	0.65	0.69	0.65	0.73	0.75
✓	✗	0.82	0.70	0.69	0.72	0.66
✗	✓	0.65	0.86	0.71	0.68	0.77
✓	✓	**0.85**	**0.88**	**0.78**	**0.83**	**0.85**

- Impact of Guardian training data on the accuracy averaged over planning and execution failures
- Combined data mix improves in all cases

Guardian – impact of training data size



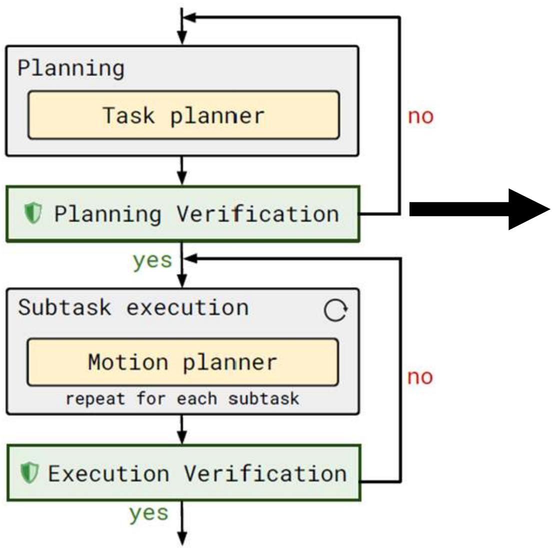
Accuracy for execution (binary classification) across benchmarks

