

APPENDIX

In this section, we highlight the scale of our evaluations, additional findings from stress-testing RT-Sketch on sketches drawn by different individuals, and results from extending our policy to accommodate sketch+language conditioning.

A. Experiments At A Glance

Cumulatively, our results encompass the following: H1 experiments comprise 270 rollouts (6 skills x 15 trials x 3 methods), H2 comprises 40 rollouts (2 skills x 5 trials x 4 sketch types), H3 comprises 30 rollouts (15 trials x 2 methods), and H4 comprises 30 rollouts (15 trials x 2 methods). All rollouts are cumulatively evaluated across 62 labelers (split across H1-4).

B. Robustness to Input Sketches

To test whether RT-Sketch generalizes to sketches drawn by different individuals, we collect 30 *line sketches* (drawn via tracing) by 6 different annotators (whose sketches were never seen during training) on 5 trials of the *move near* scenario.

We obtain the resulting rollouts produced by RT-Sketch with these sketches as input. Across ratings, RT-Sketch achieves high spatial alignment on sketches drawn by other annotators. Notably, the performance between sketches drawn by different annotators is similar, as well as the average across annotators compared to original policy performance on our original sketches (Fig. 4).

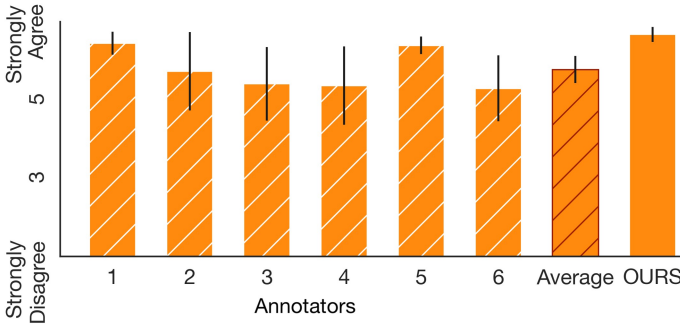


Fig. 4: Sketches Drawn by Other Annotators

C. Multimodal Goal Specification: Sketches + Language

We train a sketch-and-language conditioned model by modifying the RT-1 architecture to use FiLM along with EfficientNet layers to tokenize both visual input and language, and concatenate them at the input. In H1 experiments (Fig. 3), we evaluate all policies on the *upright* skill, where the robot must place a can or bottle from a sideways orientation initially to an upright orientation at a desired location on the table. While RT-1 typically can reorient the can/bottle properly, it struggles to place the item in the intended location on the table, as reflected in this policy’s spatial imprecision in Table I. Meanwhile, RT-Sketch struggles to reorient the can/bottle, since an imperfect sketch may fail to specify the exact desired orientation, but often places the can/bottle in the desired location. In Fig. 5, we see that while language alone (i.e. “place the can upright”) can be ambiguous in terms of spatial placement, and a sketch

alone does not encourage reorientation, we empirically see that the joint policy is better able to address the limitations of either modality alone. A similar pattern emerges for *pick drawer* (Fig. 5).

In addition to the goal alignment results reported in Fig. 3 which are based on average Likert ratings, we additionally conduct a non-parametric Mann-Whitney U (MWU) test with $\alpha = 0.05$ for H1-4 to evaluate the differences in goal alignment ratings across methods. This kind of statistical test is suitable for ordinal data and does not make specific assumptions on the normality or variance of the data distributions.

D. H1 Findings

The H1 experiments aim to evaluate how RT-Sketch compares to RT-1 and RT-Goal-Image on the standard RT-1 tabletop manipulation benchmark [8]. We conduct a MWU test under the null hypothesis that there is no difference in the goal alignment ratings from labelers across the methods. In Appendix Table III and Appendix Table IV, we report the pairs of methods for which the ratings yield a p-value of < 0.05 , rejecting the null hypothesis, along with their *U*-statistic.

TABLE III: H1: RT-1 Benchmark - Semantic Alignment

Skill	Method Pair	Stat.	p-value
Move Near			
Pick Drawer	(RT-1, RT-Goal Img)	5298.0	1.49×10^{-3}
Drawer Open	(RT-1, RT-Goal Img)	4797.0	1.22×10^{-3}
Drawer Close	(RT-1, RT-Goal Img)	4089.5	2.01×10^{-8}
Knock			
Upright	(RT-1, RT-Sketch)	16855.0	9.49×10^{-29}
	(RT-1, RT-Goal Img)	10052.0	2.80×10^{-18}
	(RT-Sketch, RT-Goal Img)	7210.5	5.62×10^{-7}

TABLE IV: H1: RT-1 Benchmark - Spatial Alignment

Skill	Method Pair	Stat.	p-value
Move Near			
Pick Drawer			
Drawer Open	(RT-1, RT-Goal Img)	4761.5	4.59×10^{-3}
Drawer Close	(RT-1, RT-Sketch)	7780.0	1.82×10^{-5}
	(RT-1, RT-Goal Img)	4869.0	3.62×10^{-10}
Knock			
Upright	(RT-1, RT-Sketch)	15085.0	1.55×10^{-14}
	(RT-1, RT-Goal Img)	10656.0	1.32×10^{-23}

We conclude that for 5 of 6 and 4 of 6 skills, the null hypothesis is confirmed for semantic and spatial alignment ratings, respectively, suggesting that there is no dropoff in performance with sketches compared to traditional modalities. We do observe that for the *upright* skill, the rating difference between RT-Sketch and RT-1 is significant, and RT-Sketch suffers a slight performance drop as re-orientation is particularly difficult to infer from a sketch alone. However, we have since addresses this challenge with the multimodal sketch and language-conditioned policy which performs reorientation better than sketches-alone and with more spatial precision than language-alone (Appendix C).

The highlighted rows above indicate when the goal alignment ratings for RT-Sketch compared to either RT-1 or RT-Goal-Image were found to be statistically significant. Notably, there are very few such findings, in alignment with H1. This

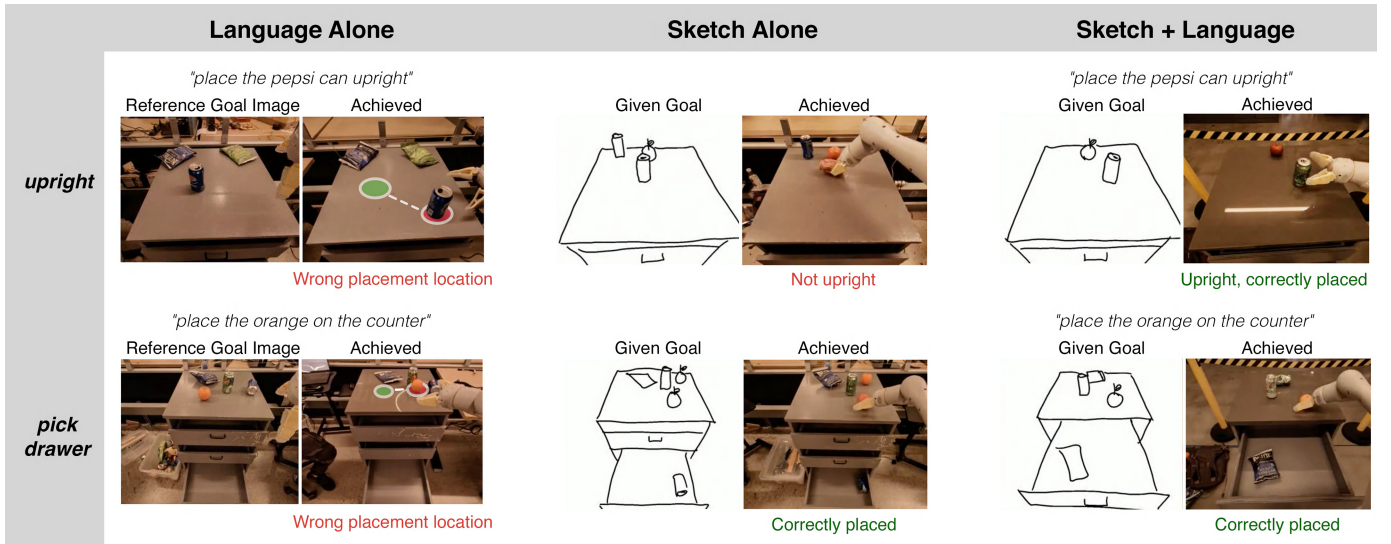


Fig. 5: **Multimodal Goal Specification: Sketch+Language**: Empirically, we find that while a language-only policy can struggle with spatial precision, and a sketch-only policy can fail to interpret intended object orientations from a sketch alone, a multimodal policy is better able to address the limitations of both.

