

RT-Sketch: Goal-Conditioned Imitation Learning from Hand-Drawn Sketches

Anonymous Author(s)

Affiliation

Address

email

1 **Abstract:** Natural language and images are commonly used as goal representations
2 in goal-conditioned imitation learning. However, language can be ambigu-
3 ous and images can be over-specified. In this work, we study hand-drawn sketches
4 as a modality for goal specification. Sketches can be easy to provide on the fly like
5 language, but like images they can also help a downstream policy to be spatially-
6 aware. By virtue of being minimal, sketches can further help disambiguate task-
7 relevant from irrelevant objects. We present RT-Sketch, a goal-conditioned policy
8 for manipulation that takes a hand-drawn sketch of the desired scene as input, and
9 outputs actions. We train RT-Sketch on a dataset of trajectories paired with syn-
10 thetically generated goal sketches. We evaluate this approach on six manipulation
11 skills involving tabletop object rearrangements on an articulated countertop. Ex-
12 perimentally we find that RT-Sketch performs comparably to image or language-
13 conditioned agents in straightforward settings, while achieving greater robustness
14 when language goals are ambiguous or visual distractors are present. Additionally,
15 we show that RT-Sketch handles sketches with varied levels of specificity, rang-
16 ing from minimal line drawings to detailed, colored drawings. For supplementary
17 material and videos, please visit <http://rt-sketch-anon.github.io>.

18 **Keywords:** Visual Imitation Learning, Goal-Conditioned Manipulation

19 1 Introduction

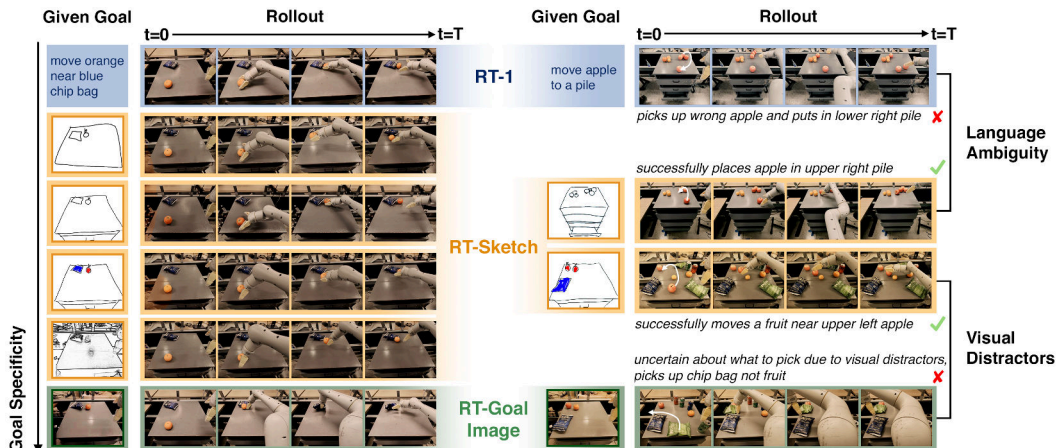


Figure 1: Rollouts showing RT-Sketch’s robustness to sketch detail, ambiguous language, and visual distractors.

20 Robots operating alongside humans in the home or workplace have an immense potential for assis-
21 tance and autonomy, but careful consideration is needed of what goal representations are easiest *for*
22 *humans* to convey to robots, and *for robots* to interpret and act upon.

23 Instruction-following robots attempt to address this problem using the intuitive interface of natural
24 language commands as inputs to language-conditioned imitation learning policies [1, 2, 3, 4, 5]. For

25 instance, imagine asking a household robot to set the dinner table. A language description such
26 as “*put the utensils, the napkin, and the plate on the table*” is under-specified or ambiguous. It is
27 unclear how exactly the utensils should be positioned relative to the plate or the napkin, or whether
28 their distances to each other matter or not. To achieve this higher level of precision, a user may need
29 to give lengthier descriptions such as “*put the fork 2cm to the right of the plate, and 5cm to the*
30 *leftmost edge of the table.*”, or even online corrections (“*no, you moved too far to the right, move*
31 *back a bit!*”) [6, 5]. While intuitive, the qualitative nature and ambiguity of language can make it
32 both inconvenient for humans to provide without lengthy instructions or corrections, and for robot
33 policies to interpret for downstream precise manipulation.

34 Using a goal image (i.e. an image of the scene in its final desired state) to specify objectives and
35 train goal-conditioned imitation learning policies has shown to be quite successful in recent years,
36 with or without language [7, 8, 9]. However, this has its own shortcomings: access to a goal image is
37 a strong prior assumption, and a pre-recorded goal image is tied to a particular environment, making
38 it difficult to reuse for generalization. To summarize: while natural language is highly flexible, it
39 can also be highly ambiguous or require lengthy descriptions. This quickly becomes difficult in
40 long-horizon tasks or those requiring spatial awareness. Meanwhile, goal images over-specify goals
41 in unnecessary detail, leading to the need for internet-scale data for generalization.

42 To address these challenges, we study *hand-drawn sketches* as a convenient yet expressive modality
43 for goal specification. By virtue of being minimal, sketches are still easy to provide on the fly like
44 language, but allow for more spatially-aware task specification. Like goal images, sketches readily
45 integrate with off-the-shelf policy architectures that take visual input, but provide an added level of
46 goal abstraction that ignores unnecessary pixel-level details. Finally, sketches can inform a policy
47 of task relevant/irrelevant objects based on whether details are included/excluded in a sketch.

48 In this work, we present RT-Sketch, a goal-conditioned policy for manipulation that takes a hand-
49 drawn sketch of the desired scene as input, and outputs actions. The novel architecture of RT-
50 Sketch modifies the original RT-1 language-to-action Transformer architecture [1] to consume visual
51 goals rather than language, allowing for flexible conditioning on sketches, images, or any other
52 visually representable goals. To enable this, we concatenate a goal sketch and history of observations
53 as input before tokenization, omitting language. We train RT-Sketch on a dataset of 80K trajectories
54 paired with synthetic goal sketches, generated by an image-to-sketch stylization network trained
55 from a few hundred image-sketch pairs.

56 We evaluate RT-Sketch across six manipulation skills on real robots involving tabletop object rear-
57 rangements on a countertop with drawers, subject to a wide range of scene variations. These skills
58 include rearranging objects, placing cans and bottles sideways or upright, and opening and closing
59 drawers. Experimentally, we find that RT-Sketch performs on a similar level to image or language-
60 conditioned agents in straightforward settings. When language instructions are ambiguous, or in the
61 presence of visual distractors (Figure 1, right), we find that RT-Sketch achieves 2.71X and 1.63X
62 higher spatial alignment scores over language or goal image-conditioned policies, respectively (see
63 Fig. 3 (H3/4)). Additionally, we show that RT-Sketch can handle different levels of input specificity,
64 ranging from rough sketches to more scene-preserving, colored drawings (Fig. 1, left). Finally, we
65 also include results that suggest the compatibility of sketches with language, showing promise of
66 multimodal goal specification in the future.

