

Articulate-Anything: Automatic Modeling of Articulated Objects via a Vision-Language Foundation Model

Long Le , Jason Xie, William Liang, Hung-Ju Wang, Yue Yang,
Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, Eric Eaton

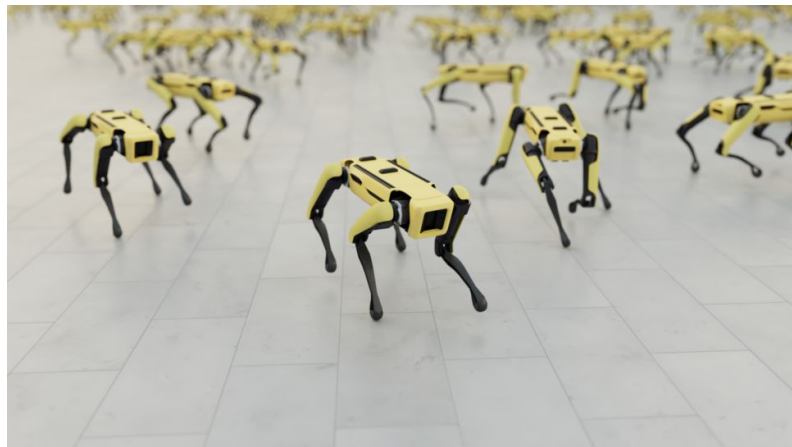
University of Pennsylvania

ICLR 2025

artikulate-anything.github.io



Robot Learning via Sim2real



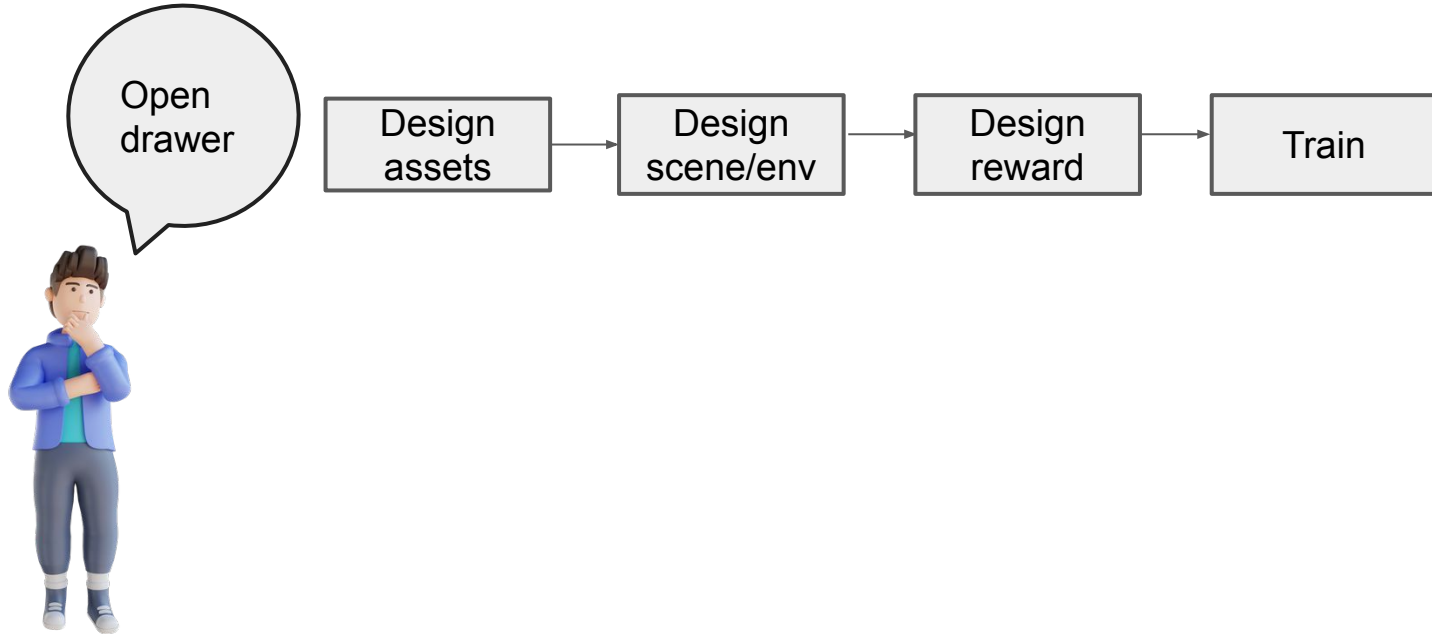
Faster

- Buy more gpus
- Faster fps
- Can cheat

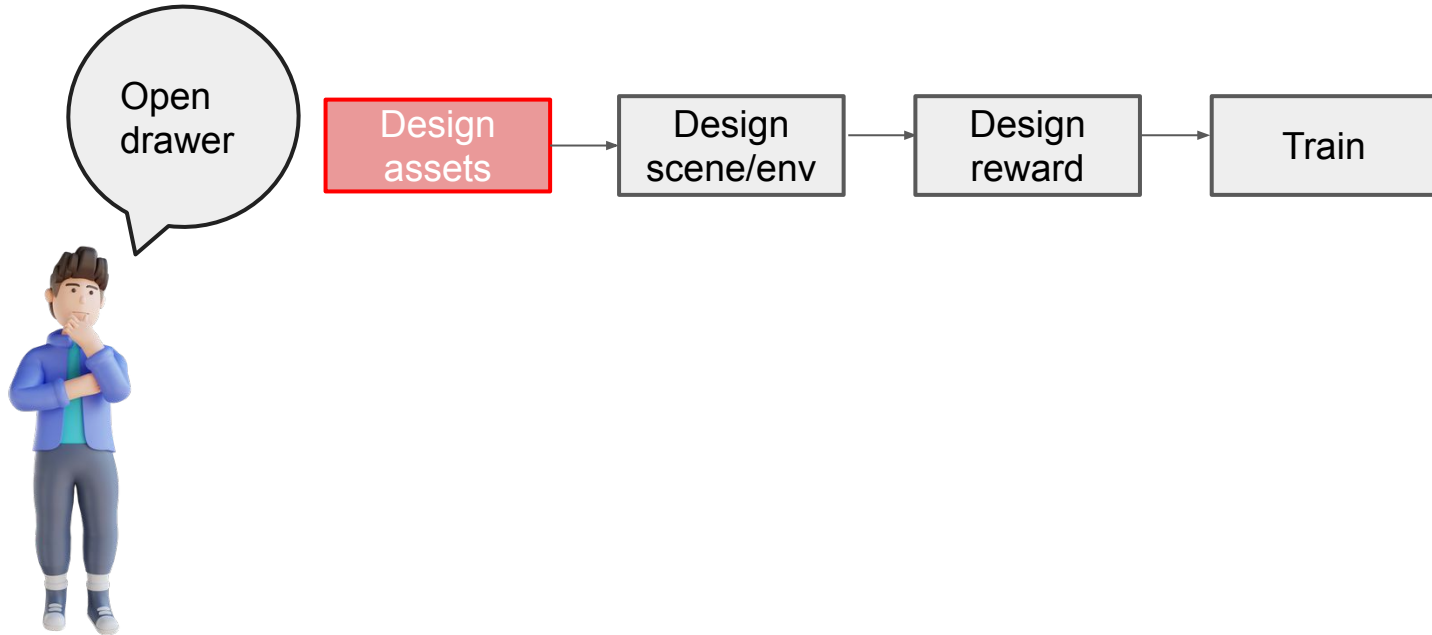