is in accordance with what we observe Fig. 3: nearly no noticeable difference in performance between methods for most of the skills, and the slightly better performance of RT-1 compared to RT-Sketch (and the slightly better performance of RT-Sketch compared to RT-Goal-Image) for the *upright* skill.

TABLE V: **H2: Robustness to Sketch Specificity - Semantic Alignment**

Pair	Stat.	p-value
Free-Hand, Line Sketch	1059.0	9.58×10^{-12}
Free-Hand, Colored Sketch	960.0	2.54×10^{-10}
Free-Hand, Sobel Edges	1099.5	9.16×10^{-11}
Line Sketch, Colored Sketch	-	-
Line Sketch, Sobel Edges	-	-
Colored Sketch, Sobel Edges	-	-

TABLE VI: **H2: Robustness to Sketch Specificity - Spatial Alignment**

Pair	Stat.	p-value
Free-Hand, Line Sketch	478.0	5.18×10^{-17}
Free-Hand, Colored Sketch	567.5	3.49×10^{-13}
Free-Hand, Sobel Edges	629.0	3.09×10^{-14}
Line Sketch, Colored Sketch	-	-
Line Sketch, Sobel Edges	-	-
Colored Sketch, Sobel Edges	-	-

E. H2 Findings

For H2 experiments, we evaluate RT-Sketch’s robustness to the input specificity of the sketch. We find that across the 4 sketch types, the only pairings which garner statistically significant differences in ratings are free-hand sketches as compared to other types (Appendix Table V and Appendix Table VI). This is natural given the drastic perspective and geometric differences of free-hand sketches compared to those which are *traced* or derived from a transform of the goal image itself (edge detection).

However, there are notably no statistically significant pairings between line-sketches and even the most detailed type of input representation we evaluate (Sobel Edges). This suggests that RT-Sketch is indeed able to handle a range of input specificity levels, and more importantly that RT-Sketch can deal with representations that are minimal and imperfect.

TABLE VII: **H3: Visual Distractors**

Alignment	Method Pair	Stat.	p-value
Semantic	RT-Sketch, RT-Goal Img.	20622.5	4.62×10^{-8}
Spatial	RT-Sketch, RT-Goal Img.	22233.0	3.07×10^{-12}

TABLE VIII: **H4: Language Ambiguity**

Alignment	Method Pair	Stat.	p-value
Semantic	RT-Sketch, RT-1	4756.0	1.34×10^{-24}
Spatial	RT-Sketch, RT-1	3680.5	3.53×10^{-30}

F. H3 and H4 Findings

Finally, we conduct a MWU test over the semantic/spatial goal alignment ratings between RT-Sketch and RT-Goal-Image in the setting of visual distractors (H3, Appendix Table VII) as well as RT-Sketch and RT-1 in the setting of language ambiguity (H4, Appendix Table VIII). We hypothesize that RT-Sketch does indeed achieve *higher* ratings than baselines in these settings, as sketches are by nature 1) minimal, which may enable emergent robustness to distractors, and 2) agnostic to language.

We do find a statistically significant difference across semantic and spatial ratings (highlighted in orange), concluding that RT-Sketch is favorable to traditional modalities in these particular settings.

G. Summary of Mann-Whitney U Findings

In short, the additional findings from conducting more thorough MWU testing over H1-4 align very closely with what

we observe and report in Fig. 3 and suggest the merits of sketches across a range of scenarios.

Learning a policy conditioned on view-invariant sketches can be an initial step before moving to even more abstract representations like schematics or diagrams for assembly tasks. Additionally, alternative ways to condition on sketches is a powerful avenue for future work. RT-Sketch currently only considers goal observations in sketch space, but projecting all observations to a sketch-based or latent space is another underexplored but promising direction. Sketches are not without their own limitations, however, as ambiguity due to omitted details or poor quality sketches are persistent challenges. In the future, we are excited to continue exploring multimodal goal specification which can leverage the benefits of language, sketches, and other modalities to jointly resolve ambiguity from any single modality alone. This may include both end-to-end approaches that can jointly condition on multiple modalities, or hierarchical strategies that can leverage the spatial awareness of sketches and the summarization capabilities of VLMs to supplement ambiguous language with more informed descriptions derived from visual observations of a sketch. Lastly, exploring what combination of modalities humans prefer to use when providing goals, and how best they capture intent, is an important future direction not addressed in this work.

Since the main bottleneck to training a sketch-to-action policy like RT-Sketch is collecting a dataset of paired trajectories and goal sketches, we first train an image-to-sketch translation network \mathcal{T} mapping image observations o_i to sketch representations g_i , discussed in Section III. To train \mathcal{T} , we first take a pre-trained network for sketch-to-image translation [25] trained on the ContourDrawing dataset of paired images and edge-aligned sketches (Fig. 6). This dataset contains $L^{(i)} = 5$ crowdsourced sketches per image for 1000 images. By pre-training on this dataset, we hope to embed a strong prior in \mathcal{T} and accelerate learning on our much smaller dataset. Next, we finetune \mathcal{T} on a dataset of 500 manually drawn line sketches for RT-1 robot images. We visualize a few examples of our manually sketched goals in Fig. 7 under ‘Line Drawings’.

Notably, while we only train \mathcal{T} to map an image to a black-and-white line sketch \hat{g}_i , we consider various augmentations \mathcal{A} on top of generated goals to simulate sketches with varied colors, affine and perspective distortions, and levels of detail. Fig. 7 visualizes a few of these augmentations, such as automatically colorizing black-and-white sketches by superimposing a blurred version of the original RGB image, and treating an edge-detected version of the original image as a generated sketch to simulate sketches with a lot of details. We generate a dataset for training RT-Sketch by ‘sketchifying’ hind-sight relabeled goal images via \mathcal{T} and \mathcal{A} .