67 2 Related Work

68 In this section, we discuss prior methods for goal-conditioned imitation learning (IL) and recent
69 efforts towards image-sketch translation, which we build on towards sketch-condition IL.

70 **Goal-Conditioned Imitation Learning** Reinforcement learning (RL) is not easily applicable in
71 our scenario, as it is nontrivial to define a reward objective which accurately quantifies alignment
72 between a provided scene sketch and states achieved by an agent. We instead focus on IL techniques,
73 particularly the goal-conditioned setting [10]. Goal-conditioned IL has proven useful in settings
74 where a policy needs to handle different variations of the same task [11]. Examples include moving
75 objects into different arrangements [1, 2, 5, 12, 9], kitting [13], folding of deformable objects into
76 different configurations [14], and search for different target objects in clutter [15]. However, these

77 approaches tend to condition on either language [1, 4, 5, 3, 16], or images [15] to specify goals.
78 Follow-up work enabled multimodal conditioning on either goal images and language [8], in-prompt
79 images [7], or image embeddings [12, 13, 14]. All of these representations are ultimately derived
80 from raw images or language, which overlooks the potential for more abstract goals like sketches.

81 Beyond inflexible goal representations, goal-conditioned IL tends to overfit to demonstration data
82 and fails to handle even slight distribution shifts [17]. For language, this can encompass ambigu-
83 ous or novel phrasing or unseen objects [8, 1]. Goal-image conditioning is similarly susceptible to
84 out-of-distribution visual shifts, such as lighting variations or unseen object and background appear-
85 ances [18, 19]. Instead, sketches are minimal enough to combat visual distractors, yet expressive
86 enough to provide unambiguous goals. Prior work, including [20] and [21], have shown the utili-
87 ty of sketches over pure language for navigation and limited manipulation. However, the sketches
88 explored in these works are largely intended to guide low-level motion at the joint-level for manip-
89 ulation, or provide explicit directional cues for navigation. [22] considers sketches amongst other
90 modalities as an input for goal-conditioned manipulation, but does not explicitly train a policy condi-
91 tioned on sketches. They thus came to the conclusion that the scene image is better than the sketch at
92 goal specification. Our result is different and complementary, in that policies trained to take sketches
93 as input outperform a scene image conditioned policy, by 1.63x and 1.5x in terms of Likert ratings
94 for perceived spatial and semantic alignment, subject to visual distractors. Other recent works pro-
95 pose goal-conditioning on sketches that either represent the intended direction of positional [23, 24]
96 or joint-level [25] robot movement. In contrast to these *motion-centric* representations, the sketches
97 in our work are *scene-centric*, representing the desired visual goal state rather than the desired robot
98 actions.

99 **Image-Sketch Conversion** Sketches have been studied within the computer vision community
100 for object detection [26, 27, 28], visual question answering [29, 30], and scene understanding [31],
101 either in isolation or in addition to text and images. When considering how best to incorporate
102 sketches in IL, an important design choice is whether to take sketches into account (1) at test time
103 (by converting a sketch to another modality compatible with a pre-trained policy), or (2) at train
104 time (by explicitly training a policy conditioned on sketches). For (1), one could first convert a
105 given sketch to a goal image, and then roll out a vanilla goal-image conditioned policy. Existing
106 frameworks tackle sketch-to-image conversion, such as ControlNet [32], GAN-style approaches
107 [33], or text-to-image synthesis, such as InstructPix2Pix [34] or Stable Diffusion [35]. While these
108 models can produce photorealistic visuals, they do not jointly handle image generation and style
109 transfer, making it unlikely for generated images to match the style of agent observations. These
110 approaches are also susceptible to hallucinated artifacts, introducing distribution shifts [32].

111 Thus, we instead opt for (2), and consider image-to-sketch conversion techniques for hindsight re-
112 labeling of demonstrations. Recently, Vinker et al. [36, 37] propose networks for predicting Bezier
113 curve-based sketches of input images, supervised by a CLIP-based alignment metric. While these
114 approaches generate visually compelling sketches, test-time generation takes on the order of min-
115 utes, which does not scale to the typical size of robot learning datasets with hundreds to thousands of
116 trajectories. Meanwhile, conditional generative adversarial networks (cGANs) such as Pix2Pix [38]
117 have proven useful for scalable image-to-image translation. Most related to our work is that of Li
118 et al. [39], which trains a Pix2Pix model to produce sketches from given images on a large crowd-
119 sourced dataset of 5K paired images and line drawings. We build on this work to fine-tune an
120 image-to-sketch model that maps robot observations to sketches, with which to train an IL policy.

121 3 Sketch-Conditioned Imitation Learning

122 **Problem Statement** We first formalize the problem of learning a manipulation policy conditioned
123 on a goal *sketch* of the desired scene state and a history of interactions. We denote such a pol-
124 icy by $\pi_{\text{sketch}}(a_t|g, \{o_j\}_{j=1}^t)$, where a_t denotes an action at timestep t , $g \in \mathbb{R}^{W \times H \times 3}$ is a given
125 goal sketch with width and height W and H , and $o_t \in \mathbb{R}^{W \times H \times 3}$ is an observation at t . At in-
126 ference time, the policy takes a given goal sketch along with a history of D previous RGB im-
127 age observations, and outputs an action. To train such a policy, we assume access to a dataset
128 $\mathcal{D}_{\text{sketch}} = \{g^n, \{(o_t^n, a_t^n)\}_{t=1}^{T^{(n)}}\}_{n=1}^N$ of N successful demonstrations, where $T^{(n)}$ refers to the length

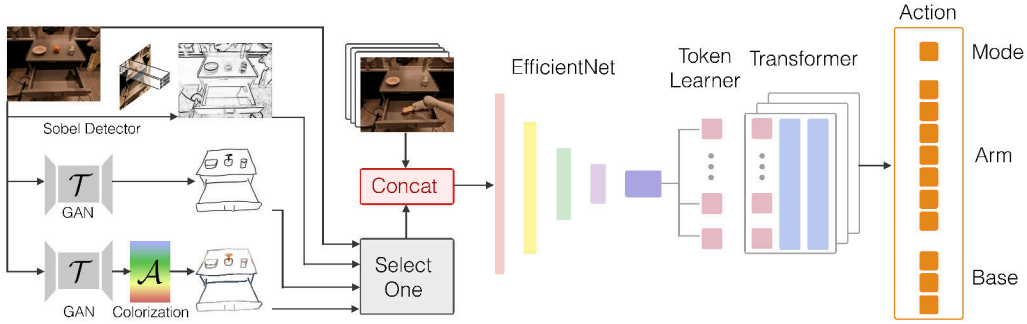


Figure 2: Architecture of RT-Sketch allowing different kinds of visual input. RT-Sketch adopts the Transformer [40] architecture with EfficientNet [41] tokenization at the input, and outputs bucketized actions.