Safer

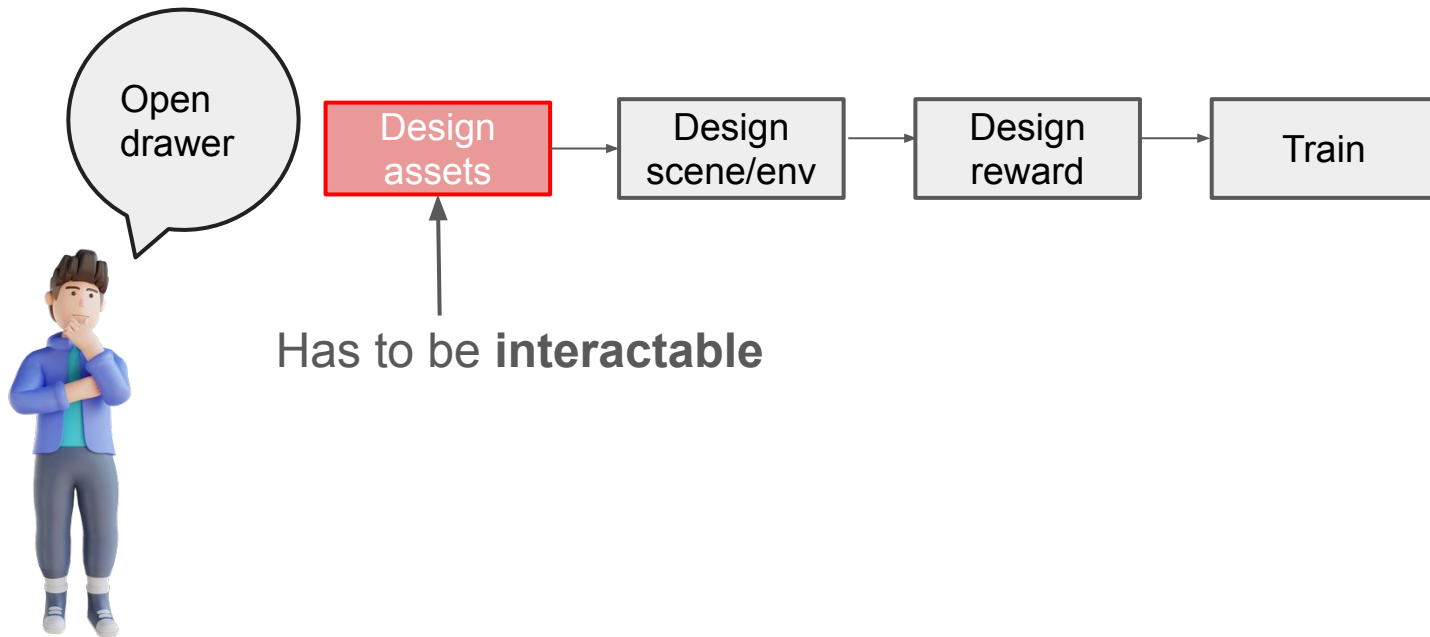
Policy Learning has manual components



Policy Learning has manual components



Policy Learning has manual components



Interactable 3D assets is sparsely available

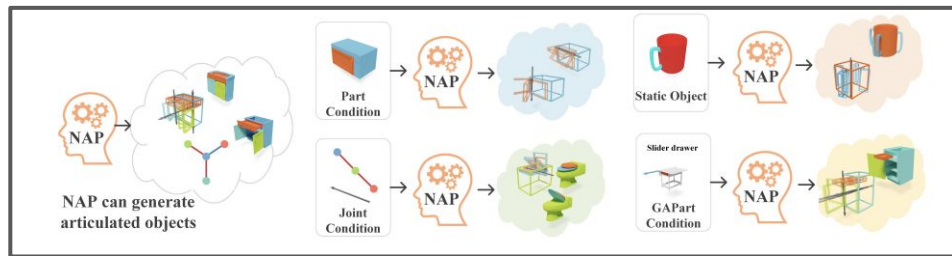


Objaverse-XL (10M objects)



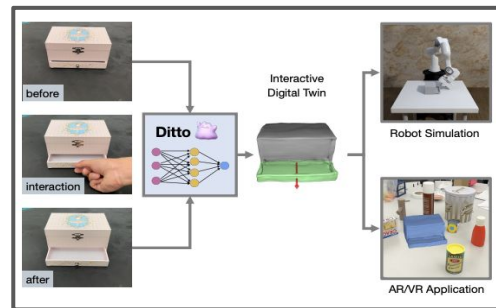
PartNet-Mobility (2k objects)

Prior Work



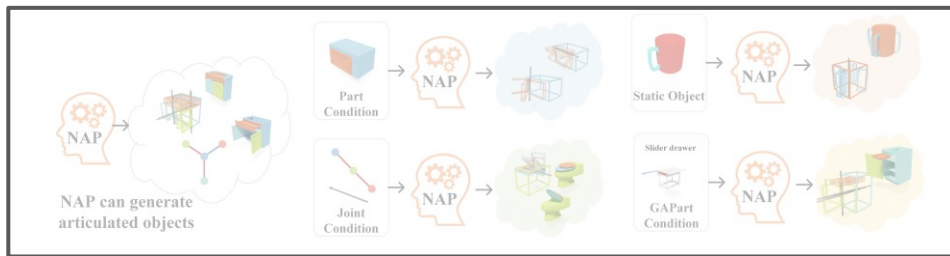
NAP: from graphs

✗ un-intuitive and difficult to obtain inputs.