Although RT-Sketch is only trained on generated line sketches, colorized line sketches, edge-detected images, and goal images, we find that it is able to handle sketches of even greater diversity. This includes non-edge aligned free-hand sketches and sketches with color infills, like those shown

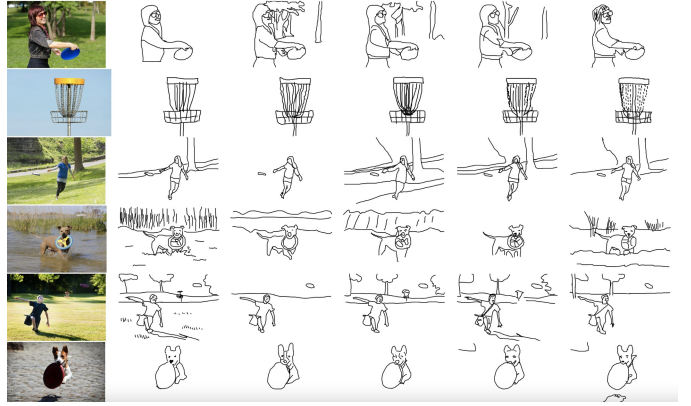


Fig. 6: **ContourDrawing Dataset:** We visualize 6 samples from the ContourDrawing Dataset from [25]. For each image, 5 separate annotators provide an edge-aligned sketch of the scene by outlining on top of the original image. As depicted, annotators are encouraged to preserve main contours of the scene, but background details or fine-grained geometric details are often omitted. Li et al. [25] then train an image-to-sketch translation network \mathcal{T} with a loss that encourages aligning with at least one of the given reference sketches.

in Fig. 7.

H. Alternate Image-to-Sketch Techniques

The choice of image-to-sketch technique we use is critical to the overall success of the RT-Sketch pipeline. We experiment with various other techniques before converging on the above approach.

Recently, two recent works, CLIPAsso [44] and CLIPAScene [43] explore methods for automatically generating a sketch from an image. These works pose sketch generation as inferring the parameters of Bezier curves representing ‘strokes’ in order to produce a generated sketch with maximal CLIP-similarity to a given input image. These methods perform a per-image optimization to generate a plausible sketch, rather than a global batched operation across many images, limiting their scalability. Additionally, they are fundamentally more concerned with producing high-quality, aesthetically pleasing sketches which capture a lot of extraneous details.

We, on the other hand, care about producing a minimal but reasonable-quality sketch. The second technique we explore is trying the pre-trained Photosketching GAN [25] on internet data of paired images and sketches. However, this model output does not capture object details well, likely due to not having been trained on robot observations, and contains irrelevant sketch details. Finally, by finetuning this PhotoSketching GAN on our own data, the outputs are much closer to real, hand-drawn human sketches that capture salient object details as minimally as possible. We visualize these differences in Fig. 8.

To further interpret RT-Sketch’s performance, we provide visualizations of the precision metrics and experimental roll-outs. In Fig. 9, we visualize the degree of alignment RT-Sketch achieves, as quantified by the pixelwise distance of object centroids in achieved vs. given goal images. In Fig. 10, Fig. 11, Fig. 12, and Fig. 14, we visualize each policy’s behavior for H1, H2, H3 and H4, respectively. Fig. 13 visualizes the four

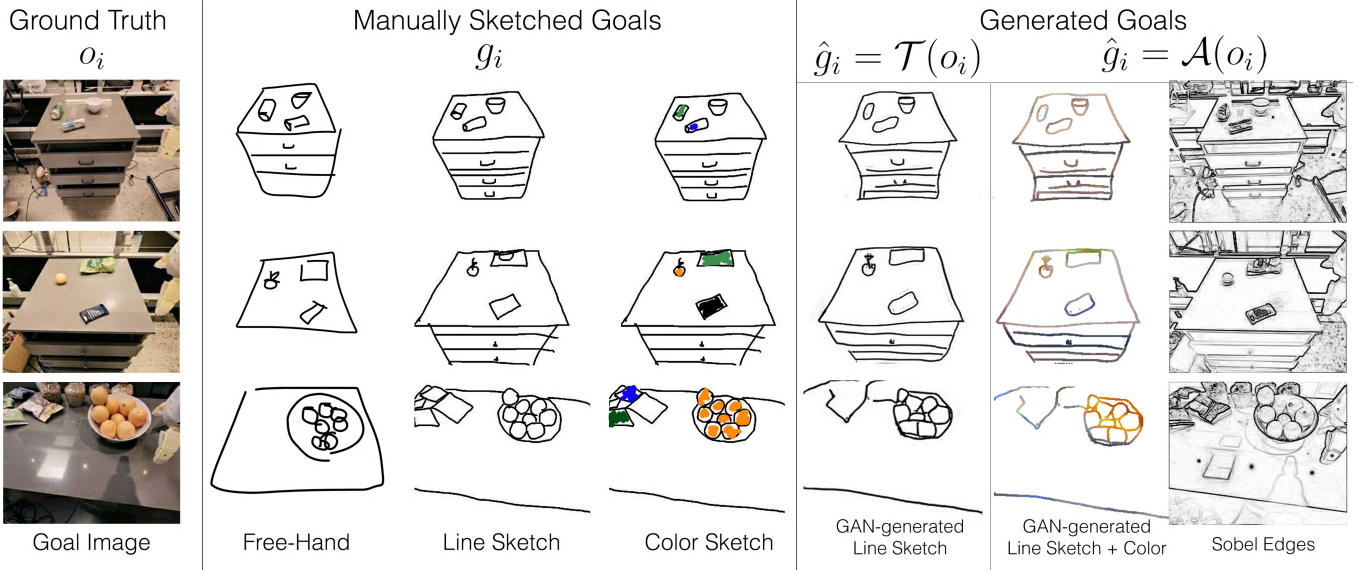


Fig. 7: **Visual Goal Diversity**: RT-Sketch is capable of handling a variety of visual goals at both train and test time. RT-Sketch is trained on generated and augmented images like those shown on the right below ‘Generated Goals’. But it can also interpret free-hand, line sketches, and colored sketches at test time such as those on the left below ‘Manually Sketched Goals’.

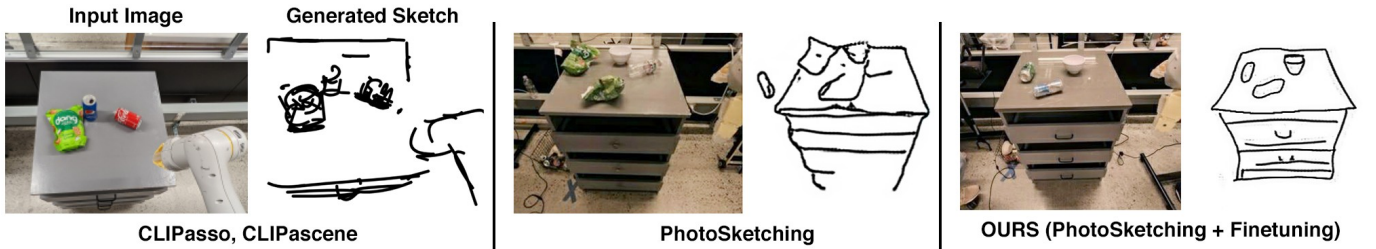


Fig. 8: **Alternate Image-to-Sketch Techniques**

tiers of difficulty in language ambiguity that we analyze for **H4**.

While RT-Sketch is performant at several manipulation benchmark skills, capable of handling different levels of sketch detail, robust to visual distractors, and unaffected by ambiguous language, it is not without failures and limitations.

In Fig. 16, we visualize the failure modes of RT-Sketch. One failure mode we see with RT-Sketch is occasionally re-trying excessively, as a result of trying to align the scene as closely as possible. For instance, in the top row, Rollout Image 3, the scene is already well-aligned, but RT-Sketch keeps shifting the chip bag which causes some misalignment in terms of the chip bag orientation. Still, this kind of failure is most common with RT-Goal-Image (Table I), and is not nearly as frequent for RT-Sketch. We posit that this could be due to the fact that sketches enable high-level spatial reasoning without over-attending to pixel-level details.

One consequence of spatial reasoning at such a high level, though, is an occasional lack of precision. This is noticeable when RT-Sketch orients items incorrectly (second row) or positions them slightly off, possibly disturbing other items in the scene (third row). This may be due to the fact that

sketches are inherently imperfect, which makes it difficult to reason with such high precision.