129 of the n^{th} trajectory in timesteps. Each episode of the dataset consists of a given goal sketch and a
 130 corresponding demonstration trajectory, with images recorded at each timestep. Our goal is to thus
 131 learn the sketch-conditioned imitation policy $\pi_{\text{sketch}}(a_t | g, \{o_j\}_{j=1}^t)$ trained on $\mathcal{D}_{\text{sketch}}$.

132 3.1 Image-to-Sketch Translation

133 Training a sketch-conditioned policy requires a dataset of robot trajectories, each paired with a goal
 134 sketch. Collecting both demonstration trajectories and manually drawn sketches at scale is imprac-
 135 tical. Thus, we instead aim to learn an image-to-sketch translation network $\mathcal{T}(g|o)$ that takes an
 136 image observation o and outputs the corresponding goal sketch g . This network can be used to
 137 post-process an existing dataset of demonstrations $\mathcal{D} = \{(o_t^n, a_t^n)\}_{t=1}^{T(n)}\}_{n=1}^N$ with image observa-
 138 tions by appending a synthetically generated goal sketch to each demonstration. This produces a
 139 dataset for sketch-based IL: $\mathcal{D}_{\text{sketch}} = \{g^n, \{(o_t^n, a_t^n)\}_{t=1}^{T(n)}\}_{n=1}^N$. In practice, we use the existing
 140 large-scale dataset of VR-teleoperated robot demonstrations from prior work, which included skills
 141 such as object pick and place, placing cans and bottles upright or sideways, and opening and closing
 142 cabinets [1]. Prior work previously trained a language-conditioned IL policy RT-1 from this data,
 143 but we extend this policy architecture to accommodate sketches, detailed in Section 3.2.

144 **Assumptions on Sketches** There are innumerable ways for a human to provide a sketch corre-
 145 sponding to a given image of a scene. For controlled evaluation, we first assume that a given sketch
 146 respects the task-relevant contours of an associated image, such that tabletop edges, drawer handles,
 147 and task-relevant objects are included and discernible in the sketch. We do not assume contours in
 148 the sketch to be edge-aligned or pixel-aligned with those in an image. We do assume that the input
 149 sketch consists of black outlines at the very least, with optional color shading. We further assume
 150 that sketches do not contain information not present in the associated image, such as hallucinated
 151 objects, scribbles, or text, but may omit task-irrelevant details that appear in the original image.

152 **Sketch Dataset Generation** To train an image-to-sketch translation network \mathcal{T} , we collect a new
 153 dataset $\mathcal{D}_{\mathcal{T}} = \{(o_i, g_i^1, \dots, g_i^{L(i)})\}_{i=1}^M$ consisting of M image observations o_i each paired with a
 154 set of goal sketches $g_i^1, \dots, g_i^{L(i)}$. Those represent $L(i)$ different representations of the same image
 155 o_i , in order to account for the fact that there are multiple, valid ways of sketching the same scene.
 156 To collect $\mathcal{D}_{\mathcal{T}}$, we take 500 randomly sampled terminal images from demonstration trajectories in
 157 the RT-1 dataset, and manually draw sketches with black lines on a white background capturing the
 158 tabletop, drawers, and relevant objects visible on the table. While we personally annotate each robot
 159 observation with just one single sketch, we add this data to an existing, much larger non-robotic
 160 dataset of paired images and sketches [39]. This dataset captures inter-sketch variation via multiple
 161 crowdsourced sketches per image. We do not include the robot arm in our manual sketches, as we
 162 find a minimal representation to be most natural. Empirically, we find that our policy can handle
 163 such sketches despite actual goal configurations likely having the arm in view. We collect these
 164 drawings using a custom digital stylus drawing interface where user draws an edge-aligned sketch
 165 over the original image (Appendix Fig. 17) by *tracing outlines*. The final recorded sketch includes
 166 the user’s strokes in black on a white canvas.

167 **Image-to-Sketch Training** We implement the image-to-sketch translation network \mathcal{T} with the
 168 Pix2Pix conditional generative adversarial network (cGAN) architecture, which is composed of a
 169 generator $G_{\mathcal{T}}$ and a discriminator $D_{\mathcal{T}}$ [38]. The generator $G_{\mathcal{T}}$ takes an input image o , a random
 170 noise vector z , and outputs a goal sketch g . The discriminator $D_{\mathcal{T}}$ is trained to discriminate amongst
 171 artificially generated versus ground truth sketches. We utilize the standard cGAN supervision loss to
 172 train both [39, 38]: $\mathcal{L}_{\text{cGAN}} = \min_{G_{\mathcal{T}}} \max_{D_{\mathcal{T}}} \mathbb{E}_{o,g} [\log D_{\mathcal{T}}(o, g)] + \mathbb{E}_{o,g} [\log(1 - D_{\mathcal{T}}(o, G_{\mathcal{T}}(o, g)))]$.

173 We also add the \mathcal{L}_1 loss to encourage the produced sketches to align with ground truth sketches as
 174 in [39]. To account for the fact that there may be multiple valid sketches for a given image, we only
 175 penalize the minimum \mathcal{L}_1 loss incurred across all $L^{(i)}$ sketches provided for a given image as in Li
 176 et al. [39]. This is to prevent wrongly penalizing \mathcal{T} for producing a valid sketch that aligns well with
 177 one example but not another simply due to stylistic differences in the ground truth sketches. The
 178 final objective is a λ -weighted combination of the average cGAN loss and the minimum alignment

$$179 \text{ loss: } \mathcal{L}_{\mathcal{T}} = \frac{\lambda}{L^{(i)}} \sum_{k=1}^{L^{(i)}} \mathcal{L}_{\text{cGAN}}(o_i, g_i^{(k)}) + \min_{k \in \{1, \dots, L^{(i)}\}} \mathcal{L}_1(o_i, g_i^{(k)})$$

180 In practice, we supplement the 500 manually drawn sketches from $\mathcal{D}_{\mathcal{T}}$ by leveraging the existing
 181 larger-scale Contour Drawing Dataset [39]. We refer to this dataset as \mathcal{D}_{CD} , which contains 1000
 182 examples of internet-scraped images containing objects, people, animals from Adobe Stock, paired
 183 with $L^{(i)} = 5$ crowd-sourced black and white outline drawings per image collected on Amazon
 184 Mechanical Turk (see Appendix Fig. 6 for examples). We first take a pre-trained image-to-sketch
 185 translation network \mathcal{T}_{CD} [39] trained on \mathcal{D}_{CD} , with $L^{(i)} = 5$ sketches per image. Then, we fine-tune
 186 \mathcal{T}_{CD} on $\mathcal{D}_{\mathcal{T}}$, with only $L^{(i)} = 1$ manually drawn sketch per robot observation, to obtain our final
 187 image-to-sketch network \mathcal{T} . Visualizations of sketches generated by \mathcal{T} are available in Fig. 7.