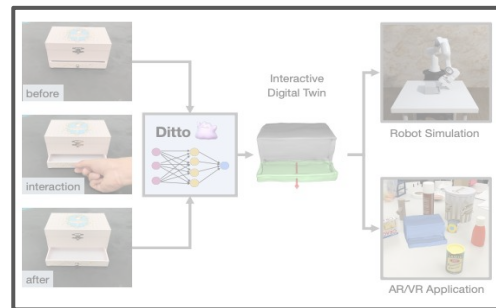


Ditto: from point-cloud

Prior Work



NAP: from graphs



Ditto: from point-cloud



urdformer: from images

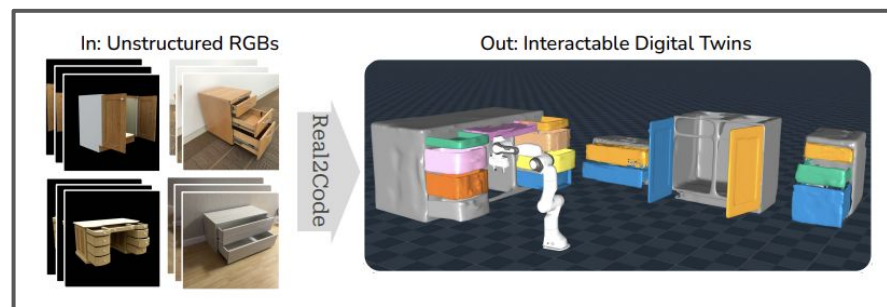
- ✗ Train a transformer from scratch.
- ✗ Little data. Hard to generalize

Prior Work

- ✗ Use impoverished modalities.
- ✗ Open-loop prediction.



urdformer: from images



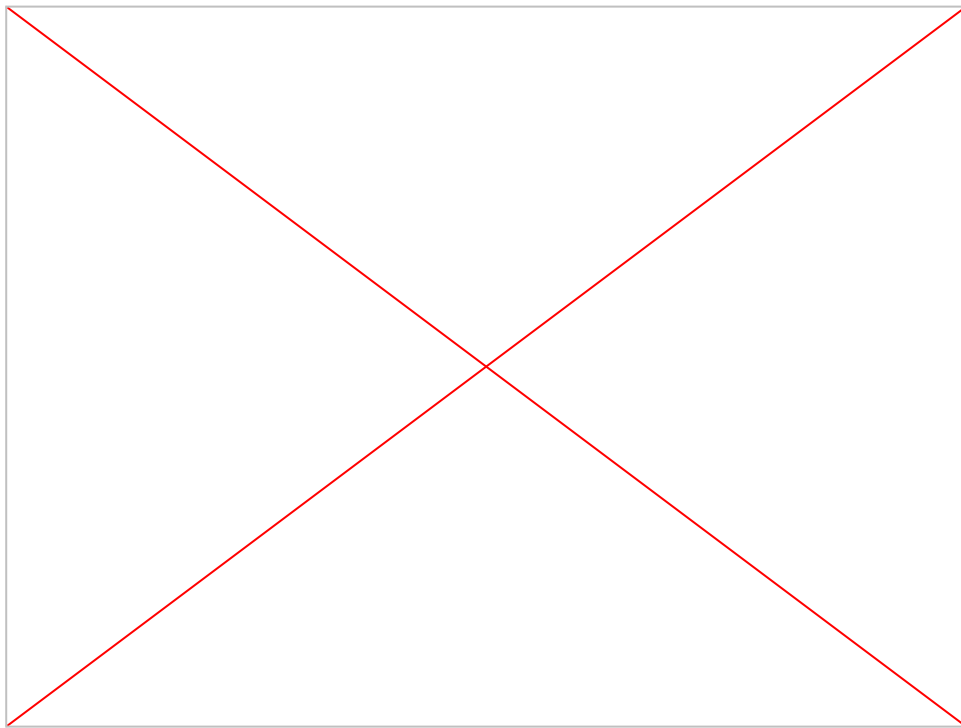
real2code: from text

Advantages of Articulate-Anything

1. **Multimodal**: can articulate any text, image, or video inputs
2. Use high-level python **abstraction** for articulation, leveraging VLM's common sense.
3. **Closed-loop** self-evaluation and improvement with actor-critic.



How does our method work?



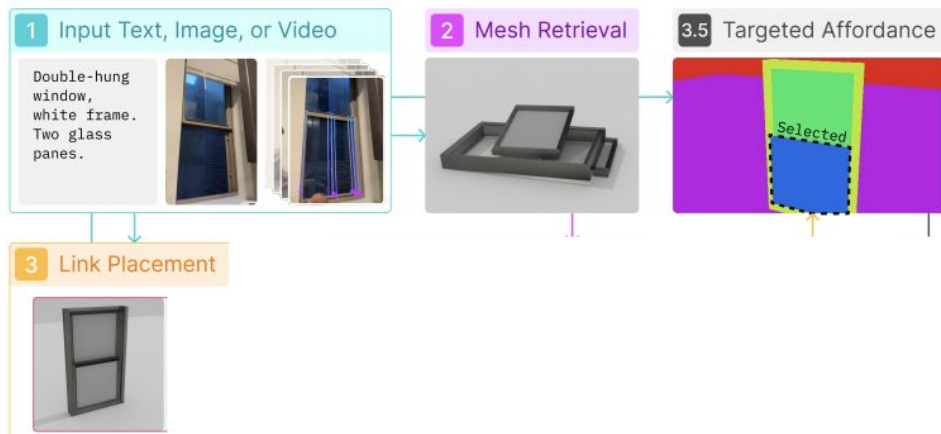
How does our method work?



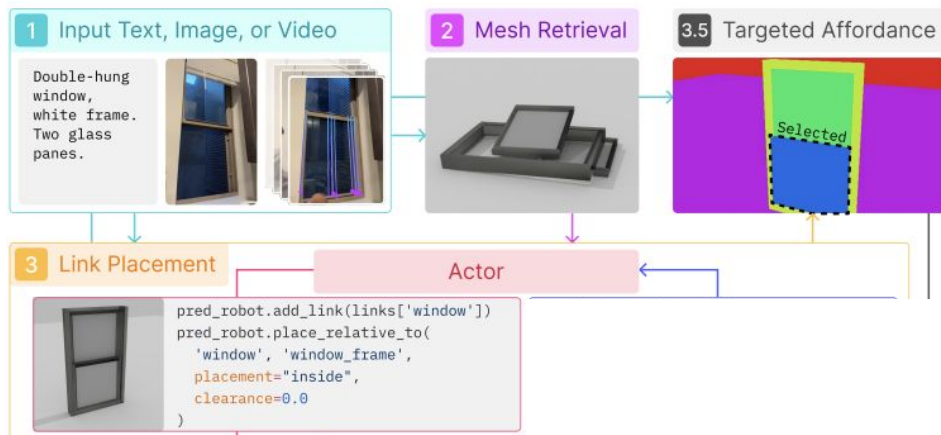
How does our method work?



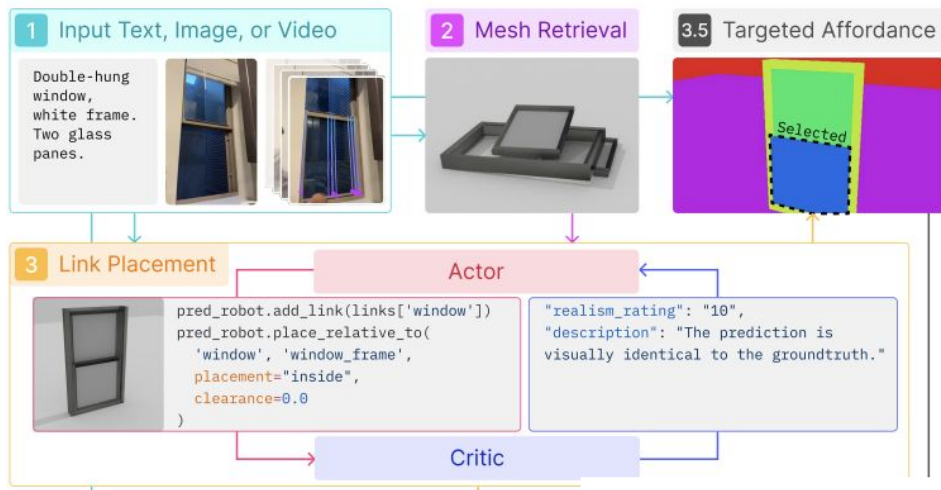
How does our method work?