Finally, we see that RT-Sketch occasionally manipulates the wrong object (rows 4 and 5). Interestingly, we see that a fairly frequent pattern of behavior is to manipulate the wrong object (orange in row 4) to the right target location (near green can in row 4). This may be due to the fact that the sketch-generating GAN has occasionally hallucinated artifacts or geometric details missing from the actual objects. Having been trained on some examples like these, RT-Sketch can mistakenly perceive the wrong object to be aligned with an object drawn in the sketch. However, the sketch still indicates the relative desired spatial positioning of objects in the scene, so in this case RT-Sketch still attempts to align the incorrect object with the proper place.

Finally, the least frequent failure mode is manipulating the wrong object to the wrong target location (i.e. opening the wrong drawer handle). This is most frequent when the input is a free-hand sketch, and could be mitigated by increasing sketch detail (Table II).

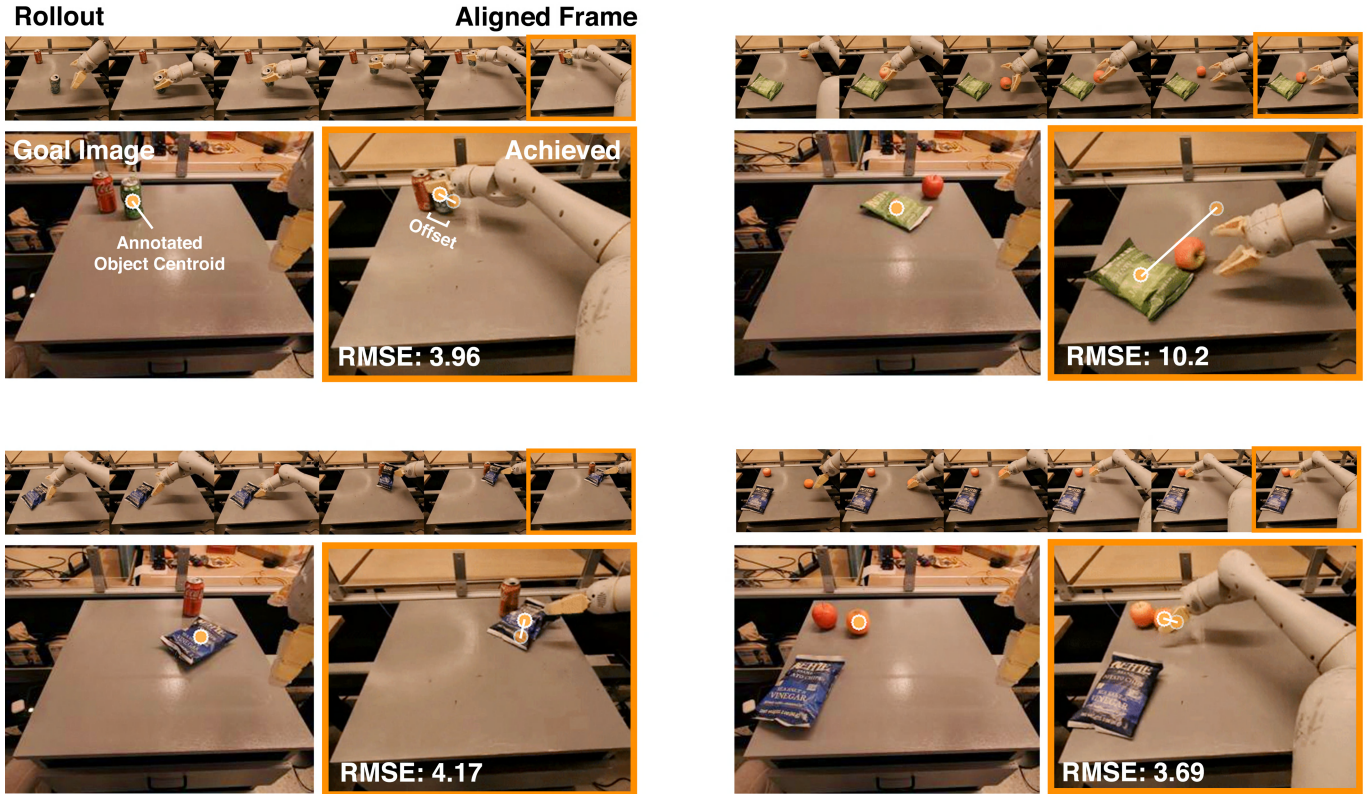


Fig. 9: **Spatial Precision Visualization:** We visualize four trials of RT-Sketch on the Move Near skill, along with the measured spatial precision in terms of RMSE. To evaluate spatial precision, we have a human annotator annotate the frame that is visually most aligned, and then keypoints for the object that was moved in this frame and in the provided reference goal image. For each of the four trials, we visualize the rollout frames until alignment is achieved, along with the labeled object centroids and the offset in achieved vs. desired positions. The upper right example shows a failure of RT-Sketch in which the apple is moved instead of the chip bag, incurring a high RMSE. These visualizations are intended to better contextualize the numbers from [Table I](#).