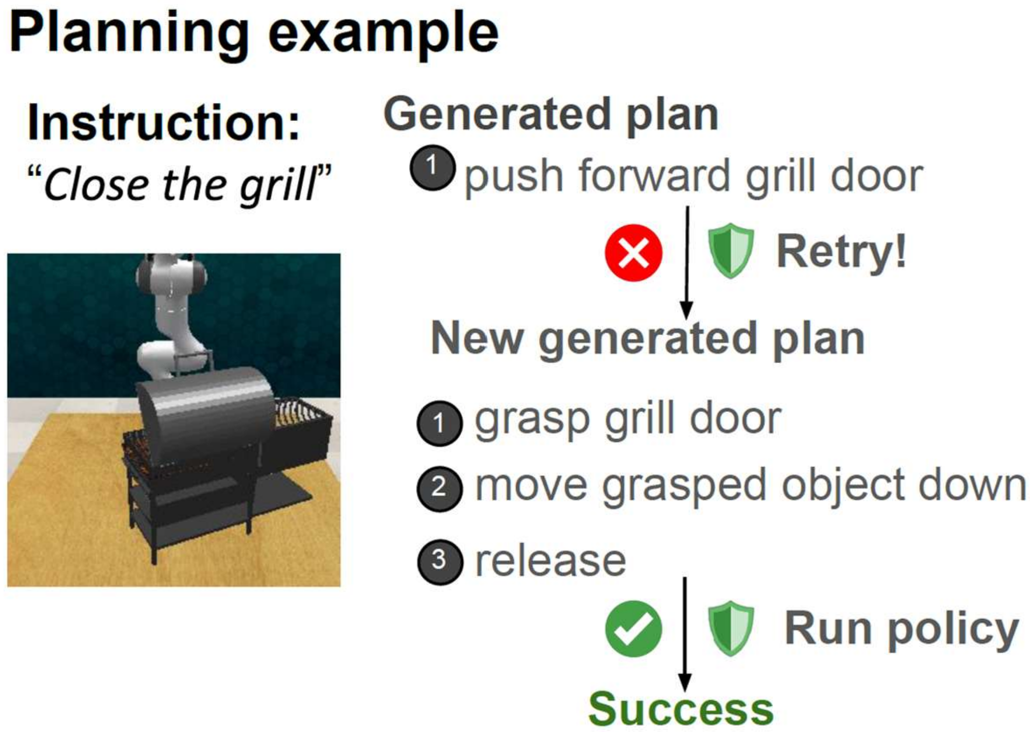
Guardian – results

Model	Trained on FailCoT	RoboFail [7] Exec	RoboFail [7] Plan	UR5-Fail Exec	UR5-Fail Plan	RVQA [27] Exec
Closed-Source VLM						
GPT-4o	✗	0.80	0.67	**0.77**	0.85	0.79
GPT-4o +Sentinel-Video-QA [10]	✗	0.80	0.63	0.76	0.62	0.66
Robotic Failure Detection VLMs						
RoboFAC-7B [18]	✗	0.25	0.05	0.54	0.02	0.52
AHA-13B* [9]	✗	0.64	-	-	-	-
I-Fail-Sense-3B [12]	✗	0.43	0.67	0.47	0.46	0.53
Cosmos-Reason2-8B [15]	✗	0.78	0.53	0.59	0.67	0.76
CLIP+MLP [46]	✓	0.42	0.43	0.51	0.51	0.52
I-Fail-Sense-3B [12]	✓	0.76	0.52	0.55	0.6	0.58
Cosmos-Reason2-8B [15]	✓	0.82	**0.70**	0.65	0.83	0.77
Guardian-8B	✓	**0.86**	**0.70**	**0.77**	**0.89**	**0.85**

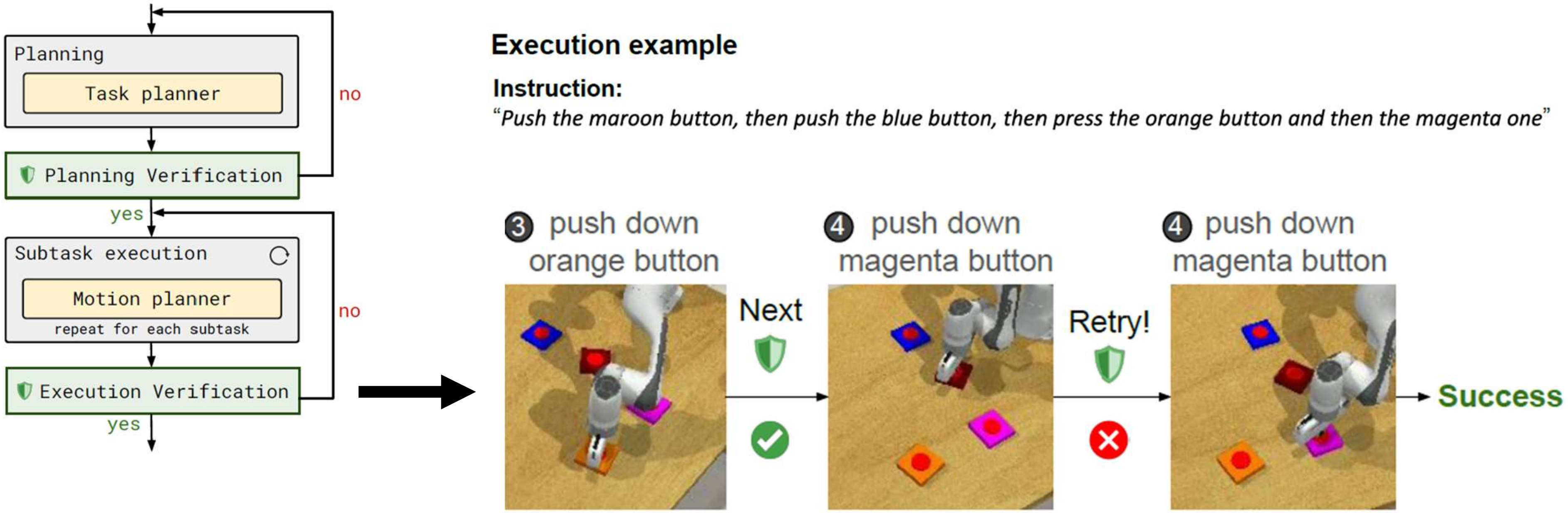
Comparison on unseen real-world benchmarks. Accuracy on execution & planning