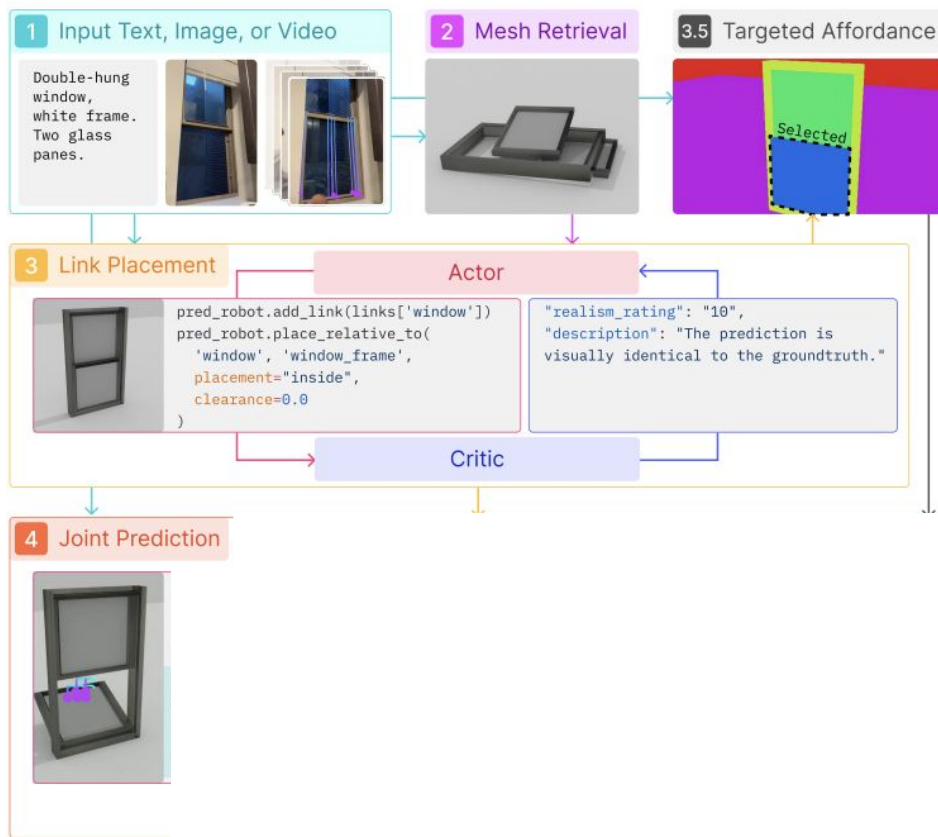
How does our method work?



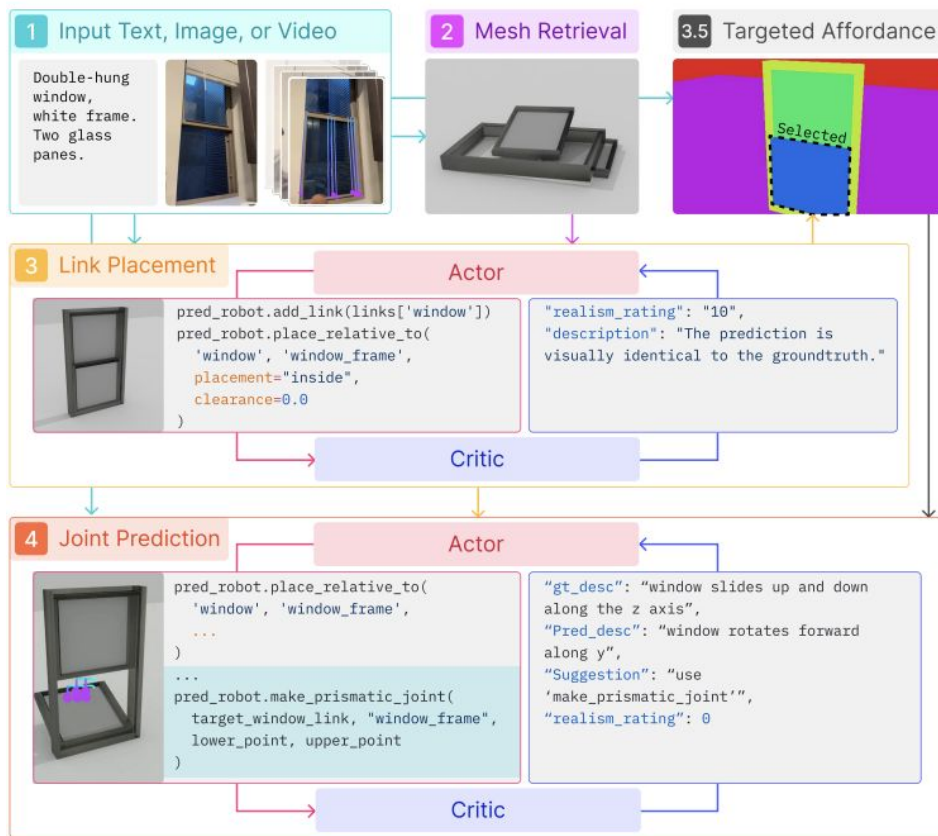
How does our method work?



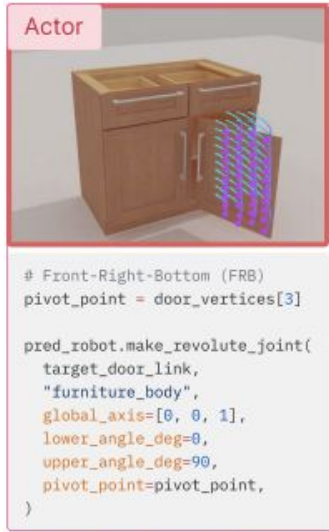
How does our method work?



How does our method work?