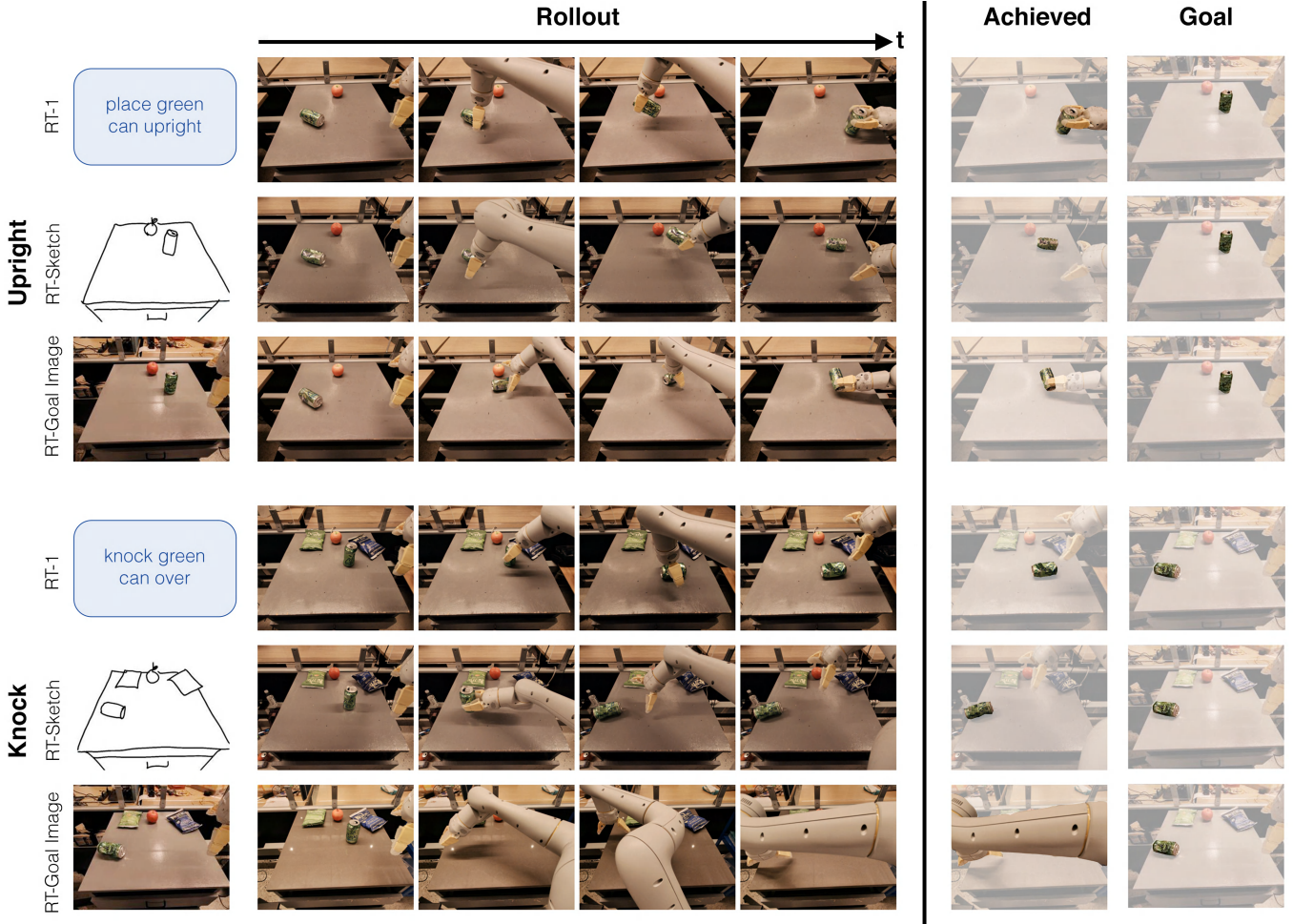


Fig. 10: **H1 Rollout Visualization:** We visualize the performance of RT-1, RT-Sketch, and RT-Goal-Image on two skills from the RT-1 benchmark (*upright* and *knock*). For each skill, we visualize the goal provided as input to each policy, along with the policy rollout. We see that for both skills, RT-1 obeys the semantic task at hand by successfully placing the can upright or sideways, as intended. Meanwhile, RT-Sketch and RT-Goal-Image struggle with orienting the can upright, but successfully knock it sideways. Interestingly, both RT-Sketch and RT-Goal-Image are able to place the can in the desired location (disregarding can orientation) whereas RT-1 does not pay attention to where in the scene the can should be placed. This is indicated by the discrepancy in position of the can in the achieved versus goal images on the right. This trend best explains the anomalous performance of RT-Sketch and RT-Goal-Image in perceived Likert ratings for the upright task (Fig. 3), but validates their comparably higher spatial precision compared to RT-1 across all benchmark skills (Table I).

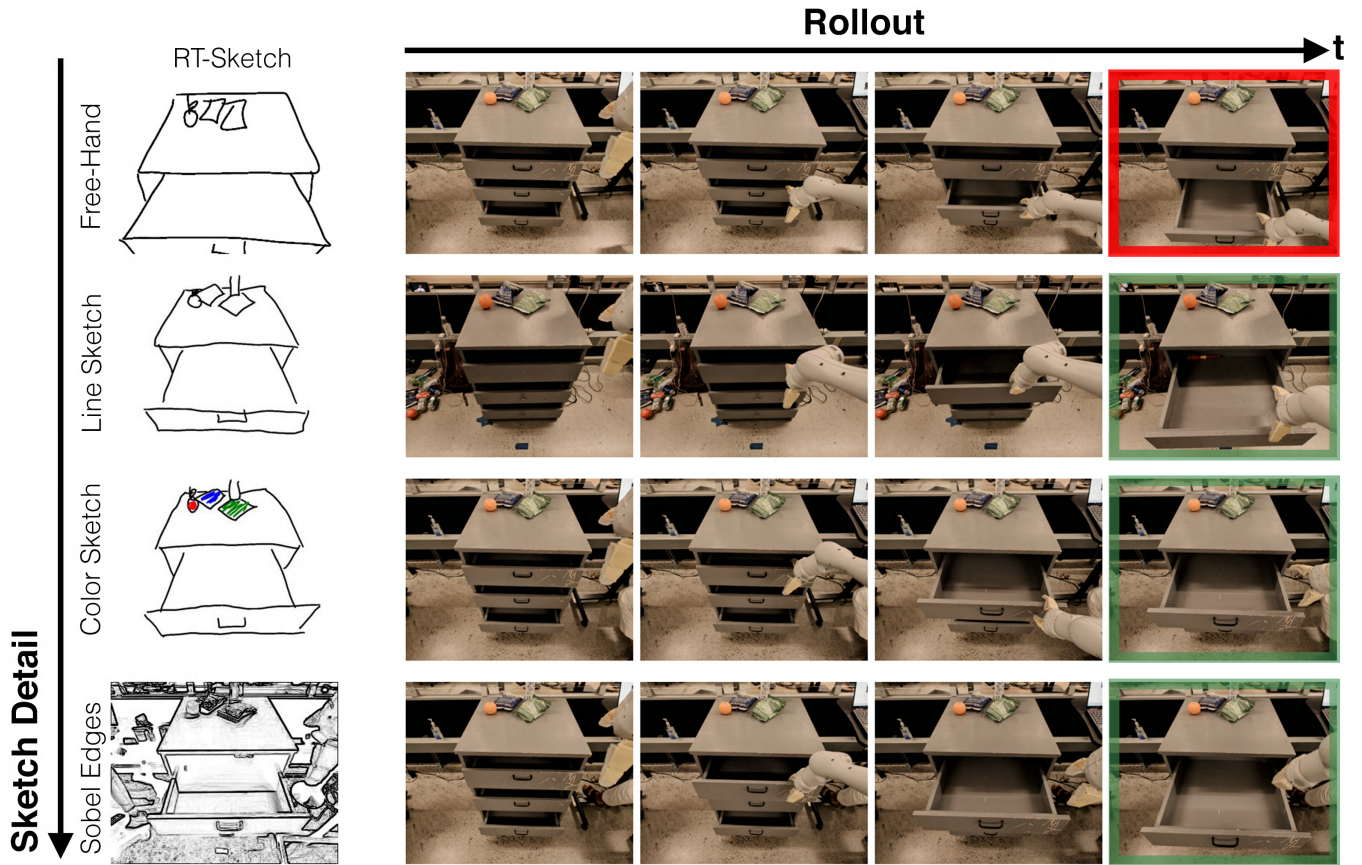


Fig. 11: **H2 Rollout Visualization:** For the *open drawer* skill, we visualize four separate rollouts of RT-Sketch operating from different input types. Free-hand sketches are drawn without outlining over the original image, such that they can contain marked perspective differences, partially obscured objects (drawer handle), and roughly drawn object outlines. Line sketches are drawn on top of the original image using the sketching interface we present in Appendix Fig. 17. Color sketches merely add color infills to the previous modality, and Sobel Edges represent an upper bound in terms of unrealistic sketch detail. We see that RT-Sketch is able to successfully open the correct drawer for any sketch input except the free-hand sketch, without a noticeable performance gain or drop. For the free-hand sketch, RT-Sketch still recognizes the need for opening a drawer, but the differences in sketch perspective and scale can occasionally cause the policy to attend to the wrong drawer, as depicted.



Fig. 12: **H3 Rollout Visualization:** We visualize qualitative rollouts for RT-Sketch and RT-Goal-Image for 3 separate trials of the *move near* skill subject to distractor objects. In Column 2, we highlight the relevant non-distractor objects that the policy must manipulate in order to achieve the given goal. In Trial 1, we see that RT-Sketch successfully attends to the relevant objects and moves the blue chip bag near the coke can. Meanwhile, RT-Goal-Image is confused about which blue object to manipulate, and picks up the blue pepsi can instead of the blue chip bag (A). In Trial 2, RT-Sketch successfully moves an apple near the fruit on the left. A benefit of sketches is their ability to capture instance multimodality, as any of the fruits highlighted in Column 2 are valid options to move, whereas this does not hold for an overspecified goal image. RT-Goal-Image erroneously picks up the green chip bag (B) instead of a fruit. Finally, Trial 3 shows a failure for both policies. While RT-Sketch successfully infers that the green can must be moved near the red one, it accidentally knocks over the red can (C) in the process. Meanwhile, RT-Goal-Image prematurely drops the green can and instead tries to pick the green chip bag (D).