188 3.2 RT-Sketch

189 With a way to translate image observations to sketches via \mathcal{T} (Section 3.1), we can automatically
 190 augment the RT-1 dataset with goal sketches $\mathcal{D}_{\text{sketch}}$ with which to train our policy RT-Sketch.

191 **RT-Sketch Dataset** The original RT-1 dataset $\mathcal{D}_{\text{lang}} = \{i^n, \{(o_t^n, a_t^n)\}_{t=1}^{T^{(n)}}\}_{n=1}^N$ consists of N
 192 episodes with a paired natural language instruction i and demonstration trajectory $\{(o_t^n, a_t^n)\}_{t=1}^{T^n}$.
 193 We can automatically hindsight-relabel such a dataset with goal images instead of language
 194 goals [42]. Let us denote the last step of a trajectory n as $T^{(n)}$. Then the new dataset with image
 195 goals instead of language goals is $\mathcal{D}_{\text{img}} = \{o_{T^{(n)}}^n, \{(o_t^n, a_t^n)\}_{t=1}^{T^{(n)}}\}_{n=1}^N$, where we treat the last
 196 observation of the trajectory $o_{T^{(n)}}^n$ as the goal g^n . To produce a dataset for π_{sketch} , we can simply
 197 replace $o_{T^{(n)}}^n$ with $\hat{g}^n = \mathcal{T}(o_{T^{(n)}}^n)$ such that $\mathcal{D}_{\text{sketch}} = \{\hat{g}^n, \{(o_t^n, a_t^n)\}_{t=1}^{T^{(n)}}\}_{n=1}^N$.

198 To encourage the policy to afford different levels of input sketch specificity, we in practice produce
 199 goals by $\hat{g}^n = \mathcal{A}(o_{T^{(n)}}^n)$, where \mathcal{A} is a randomized augmentation function. \mathcal{A} chooses between sim-
 200 ply applying \mathcal{T} , \mathcal{T} with colorization during postprocessing (e.g., superimposing a blurred version of
 201 the ground truth RGB image over the binary sketch), a Sobel operator [43] for edge detection, or an
 202 identity operation, which preserves the original image (Fig. 2). By co-training on all representations,
 203 we intend for RT-Sketch to handle a spectrum of specificity going from binary sketches; colored
 204 sketches; edge detected images; and goal images (Appendix Fig. 7).

205 **RT-Sketch Model Architecture** In our setting, we consider goals provided as sketches rather
 206 than language as was done in RT-1. The original RT-1 policy relies on a Transformer architecture
 207 backbone [40]. RT-1 first passes a history of $D = 6$ images through an EfficientNet-B3 model [41]
 208 producing image embeddings, which are tokenized, and separately extracts textual embeddings and
 209 tokens via FiLM [44] and a Token Learner [45]. The tokens are then fed into a Transformer which
 210 outputs bucketed actions: a 7-DoF output for the end-effector (x, y, z , roll, pitch, yaw, gripper
 211 width), 3-DoF for the mobile base, (x, y , yaw), and 1 mode-switching flag (base movement, arm
 212 movement, and termination). To accommodate our change in the input, we omit the FiLM language
 213 tokenization altogether. Instead, we concatenate a given visual goal with the history of images as
 214 input to EfficientNet, and extract tokens from its output, leaving the rest of the policy architecture
 215 unchanged. We train two policies using this architecture (Fig. 2): RT-Sketch refers to our policy
 216 trained from sketches, and RT-Goal-Image is a baseline policy trained from goal images.

217 **Training RT-Sketch** We now train π_{sketch} on $\mathcal{D}_{\pi_{\text{sketch}}}$ from scratch (rather than finetuning an
 218 existing backbone) using the same procedure as in RT-1 [1], with the above architectural changes.
 219 We fit the policy using the behavioral cloning objective that minimizes the negative log-likelihood
 220 of an action [46]: $J(\pi_{\text{sketch}}) = \sum_{n=1}^N \sum_{t=1}^{T^{(n)}} \log \pi_{\text{sketch}}(a_t^n | g^n, \{o_j\}_{j=1}^t)$

221 4 Experiments

222 We seek to understand the ability of RT-Sketch to perform goal-conditioned manipulation as com-
 223 pared to language or image-conditioned policies. To that end, we test the following four hypotheses:

224 **H1: RT-Sketch is successful at goal-conditioned IL.** While abstract, we hypothesize that sketches are specific
 225 enough to provide manipulation goals to a policy. We thus expect RT-Sketch to perform on a similar level to
 226 language (RT-1) or image goals (RT-Goal-Image) in straightforward tasks.

227 **H2: RT-Sketch is able to handle varying levels of specificity.** Having trained RT-Sketch on sketches of
 228 varying levels of specificity, we expect it to be robust against sketch variations for the same scene.

229 **H3: Sketches enable better robustness to distractors than goal images.** Sketches focus on task-relevant de-
 230 tails of a scene, while images capture everything. Therefore, we expect RT-Sketch to provide better robustness
 231 than RT-Goal-Image against irrelevant distractors in the environment.

232 **H4: Sketches are favorable when language is ambiguous.** We expect RT-Sketch to provide a higher success
 233 rate compared to ambiguous language inputs when using RT-1.