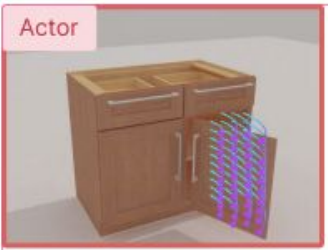


An example of iterative refinement



An example of iterative refinement

Actor



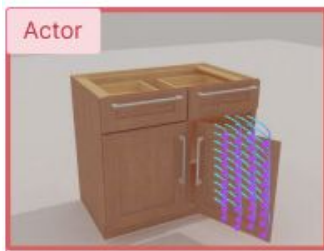

```
# Front-Right-Bottom (FRB)
pivot_point = door_vertices[3]

pred_robot.make_revolute_joint(
    target_door_link,
    "furniture_body",
    global_axis=[0, 0, 1],
    lower_angle_deg=0,
    upper_angle_deg=90,
    pivot_point=pivot_point,
)
```

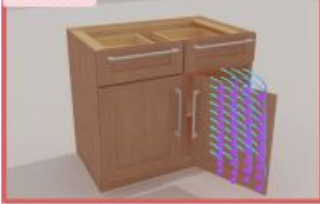

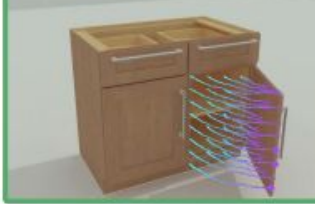
```
"improvement_suggestion": "Try
changing the pivot to the left
side of the door (e.g. Front-
Left-Bottom) to make the joint
more like the groundtruth
video.",
"realism_rating": 2
```

Critic

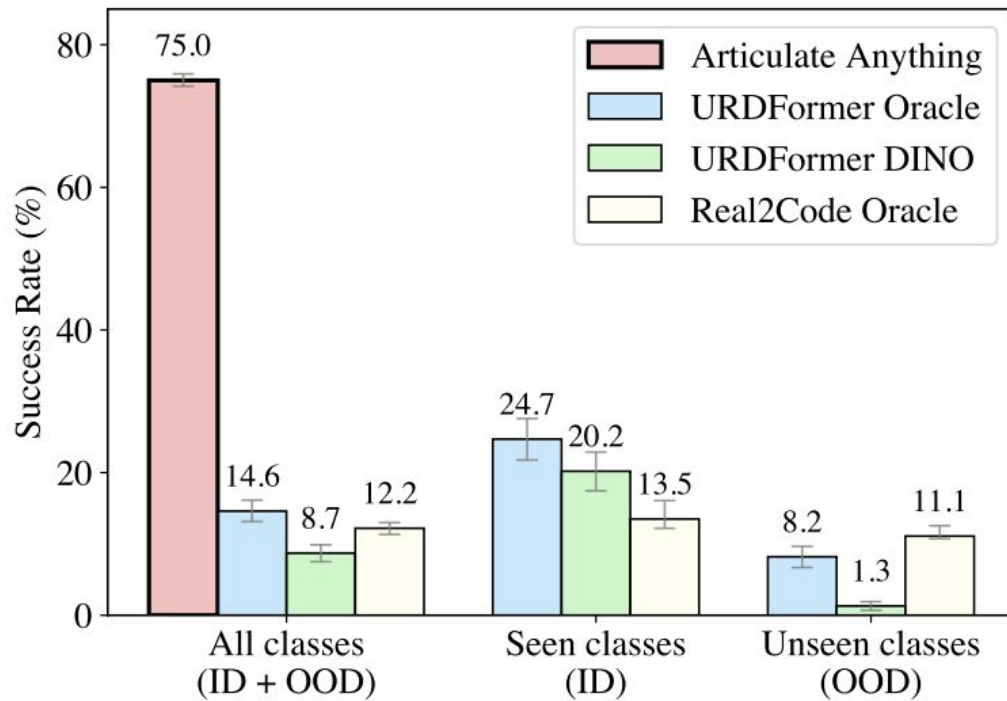
An example of iterative refinement

<p>Actor</p> 	
<pre># Front-Right-Bottom (FRB) pivot_point = door_vertices[3] pred_robot.make_revolute_joint(target_door_link, "furniture_body", global_axis=[0, 0, 1], lower_angle_deg=0, upper_angle_deg=90, pivot_point=pivot_point,)</pre>	<pre># Front-Left-Bottom (FLB) pivot_point = door_vertices[2] pred_robot.make_revolute_joint(target_door_link, "furniture_body", global_axis=[0, 0, 1], lower_angle_deg=0, upper_angle_deg=90, pivot_point=pivot_point,)</pre>
<pre>"improvement_suggestion": "Try changing the pivot to the left side of the door (e.g. Front- Left-Bottom) to make the joint more like the groundtruth video.", "realism_rating": 2</pre> <p>Critic</p>	<pre>"improvement_suggestion": "In our convention, left is negative and the limit is set from 0 to 90 so in order to open outward, the axis must be [0, 0, -1] not [0, 0, 1].", "realism_rating": 3</pre>

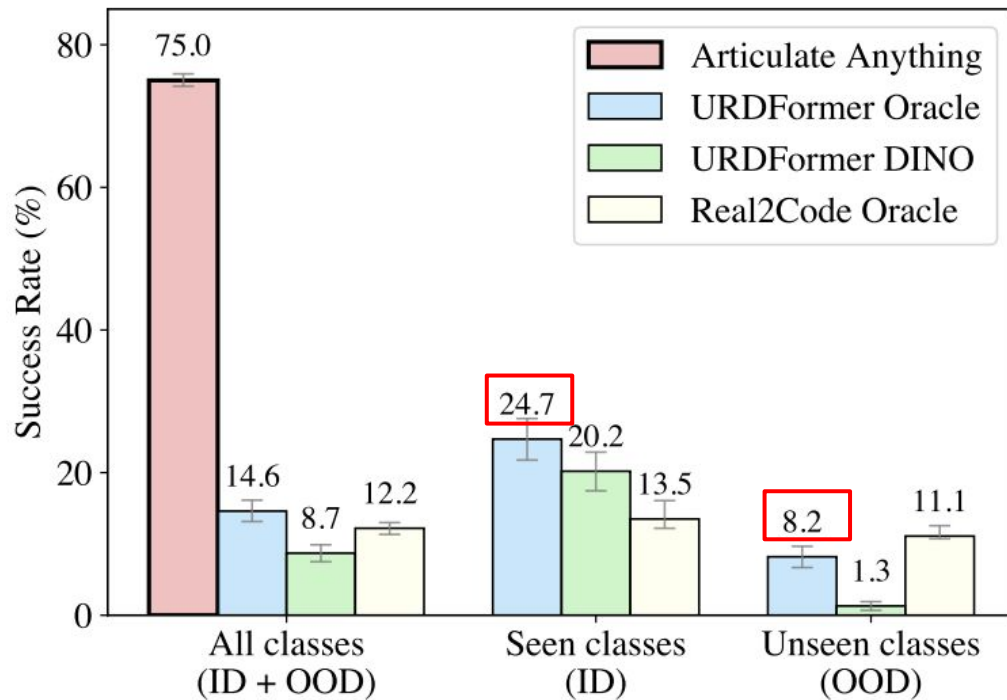
An example of iterative refinement

Actor		
		
<pre># Front-Right-Bottom (FRB) pivot_point = door_vertices[3] pred_robot.make_revolute_joint(target_door_link, "furniture_body", global_axis=[0, 0, 1], lower_angle_deg=0, upper_angle_deg=90, pivot_point=pivot_point,)</pre>	<pre># Front-Left-Bottom (FLB) pivot_point = door_vertices[2] pred_robot.make_revolute_joint(target_door_link, "furniture_body", global_axis=[0, 0, 1], lower_angle_deg=0, upper_angle_deg=90, pivot_point=pivot_point,)</pre>	<pre># Front-Left-Bottom (FLB) pivot_point = door_vertices[2] pred_robot.make_revolute_joint(target_door_link, "furniture_body", global_axis=[0, 0, -1], lower_angle_deg=0, upper_angle_deg=90, pivot_point=pivot_point,)</pre>
<pre>"improvement_suggestion": "Try changing the pivot to the left side of the door (e.g. Front- Left-Bottom) to make the joint more like the groundtruth video.", "realism_rating": 2</pre> <p>Critic</p>	<pre>"improvement_suggestion": "In our convention, left is negative and the limit is set from 0 to 90 so in order to open outward, the axis must be [0, 0, -1] not [0, 0, 1].", "realism_rating": 3</pre>	<pre>"improvement_suggestion": "None", "realism_rating": 10</pre>