Guardian – integration into Lotus3D++





Guardian – integration into Lotus3D++

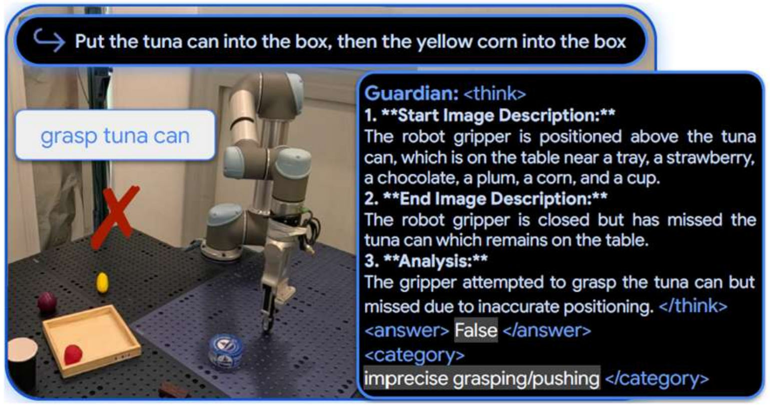


Experimental results

Verifier	Trained on FailCoT	CoT	Average	Open top drawer	Push white button	Push 4 buttons	Lift red duck	Screw maroon bulb	Slide block to yellow target	Lift black block	Close grill	Close microwave	Close bottom drawer
✗	✗	✗	$0.45_{\pm 0.03}$	$0.54_{\pm 0.03}$	$0.92_{\pm 0.01}$	$0.09_{\pm 0.03}$	$0.38_{\pm 0.05}$	$0.50_{\pm 0.04}$	$0.07_{\pm 0.03}$	$0.85_{\pm 0.02}$	$0.11_{\pm 0.02}$	$0.97_{\pm 0.01}$	$0.06_{\pm 0.02}$
✓	✗	✗	$0.49_{\pm 0.02}$	$0.64_{\pm 0.02}$	$0.93_{\pm 0.02}$	$0.12_{\pm 0.03}$	$0.40_{\pm 0.04}$	$0.48_{\pm 0.01}$	$0.15_{\pm 0.03}$	$0.86_{\pm 0.03}$	$0.12_{\pm 0.02}$	$0.98_{\pm 0.00}$	$0.19_{\pm 0.04}$
✓	✓	✗	$0.51_{\pm 0.04}$	$0.70_{\pm 0.06}$	$0.92_{\pm 0.01}$	$\mathbf{0.18}_{\pm 0.03}$	$0.40_{\pm 0.03}$	$0.48_{\pm 0.08}$	$0.20_{\pm 0.04}$	$0.86_{\pm 0.03}$	$0.16_{\pm 0.04}$	$\mathbf{1.00}_{\pm 0.00}$	$0.19_{\pm 0.05}$
✓	✓	✓	$\mathbf{0.54}_{\pm 0.03}$	$\mathbf{0.75}_{\pm 0.02}$	$\mathbf{0.96}_{\pm 0.02}$	$\mathbf{0.18}_{\pm 0.03}$	$\mathbf{0.42}_{\pm 0.03}$	$\mathbf{0.54}_{\pm 0.04}$	$\mathbf{0.31}_{\pm 0.06}$	$\mathbf{0.89}_{\pm 0.04}$	$\mathbf{0.18}_{\pm 0.03}$	$\mathbf{1.00}_{\pm 0.00}$	$\mathbf{0.20}_{\pm 0.04}$

Success rate of Lotus3D++ across unseen RLBench tasks without and with different VLM-based failure detectors. Last row = our full Guardian model