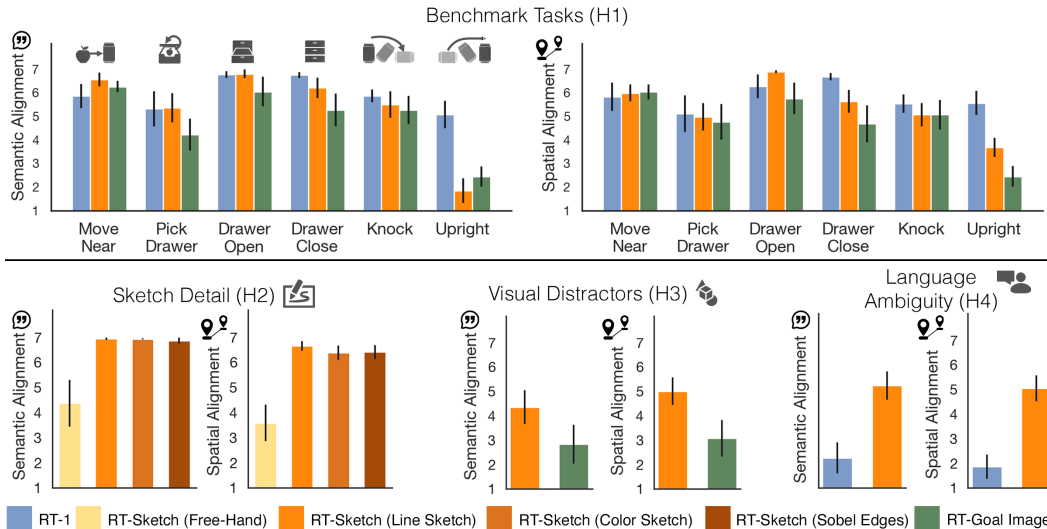


Figure 3: **Goal Alignment Results:** Average Likert scores for different policies rating perceived semantic alignment (Q1) and spatial alignment (Q2) to a provided goal. Error bars indicate standard error. To back up the visual insights from these barplots, we report additional findings on statistically significant differences between methods from a non-parametric Mann-Whitney U test in Appendix B

234 4.1 Experimental Setup

235 **Policies** We compare RT-Sketch to the original language-conditioned agent RT-1 [1], and a goal
 236 image-conditioned agent RT-Goal-Image. All policies are trained on a multi-task dataset of $\sim 80K$
 237 real-world trajectories manually collected via VR teleoperation using the setup from Brohan et al.
 238 [1]. These trajectories span 6 common household object rearrangement tasks: *move X near Y*, *place*
 239 *X upright*, *knock X over*, *open the X drawer*, *close the X drawer*, and *pick X from Y*.

240 **Evaluation protocol** To fairly compare different policies, we use a shared catalog of heldout eval-
 241 uation scenarios. Each scenario includes an initial image of the scene, a goal image with objects
 242 arranged as desired, a natural language task description, and hand-drawn sketches of the goal. At
 243 test time, a human operator retrieves a scenario, aligns the robot and scene using a reference im-
 244 age and a custom visualization utility, and places objects accordingly. We then roll out a policy
 245 conditioned on one of the available goals (language, image, sketch, etc.), and record a video for
 246 downstream evaluation (see Section 4.2). All experiments utilize the mobile Everyday Robot with
 247 an overhead camera and a 7-DoF arm with a parallel jaw gripper. All sketches for evaluation are
 248 collected by a single human annotator on a custom drawing interface with a tablet and digital stylus.

249 **Metrics** Defining a standardized evaluation protocol for goal alignment is non-trivial when binary
 250 task success is too coarse and image-similarity metrics like CLIP [47] can be brittle. We first measure
 251 performance by quantifying the policy precision as the pixel distance between object centroids in
 252 achieved and ground truth goal states, using manual keypoint annotations (see Fig. 9 in Appendix
 253 for examples). Although leveraging out-of-the box object detectors to detect object centroids is a
 254 possibility, we want to avoid conflating errors in detection (imprecise/wrong bounding box, etc.)
 255 with manipulation policy errors. Second, we gather human-provided assessments of perceived goal
 256 alignment via 2 Likert questions [48], rated from 1-7 (Strongly Disagree - Strongly Agree):

257 (Q1) *The robot achieves semantic alignment with the given goal during the rollout.*

258 (Q2) *The robot achieves spatial alignment with the given goal during the rollout.*

Skill	Spatial Precision (RMSE in px.)			Failure Occurrence (Excessive Retrying)		
	RT-1	RT-Sketch	RT-Goal-Image	RT-1	RT-Sketch	RT-Goal-Image
Move Near	5.43 ± 2.15	3.49 ± 1.38	3.89 ± 1.16	0.00	0.06	0.33
Pick Drawer	5.69 ± 2.90	4.77 ± 2.78	4.74 ± 2.01	0.00	0.13	0.20
Drawer Open	4.51 ± 1.55	3.34 ± 1.08	4.98 ± 1.16	0.00	0.00	0.07
Drawer Close	2.69 ± 0.93	3.02 ± 1.35	3.71 ± 1.67	0.00	0.00	0.07
Knock	7.39 ± 1.77	5.36 ± 2.74	5.63 ± 2.60	0.00	0.13	0.40
Upright	7.84 ± 2.37	5.08 ± 2.08	4.18 ± 1.54	0.06	0.00	0.27
Visual Distractors	-	4.78 ± 2.17	7.95 ± 2.86	-	0.13	0.67
Language Ambiguity	8.03 ± 2.52	4.45 ± 1.54	-	0.40	0.13	-

Table 1: **Spatial Precision / Failure Occurrence:** We report (1) the spatial precision (root mean squared pixel error, RMSE) of the centroids of manipulated objects in achieved vs. given reference goal images (left, darker=more precise) and (2) the occurrence of *excessive retrying* failures (right, bold=least failure-prone).

259 For **Q1**, we present labelers with the policy rollout video along with the language goal. To answer
 260 **Q2**, we present labelers with a policy rollout video side-by-side with a visual goal (ground truth
 261 image, sketch, etc.). A policy can for instance achieve high semantic alignment for the language
 262 goal *place can upright* as long as the can ends up in the right orientation, but will not achieve spatial
 263 alignment unless the can is additionally in the correct position on the table.

264 Appendix Fig. 18 visualizes the assessment interface. We perform these human assessment surveys
 265 across 62 unpaid individuals (non-expert, unfamiliar with our system) who are blind to whether they
 266 assess our approach or a baseline. We assign between 8 and 12 people to evaluate each of the 6
 267 different manipulation skills considered below. Note that this evaluation is NOT a *user study*, as
 268 we are not attempting to study humans, and is merely used as a fair means of *labeling* rollouts to
 269 measure goal alignment across policies.

270 4.2 Experimental Results

271 In this section, we present our findings related to the hypotheses of Section 4 by quantifying precision
 272 (Table 1, Table 2) and goal alignment (Fig. 3)) across policies.