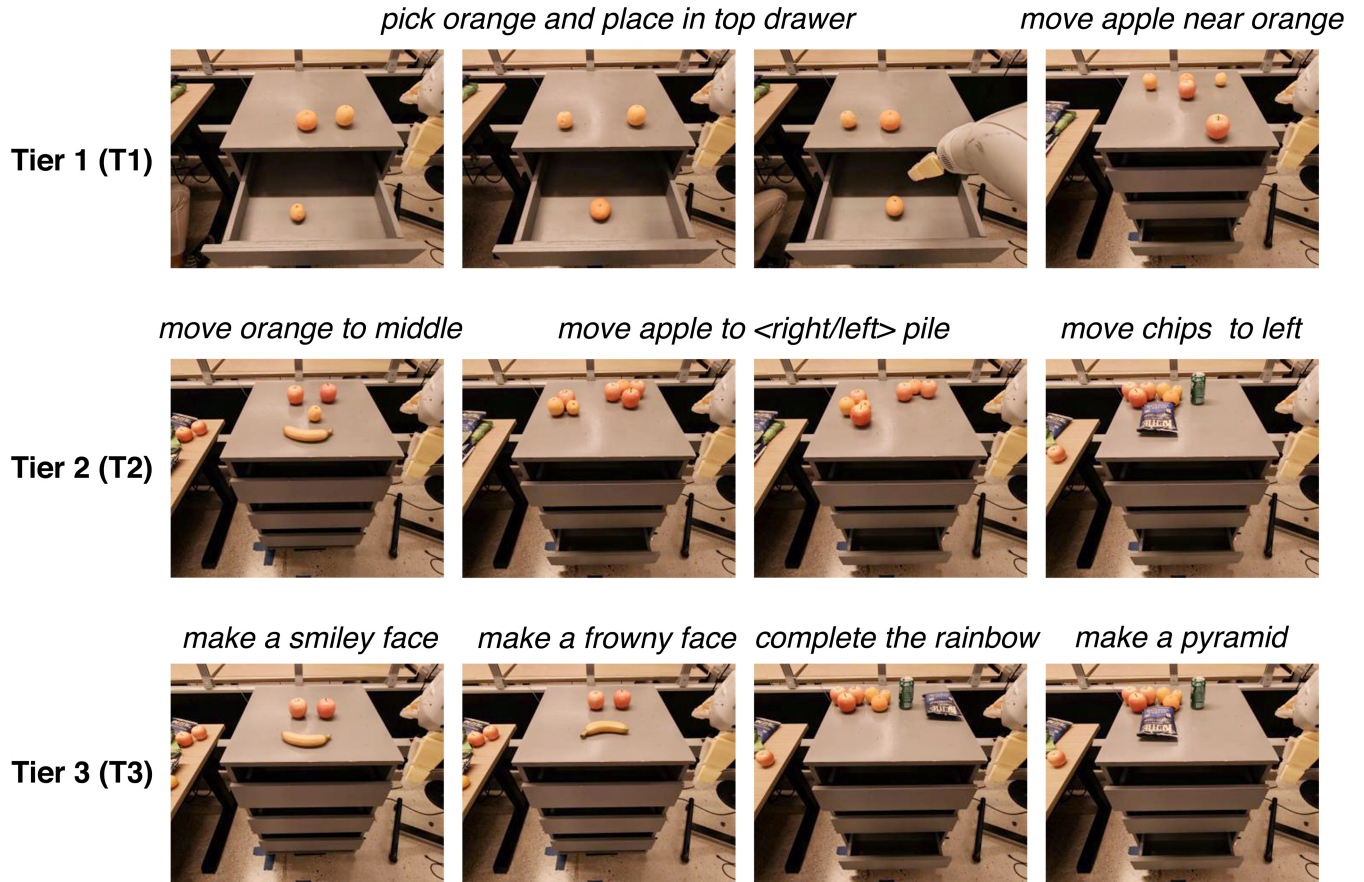


Fig. 13: **H4 Tiers of Difficulty**: To test **H4**, we consider language instructions that are either ambiguous due the presence of multiple similar object instances (**T1**), are somewhat out-of-distribution for RT-1 (**T2**), or are far out-of-distribution and difficult to specify concretely without lengthier descriptions (**T3**). Each image represents the ground truth goal image paired with the task description.