Real-world experiments

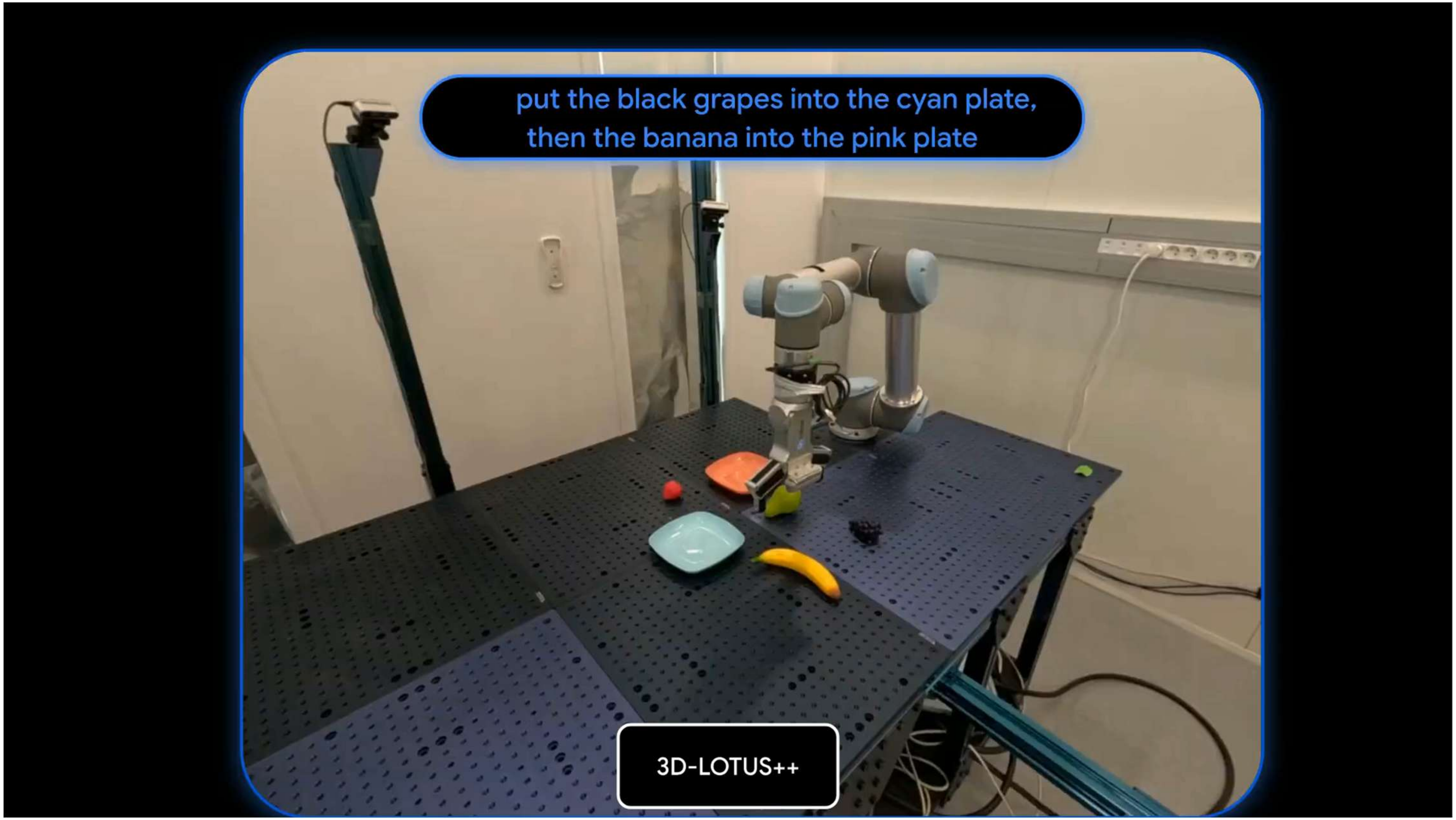


6-DoF UR5 robot with
3 RealSense cameras

Verifier	Trained on FailCoT	CoT	Put food		Arrange fruits		Stack cups	
			Norm	Pert	Norm	Pert	Norm	Pert
✗	✗	✗	15/20	4/20	10/20	3/20	9/20	2/20
✓	✗	✗	12/20	10/20	9/20	9/20	7/20	6/20
✓	✓	✗	17/20	13/20	**14/20**	**12/20**	10/20	8/20
✓	✓	✓	**18/20**	**15/20**	**14/20**	**12/20**	**12/20**	**11/20**

3D-Lotus++ results on three unseen task

Pert. : human induced perturbations


put the black grapes into the cyan plate,
then the banana into the pink plate
3D-LOTUS++

Conclusion

- Grounded VLMs and critics work well for robotics
- Need to differentiate between planning and low-level control
- More robust & precise 3D representations are important
- Improve visual / world models by real world interaction

THANK YOU

THANK YOU