273 **H1:** We evaluate all policies on each of the 6 skills on 15 different evaluation catalog scenarios per
 274 skill, varying objects (16 unique in total) and their placements. Overall, RT-Sketch performs com-
 275 parably to RT-1 and RT-Goal-Image in both semantic (**Q1**) and spatial alignment (**Q2**), achieving
 276 average ratings from ‘Agree’ to ‘Strongly Agree’ for nearly all skills (Fig. 3 (top)). The exception is
 277 *upright*; both RT-Sketch and RT-Goal-Image tend to *position* cans or bottles appropriately, without
 278 realizing the need for *reorientation* (Appendix Fig. 10). This results in low semantic alignment but
 279 somewhat higher spatial alignment (Fig. 3 (top), darker gray in Table 1 (left)). RT-1, on the other
 280 hand, reorients cans and bottles successfully, but at the expense of higher spatial error (Appendix
 281 Fig. 10, light color in Table 1 (left)). With RT-Goal-Image in particular, we also observe the oc-
 282 currence of *excessive retrying behavior*, in which a policy attempts to align the current scene with a
 283 given goal with retrying actions that inadvertently disturb the scene, knocking objects off the table or
 284 undoing task progress. In Table 1, we report the proportion of rollouts in which this occurs (via man-
 285 ual inspection) across all policies. RT-Goal-Image is most susceptible, as a result of over-attending
 286 to pixel-level details, while RT-Sketch and RT-1 are far less vulnerable, given the higher-level goal
 287 abstractions that sketches and language offer.

288 **H2:** Next, we assess RT-
 289 Sketch’s ability to handle vary-
 290 ing levels of sketch detail.
 291 Across 5 trials of the *move near*
 292 and *open drawer* skills, we see
 293 in Table 2 that many different sketch types result in reasonable levels of spatial precision, partic-
 294 ularly: free-hand sketches drawn completely free-form on a blank canvas, line sketches drawn by

Skill	Free-Hand	Line Sketch	Color Sketch	Sobel Edges
Move Near	7.21 ± 2.76	3.49 ± 1.38	3.45 ± 1.03	3.36 ± 0.66
Drawer Open	3.75 ± 1.63	3.34 ± 1.08	2.48 ± 0.50	2.13 ± 0.25

Table 2: **RT-Sketch Spatial Precision across Sketch Types:** The relatively small differences in policy precision (RMSE) across different sketch types (i.e. minimal line sketches vs. edge-detected images) suggests RT-Sketch’s robustness to input specificity (darker=better).

295 tracing an image, line sketches with color shading, and edge-detected images. Appendix Fig. 17
296 shows the interface used to sketch, and a detailed breakdown of the differences. As expected, So-
297 bel edge-detected images incur the least error, but they are impractical and merely represent an
298 upper-bound in terms of sketch detail. Even free-hand sketches, which do not necessarily preserve
299 perspective projection, and line sketches, which are far sparser in detail, are not far behind in terms
300 of precision or alignment ratings. This is reflected in the Likert ratings (Fig. 3 (left, bottom)) of
301 free-hand sketches (around 4 on average), and line sketches (nearly 7 – “Strongly Agree” on aver-
302 age). Adding color to line sketches does not further improve performance, but leads to interesting
303 behavioral differences (see Appendix Fig. 11). In Appendix A.2, we also evaluate RT-Sketch on
304 sketches drawn by 6 different individuals whose sketches were never seen during training and ob-
305 serve little-to-no policy performance drop-off compared to in-distribution sketches.

306 **H3:** Next, we compare the robustness of RT-Sketch and RT-Goal-Image to the presence of visual
307 distractors. On 15 *move X near Y* trials from the evaluation catalog, we introduce 5 – 9 distractor
308 objects into the initial visual scene, replicating the setup of the RT-1 generalization experiments
309 referred to as *medium-high* difficulty [1]. In Table 1 (left, bottom), we see that RT-Sketch exhibits
310 far lower spatial errors on average, while producing higher semantic and spatial alignment scores
311 over RT-Goal-Image (Fig. 3 (middle, bottom)). RT-Goal-Image is easily confused by the distribution
312 shift introduced by distractor objects, and often cycles between picking up and putting down the
313 wrong object. RT-Sketch, on the other hand, ignores task-irrelevant objects not captured in a sketch
314 and completes the task in most cases (see Appendix Fig. 12).

315 **H4:** Finally, we evaluate whether sketches as a representation are favorable when language goals
316 alone are ambiguous. On 15 evaluation catalog scenarios, we consider 3 types of language ambi-
317 guity: instance (**T1**) (e.g., *move apple near orange* when multiple orange instances are present),
318 somewhat out-of-distribution (OOD) phrasing (**T2**) (e.g., *move left apple near orange*), and highly
319 OOD phrasing (**T3**) (e.g., *complete the rainbow*) (see Appendix Fig. 13). Directional cues (i.e.
320 ‘left’) should intuitively help resolve ambiguities, but were unseen during RT-1 training [1], and
321 hence are out-of-distribution. In these scenarios, [RT-Sketch achieves](#) nearly half the error of RT-
322 1 (Table 1 (left, bottom)), and a 2.33-fold and 2.71-fold score increase for semantic and spatial
323 alignment, respectively (Fig. 3 (right, bottom)). For **T1** and **T2** scenarios, RT-1 often tries to pick up
324 an instance of any object mentioned in the task string, but fails to make further progress (Appendix
325 Fig. 14). This suggests the utility of sketches to express new, unseen goals with minimal overhead,
326 when language can easily veer out of distribution (Appendix Fig. 15).

327 **Towards Multimodal Goal Specification** For cases in which one modality alone is still am-
328 biguous, we provide initial demonstrations showing that a multimodal (sketch-and-language con-
329 ditioned) policy can be favorable to either alone, especially for tasks involving repositioning and
330 reorientation (see Appendix A.3).

331 4.3 Limitations and Failure Modes

332 Firstly, the image-to-sketch generation network used in this work is fine-tuned on a dataset of
333 sketches provided by a single human annotator. Although we empirically show that despite this, RT-
334 Sketch can handle sketches drawn by other annotators (Appendix A.2), we have yet to investigate
335 the effects of training RT-Sketch at scale with sketches produced by different people. [An additional](#)
336 [challenge is handling extremely minimal sketches. These kinds of sketches remain difficult for](#)
337 [our policy to handle due to obvious perspective changes or missing details. Applying our existing](#)
338 [sketch augmentations at more extremes may help further address this class of sketches.](#) Secondly,
339 we note that RT-Sketch shows some inherent biases towards performing certain skills it was trained
340 on (i.e. performing directional movements that are more represented in the demonstration trajec-
341 tories). [Performing unseen or complex tasks with low tolerance for error also remains challenging.](#)
342 [However, we posit that addressing these issues may require policy-level rather than just goal-level](#)
343 [improvements.](#) For a detailed breakdown of RT-Sketch’s limitations and failure modes, please see
344 [Appendix F](#).