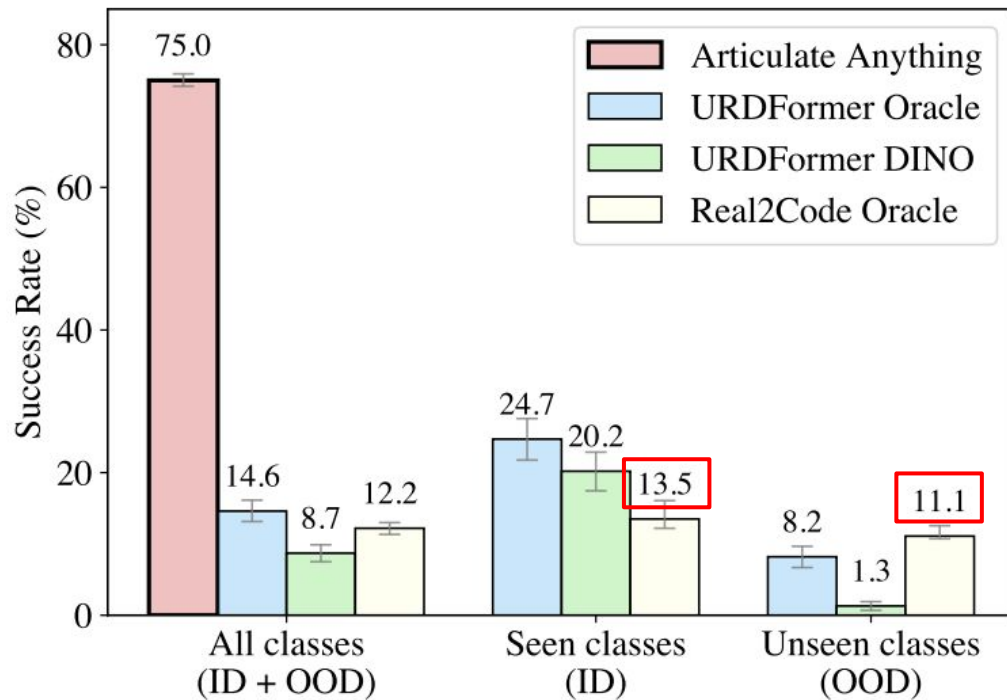
Results



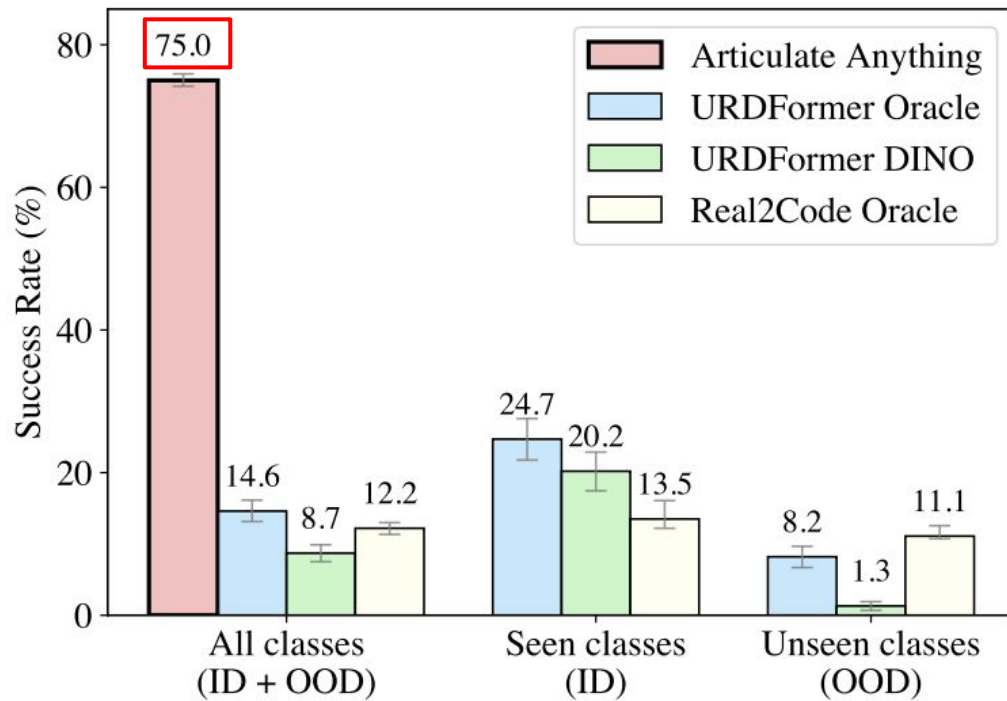
Results



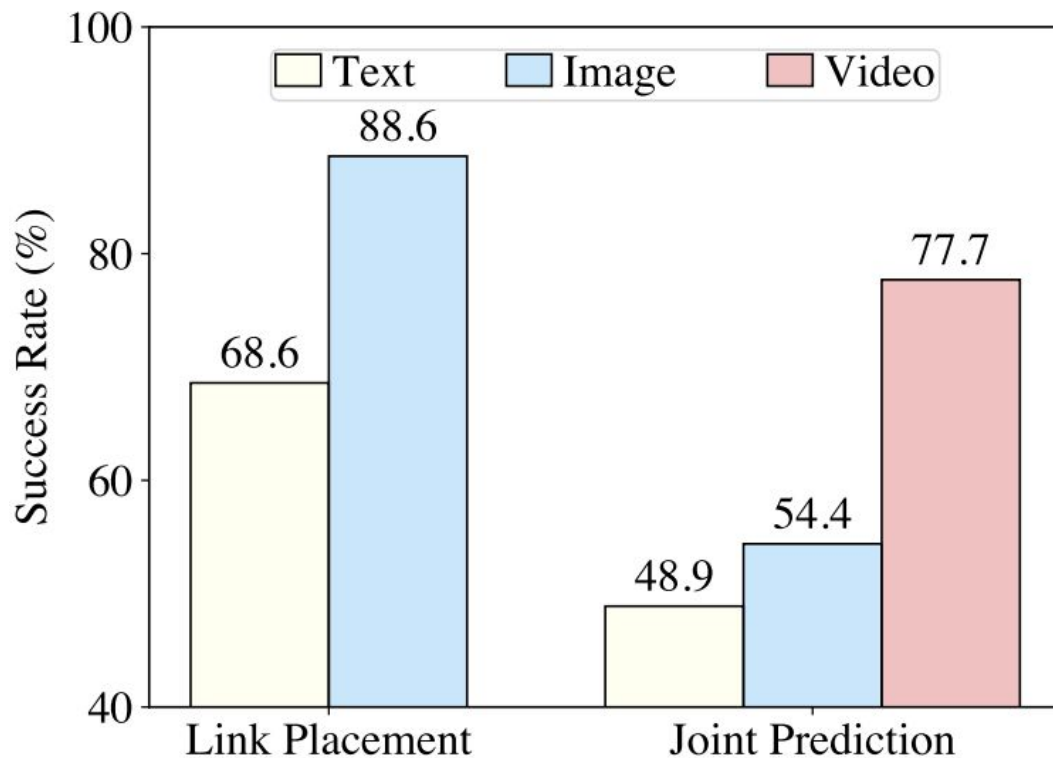
Results



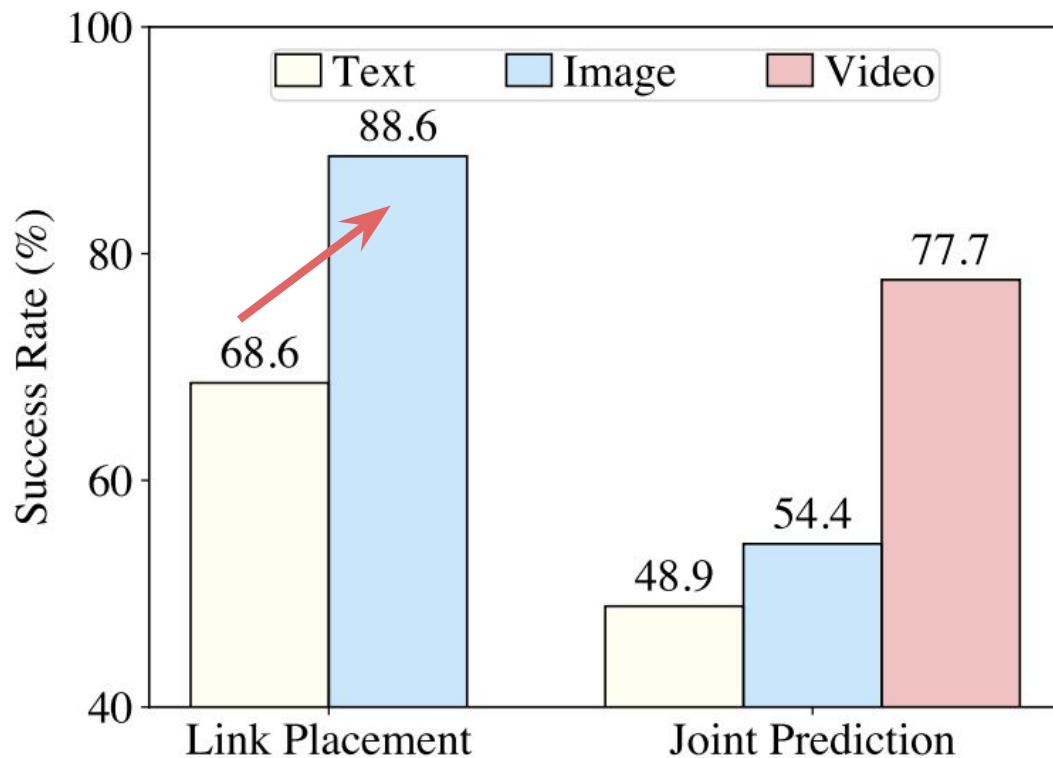
Results



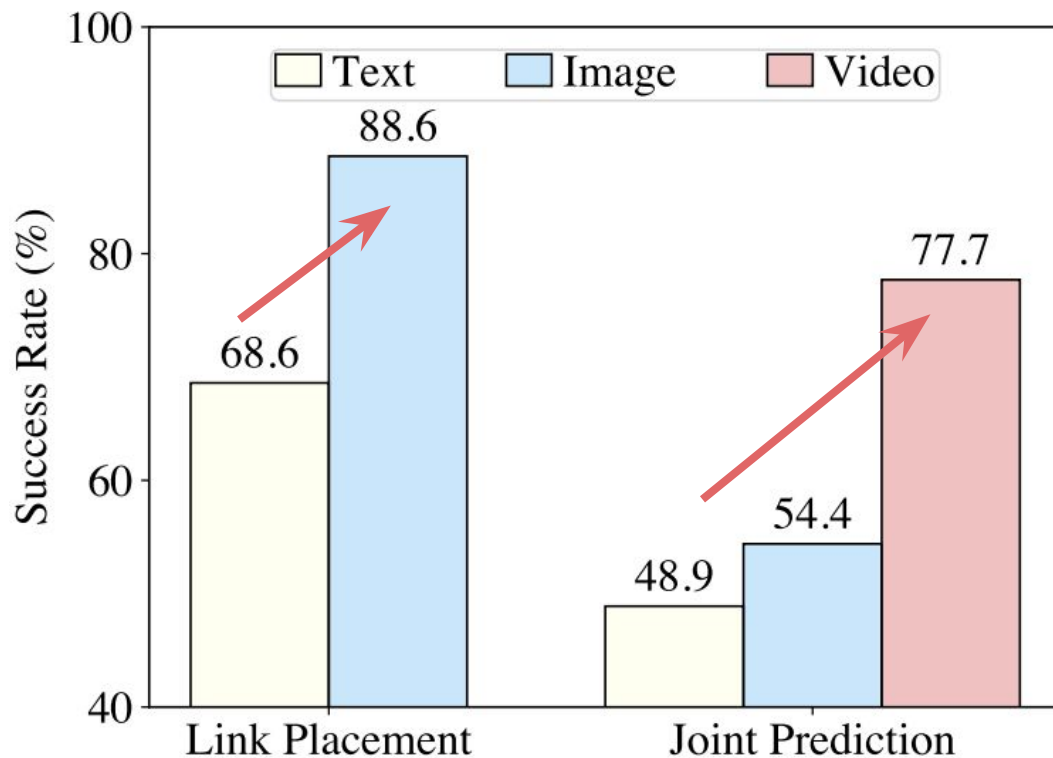
Richer inputs yield better results



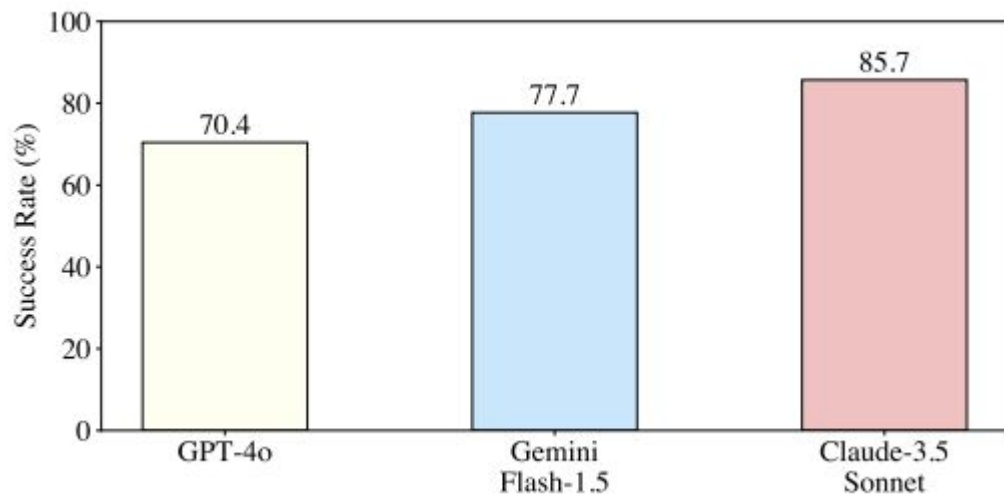