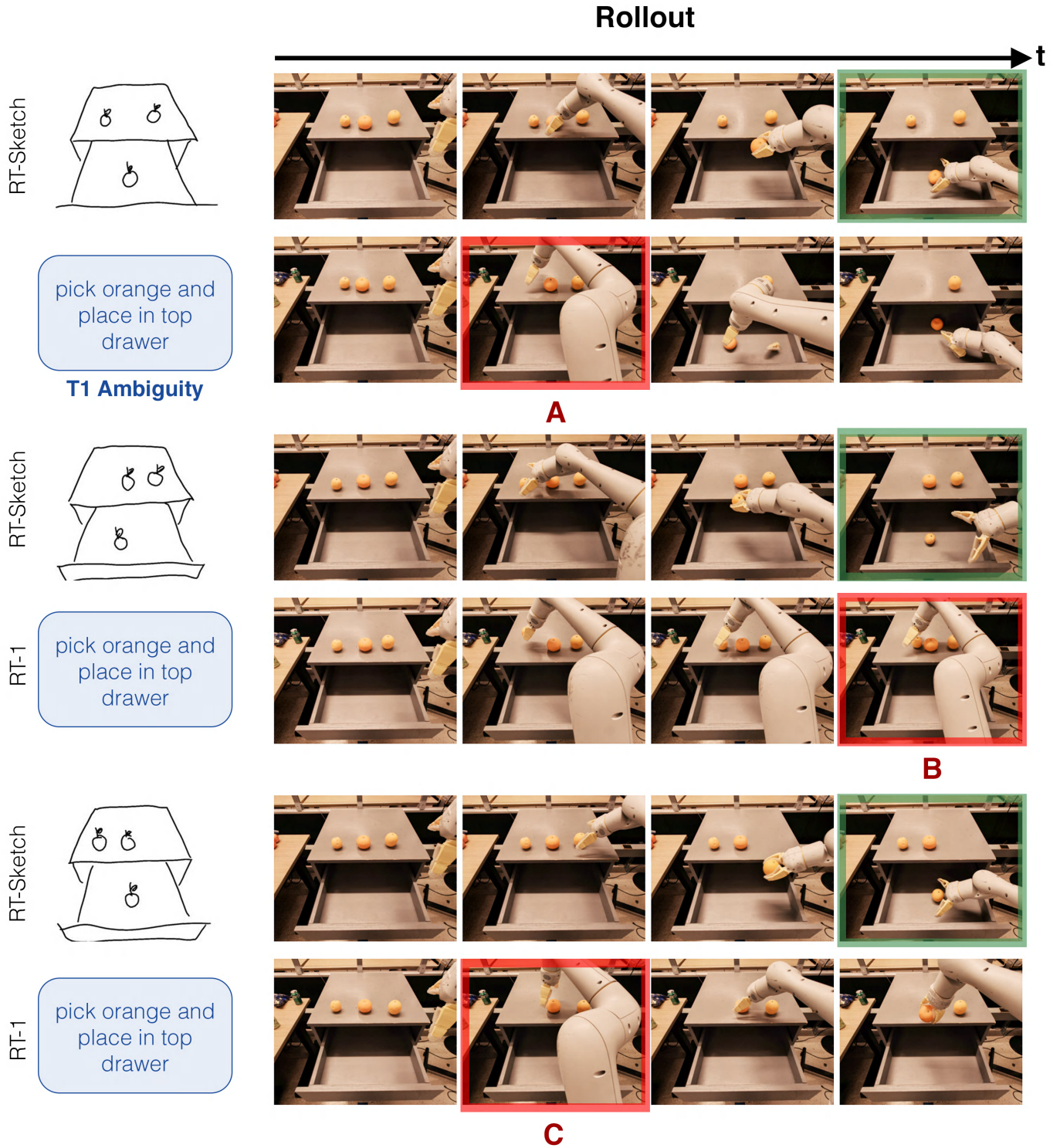


Fig. 14: **H4 Rollout Visualization (T1 as visualized in Fig. 13)**: One source of ambiguity in language descriptions is mentioning an object for which there are multiple instances present. For example, we can easily illustrate three different desired placements of an orange in the drawer via a sketch, but an ambiguous instruction cannot easily specify which orange is relevant to pick and place. In all rollouts, RT-Sketch successfully places the correct orange in the drawer, while RT-1 either picks up the wrong object (A), fails to move to the place location (B), or knocks off one of the oranges (C). Although in this case, the correct orange to manipulate could easily be specified with a spatial relation like *pick up the $\langle \text{left/middle/right} \rangle$ orange*, we show below in Appendix Fig. 15 that this type of language is still out of the realm of RT-1’s semantic familiarity.

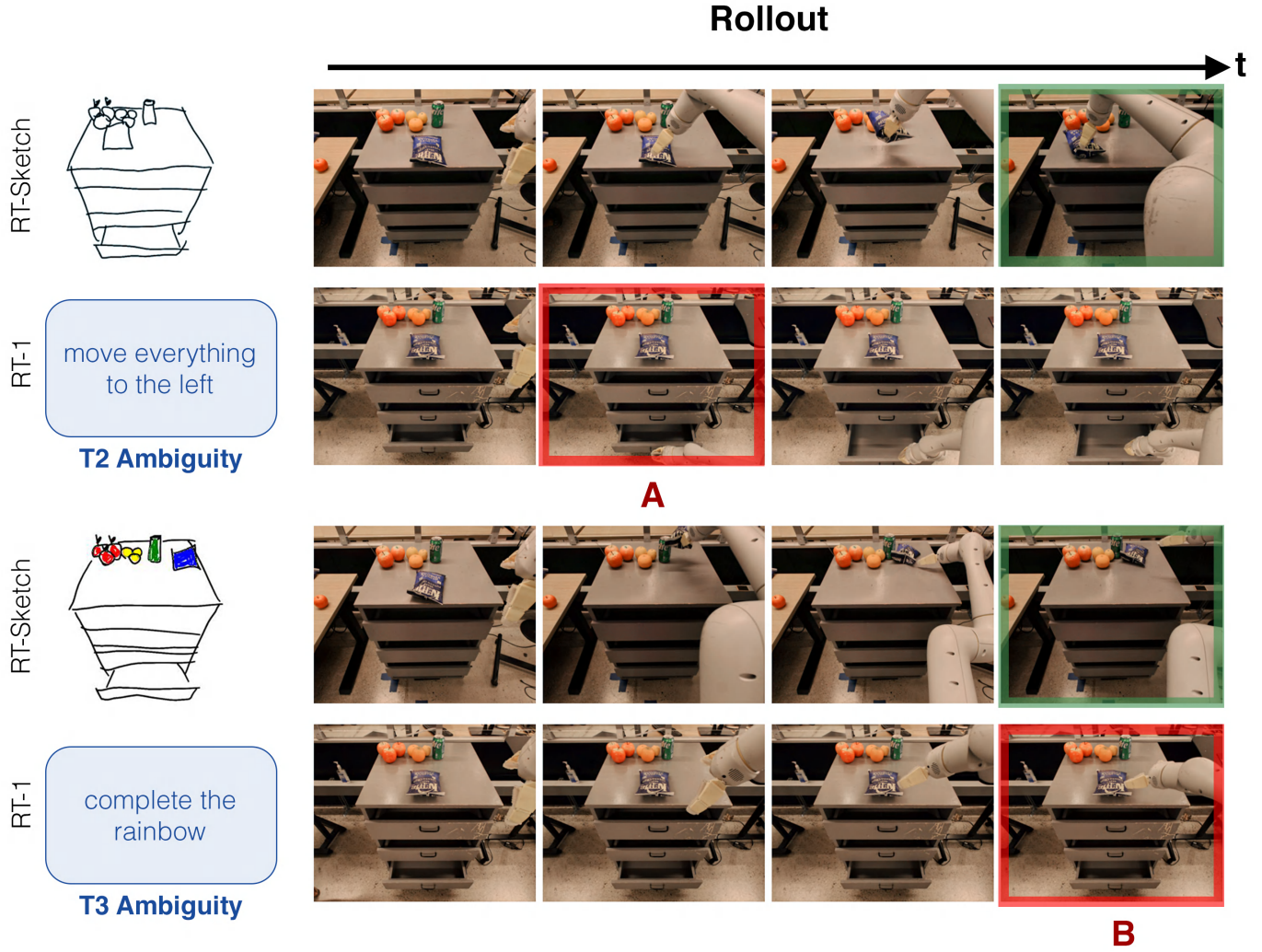


Fig. 15: **H4 Rollout Visualization (T2-3 as visualized in Fig. 13)**: For **T2**, we consider language with spatial cues that intuitively should help the policy disambiguate in scenarios like the oranges in Fig. 14. However, we find that RT-1 is not trained to handle such spatial references, and this kind of language causes a large distribution shift leading to unwanted behavior. Thus, for the top rollout of trying to move the chip bag to the left where there is an existing pile, RT-Sketch completes the skill without issues, but RT-1 attempts to open the drawer instead of even attempting to rearrange anything on the countertop (A). For **T3**, we consider language goals that are even more abstract in interpretation, without explicit objects mentioned or spatial cues. Here, sketches are advantageous in their ability to succinctly communicate goals (i.e. visual representation of a rainbow), whereas the corresponding language task string is far too underspecified and OOD for the policy to handle (B).