345 5 Conclusion

346 We propose RT-Sketch, a goal-conditioned policy for manipulation that takes a hand-drawn scene
347 sketch as input, and outputs actions. We do so by developing a scalable way to generate paired
348 sketch-trajectory training data via an image-to-sketch translation network, and modifying the ex-
349 isting RT-1 architecture to take visual information as an input. Empirically, RT-Sketch not only

350 performs comparably to existing language or goal-image conditioning policies for a number of
351 manipulation skills, but is amenable to different degrees of sketch fidelity, and more robust to visual
352 distractors or ambiguities. Our rigorous evaluations comprise 400 cumulative robot rollouts, eval-
353 uated across 62 annotators (over 8 cumulative hours). Future work will focus on multimodal goal
354 specification and moving towards even more abstract goal representations, detailed in [Appendix C](#).

355 References

- 356 [1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Haus-
357 man, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Ju-
358 lian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath,
359 I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. S.
360 Ryoo, G. Salazar, P. R. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran,
361 V. Vanhoucke, S. Vega, Q. H. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich.
362 RT-1: Robotics Transformer for Real-World Control at Scale. In *Proceedings of Robotics:
363 Science and Systems*, Daegu, Republic of Korea, July 2023. doi:10.15607/RSS.2023.XIX.025.
- 364 [2] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess,
365 A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman,
366 A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal,
367 L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao,
368 K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut,
369 H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao,
370 P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web
371 knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.
- 372 [3] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang. Language-
373 driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023.
- 374 [4] C. Lynch and P. Sermanet. Language conditioned imitation learning over unstructured data.
375 *arXiv preprint arXiv:2005.07648*, 2020.
- 376 [5] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence.
377 Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*,
378 2023.
- 379 [6] Y. Cui, S. Karamcheti, R. Palleti, N. Shivakumar, P. Liang, and D. Sadigh. No, to the right:
380 Online language corrections for robotic manipulation via shared autonomy. In *Proceedings
381 of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pages 93–101,
382 2023.
- 383 [7] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu,
384 and L. Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint
385 arXiv:2210.03094*, 2022.
- 386 [8] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z:
387 Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learn-
388 ing*, pages 991–1002. PMLR, 2022.
- 389 [9] M. Reuss, M. Li, X. Jia, and R. Lioutikov. Goal-conditioned imitation learning using score-
390 based diffusion policies. *Robotics: Science and Systems (RSS)*, 2023.
- 391 [10] Y. Ding, C. Florensa, P. Abbeel, and M. Phielipp. Goal-conditioned imitation learning. *Ad-
392 vances in neural information processing systems*, 32, 2019.
- 393 [11] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from
394 demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.
- 395 [12] L. Manuelli, W. Gao, P. Florence, and R. Tedrake. kpam: Keypoint affordances for category-
396 level robotic manipulation. In *The International Symposium of Robotics Research*, pages 132–
397 157. Springer, 2019.
- 398 [13] K. Zakka, A. Zeng, J. Lee, and S. Song. Form2fit: Learning shape priors for generalizable
399 assembly from disassembly. In *2020 IEEE International Conference on Robotics and Automa-
400 tion (ICRA)*, pages 9404–9410. IEEE, 2020.

- 401 [14] A. Ganapathi, P. Sundaresan, B. Thananjeyan, A. Balakrishna, D. Seita, J. Grannen, M. Hwang,
402 R. Hoque, J. E. Gonzalez, N. Jamali, et al. Learning dense visual correspondences in simulation
403 to smooth and fold real fabrics. In *2021 IEEE International Conference on Robotics and*
404 *Automation (ICRA)*, pages 11515–11522. IEEE, 2021.
- 405 [15] M. Danielczuk, A. Kurenkov, A. Balakrishna, M. Matl, D. Wang, R. Martín-Martín, A. Garg,
406 S. Savarese, and K. Goldberg. Mechanical search: Multi-step retrieval of a target object oc-
407 cluded by clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, pages
408 1614–1621. IEEE, 2019.
- 409 [16] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg. Concept2robot: Learning manip-
410 ulation concepts from instructions and human demonstrations. In *Proceedings of Robotics:*
411 *Science and Systems (RSS)*, 2020.
- 412 [17] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured predic-
413 tion to no-regret online learning. In *Proceedings of the fourteenth international conference*
414 *on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Pro-
415 ceedings, 2011.
- 416 [18] K. Burns, T. Yu, C. Finn, and K. Hausman. Robust manipulation with spatial features. In *CoRL*
417 *2022 Workshop on Pre-training Robot Learning*, 2022.
- 418 [19] S. Belkhale, Y. Cui, and D. Sadigh. Data quality in imitation learning. *arXiv preprint*
419 *arXiv:2306.02437*, 2023.
- 420 [20] C. M. Barber, R. J. Shucksmith, B. MacDonald, and B. C. Wünsche. Sketch-based robot
421 programming. In *2010 25th International Conference of Image and Vision Computing New*
422 *Zealand*, pages 1–8. IEEE, 2010.
- 423 [21] D. Porfirio, L. Stegner, M. Cakmak, A. Sauppé, A. Albarghouthi, and B. Mutlu. Sketching
424 robot programs on the fly. In *Proceedings of the 2023 ACM/IEEE International Conference*
425 *on Human-Robot Interaction, HRI '23*, page 584–593, New York, NY, USA, 2023. Associa-
426 tion for Computing Machinery. ISBN 9781450399647. doi:10.1145/3568162.3576991. URL
427 <https://doi.org/10.1145/3568162.3576991>.
- 428 [22] Y. Cui, S. Niekum, A. Gupta, V. Kumar, and A. Rajeswaran. Can foundation models perform
429 zero-shot task specification for robot manipulation? In *Learning for Dynamics and Control*
430 *Conference*, pages 893–905. PMLR, 2022.
- 431 [23] J. Gu, S. Kirmani, P. Wohlhart, Y. Lu, M. G. Arenas, K. Rao, W. Yu, C. Fu, K. Gopalakrishnan,
432 Z. Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *arXiv*
433 *preprint arXiv:2311.01977*, 2023.
- 434 [24] W. Zhi, T. Zhang, and M. Johnson-Roberson. Learning from demonstration via probabilistic
435 diagrammatic teaching. *arXiv preprint arXiv:2309.03835*, 2023.
- 436 [25] S. Masnadi, J. J. LaViola Jr, X. Zhu, K. Desingh, and O. C. Jenkins. A sketch-based system
437 for human-guided constrained object manipulation. *arXiv preprint arXiv:1911.07340*, 2019.
- 438 [26] P. N. Chowdhury, A. K. Bhunia, A. Sain, S. Koley, T. Xiang, and Y.-Z. Song. What can human
439 sketches do for object detection? In *Proceedings of the IEEE/CVF Conference on Computer*
440 *Vision and Pattern Recognition*, pages 15083–15094, 2023.
- 441 [27] A. K. Bhunia, S. Koley, A. Kumar, A. Sain, P. N. Chowdhury, T. Xiang, and Y.-Z. Song.
442 Sketch2saliency: Learning to detect salient objects from human drawings. In *Proceedings of*
443 *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2733–2743,
444 2023.
- 445 [28] A. K. Bhunia, V. R. Gajjala, S. Koley, R. Kundu, A. Sain, T. Xiang, and Y.-Z. Song. Doodle
446 it yourself: Class incremental learning by drawing a few sketches. In *Proceedings of the*
447 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2293–2302, 2022.