Richer inputs yield better results



Richer inputs yield better results



What matters for VLM?

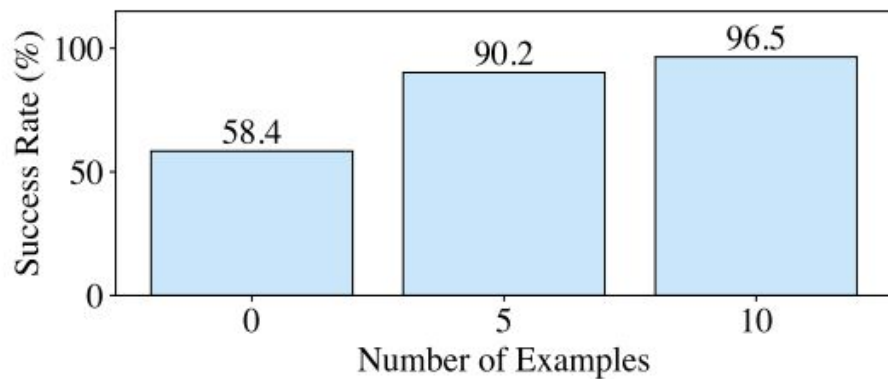


Base model? kinda....

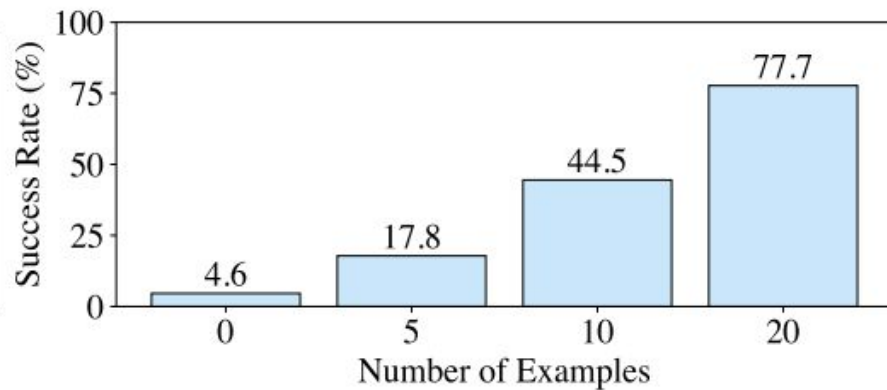
What matters for VLM?

In-context examples? definitely

Link Placement

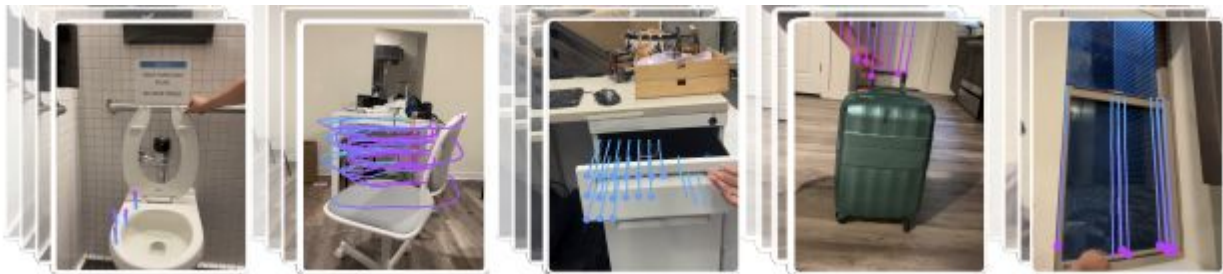


Joint Prediction