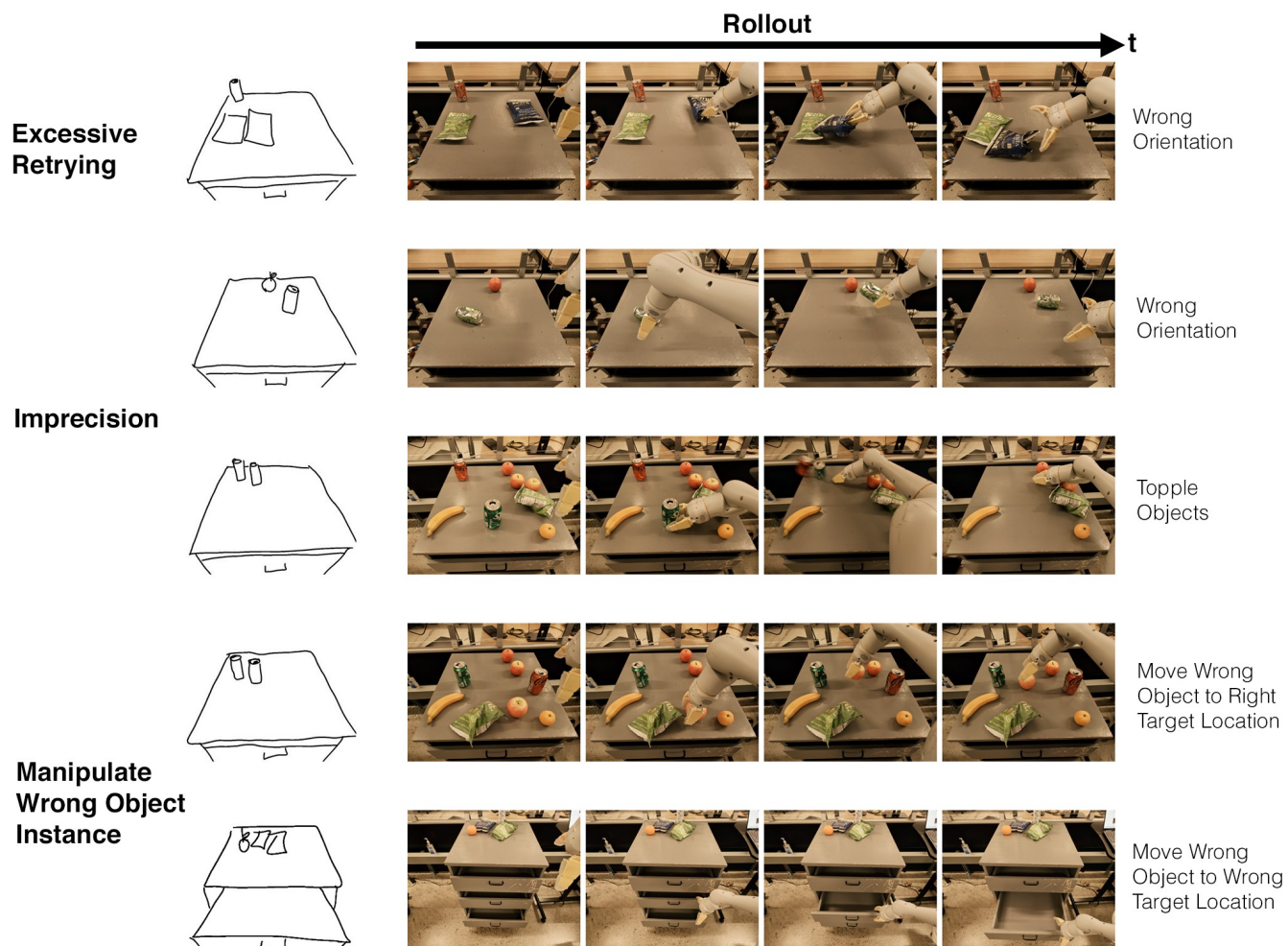


Fig. 16: RT-Sketch Failure Modes

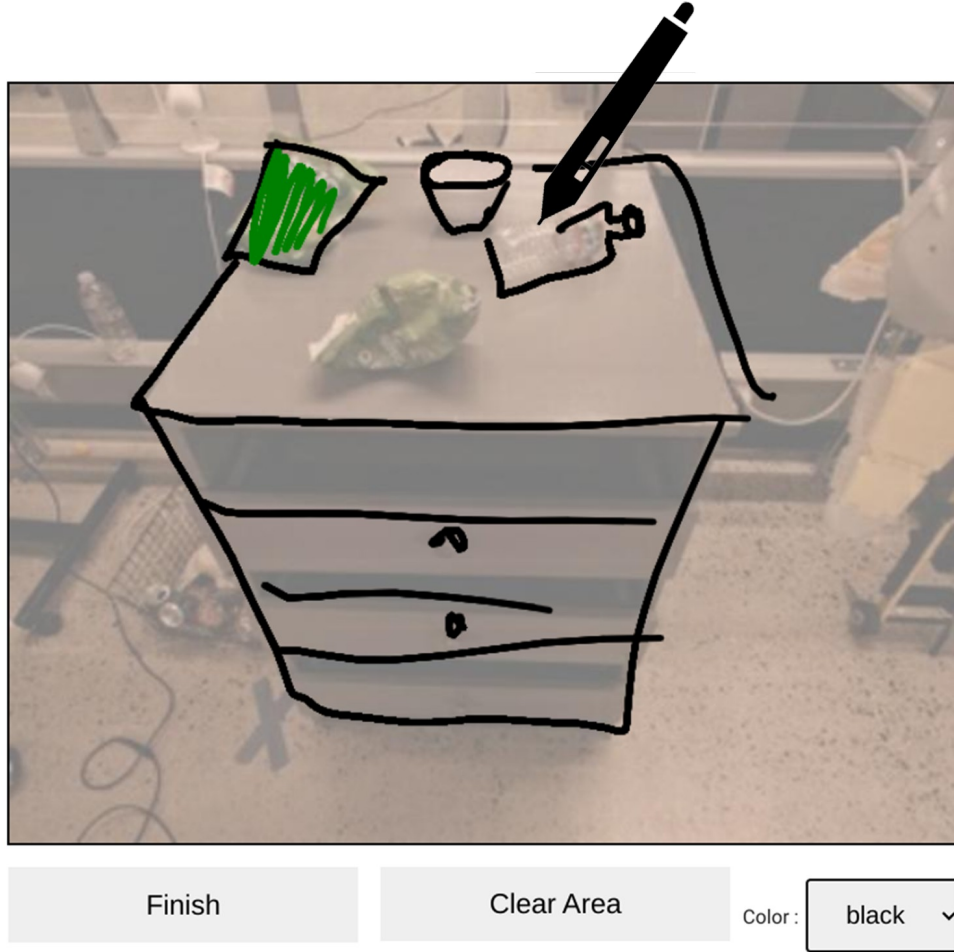


Fig. 17: **Sketching UI**: We design a custom sketching interface for manually collecting paired robot images and sketches with which to train \mathcal{T} , and for sketching goals for evaluation. The interface visualizes the current robot observation, and provides the ability to draw on a digital screen with a stylus. The above visualization shows the color-sketching modality, which is a traced representation with color shading. The interface supports different colors and erasure, along with either *tracing* over the image (line-sketching) or drawing free-form over a blank canvas (free-hand sketches). We note that intuitively, drawing on top of the image is not an unreasonable assumption to make, since current agent observations are typically readily available compared to a goal image, for instance. Additionally, the overlay is intended to make the sketching interface easy for the user to provide, without having to eyeball edges for the drawers or handles blindly. This provides helpful guides for sketching and is an easy way to obtain sketches that more closely align with current observations for free.

Q1

Reference Instruction + Actual Rollout



The robot achieves **semantic alignment** with the goal during the rollout. *

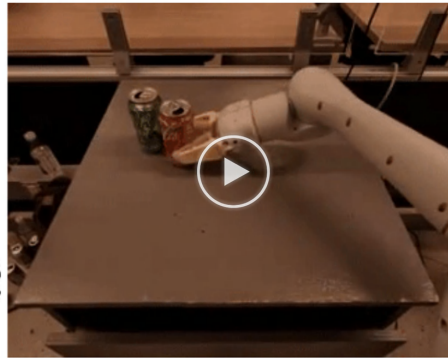
	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

Q2

Reference Goal



Actual Rollout



The robot achieves **spatial alignment** with the goal during the rollout. *

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

Fig. 18: **Assessment UI**: For all skills and methods, we ask labelers to assess semantic and spatial alignment of the recorded rollout relative to the ground truth semantic instruction and visual goal. We show the interface above, where labelers are randomly assigned to skills and methods (anonymized). The results of these surveys are reported in Fig. 3.