Qualitative examples

Input Videos

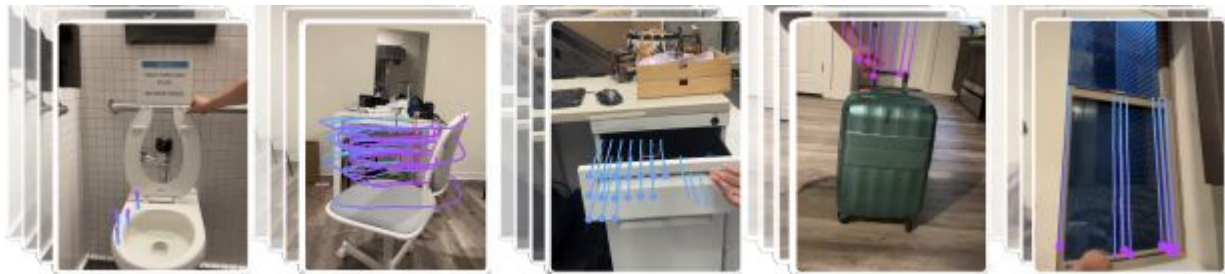


Ours w/
**Video +
Generation**



Qualitative examples

Input Videos

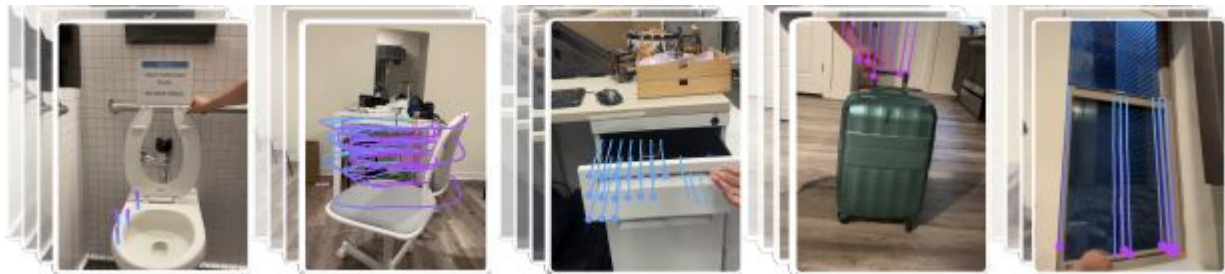


URDFormer
(use cropped images)



Qualitative examples

Input Videos



Real2Code
(use OBB texts)



Questions?

articulate-anything.github.io