- 448 [29] S. Qiu, S. Xie, L. Fan, T. Gao, J. Joo, S.-C. Zhu, and Y. Zhu. Emergent graphical conventions
449 in a visual communication game. *Advances in Neural Information Processing Systems*, 35:
450 13119–13131, 2022.
- 451 [30] Z. Lei, Y. Zhang, Y. Xiong, and S. Chen. Emergent communication in interactive sketch
452 question answering. *arXiv preprint arXiv:2310.15597*, 2023.
- 453 [31] P. N. Chowdhury, A. K. Bhunia, A. Sain, S. Koley, T. Xiang, and Y.-Z. Song. Scenetri-
454 logy: On human scene-sketch and its complementarity with photo and text. In *Proceedings of
455 the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10972–10983,
456 2023.
- 457 [32] L. Zhang and M. Agrawala. Adding conditional control to text-to-image diffusion models.
458 *arXiv preprint arXiv:2302.05543*, 2023.
- 459 [33] S. Koley, A. K. Bhunia, A. Sain, P. N. Chowdhury, T. Xiang, and Y.-Z. Song. Picture
460 that sketch: Photorealistic image generation from abstract sketches. In *Proceedings of the
461 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6850–6861, 2023.
- 462 [34] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing
463 instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
464 Recognition*, pages 18392–18402, 2023.
- 465 [35] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image syn-
466 thesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer
467 Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.
- 468 [36] Y. Vinker, E. Pajouheshgar, J. Y. Bo, R. C. Bachmann, A. H. Bermanno, D. Cohen-Or, A. Zamir,
469 and A. Shamir. Clipasso: Semantically-aware object sketching. *ACM Transactions on
470 Graphics (TOG)*, 41(4):1–11, 2022.
- 471 [37] Y. Vinker, Y. Alaluf, D. Cohen-Or, and A. Shamir. Clipascene: Scene sketching with different
472 types and levels of abstraction. *arXiv preprint arXiv:2211.17256*, 2022.
- 473 [38] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional
474 adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern
475 recognition*, pages 1125–1134, 2017.
- 476 [39] M. Li, Z. Lin, R. Mech, E. Yumer, and D. Ramanan. Photo-sketching: Inferring contour
477 drawings from images. In *2019 IEEE Winter Conference on Applications of Computer Vision
478 (WACV)*, pages 1403–1412. IEEE, 2019.
- 479 [40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polo-
480 sukhin. Attention is all you need. *Advances in neural information processing systems*, 30,
481 2017.
- 482 [41] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks.
483 In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.
- 484 [42] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin,
485 P. Abbeel, and W. Zaremba. Hindsight experience replay. In *31st Conference on Neural
486 Information Processing Systems (NIPS 2017)*, 2017.
- 487 [43] I. Sobel. An isotropic 3x3 image gradient operator. *Presentation at Stanford A.I. Project 1968*,
488 1968.
- 489 [44] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with
490 a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*,
491 volume 32, 2018.
- 492 [45] M. Ryoo, A. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova. Tokenlearner: Adaptive
493 space-time tokenization for videos. *Advances in Neural Information Processing Systems*, 34:
494 12786–12797, 2021.

- 495 [46] F. Torabi, G. Warnell, and P. Stone. Behavioral cloning from observation. *arXiv preprint*
496 *arXiv:1805.01954*, 2018.
- 497 [47] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell,
498 P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervi-
499 sion. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- 500 [48] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 1932.

501 **A Additional Evaluations**

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

505 **A.1 Experiments At A Glance**

506 Cumulatively, our results encompass the following: H1 experiments comprise 270 rollouts (6 skills
 507 x 15 trials x 3 methods), H2 comprises 40 rollouts (2 skills x 5 trials x 4 sketch types), H3 comprises
 508 30 rollouts (15 trials x 2 methods), and H4 comprises 30 rollouts (15 trials x 2 methods). All rollouts
 509 are cumulatively evaluated across 62 labelers (split across H1-4).

510 **A.2 Robustness to Input Sketches**

511 To test whether RT-Sketch generalizes to sketches drawn by different individu-
 512 als, we collect 30 *line sketches* (drawn via tracing) by 6 different annotators (whose
 513 sketches were never seen during training) on 5 trials of the *move near* scenario.
 514 We obtain the resulting rollouts produced by RT-Sketch with these sketches as input. Across ratings,
 515 RT-Sketch achieves high spatial alignment on sketches drawn by other annotators. Notably, the
 516 performance between sketches drawn by different annotators is similar, as well as the average across
 517 annotators compared to original policy performance on our original sketches (Fig. 4).
 518
 519
 520
 521

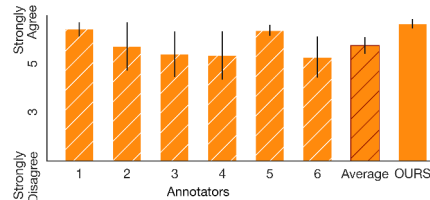


Figure 4: **Sketches Drawn by Other Annotators**

522 **A.3 Multimodal Goal Specification: Sketches + Language**

523 We train a sketch-and-language conditioned model by modifying the RT-1 architecture to use FiLM
 524 along with EfficientNet layers to tokenize both visual input and language, and concatenate them at
 525 the input. In H1 experiments (Fig. 3), we evaluate all policies on the *upright* skill, where the robot
 526 must place a can or bottle from a sideways orientation initially to an upright orientation at a desired
 527 location on the table. While RT-1 typically can reorient the can/bottle properly, it struggles to place
 528 the item in the intended location on the table, as reflected in this policy’s spatial imprecision in
 529 Table 1. Meanwhile, RT-Sketch struggles to reorient the can/bottle, since an imperfect sketch may
 530 fail to specify the exact desired orientation, but often places the can/bottle in the desired location.
 531 In Fig. 5, we see that while language alone (i.e. ”place the can upright”) can be ambiguous in terms
 532 of spatial placement, and a sketch alone does not encourage reorientation, we empirically see that
 533 the joint policy is better able to address the limitations of either modality alone. A similar pattern
 emerges for *pick drawer* (Fig. 5).

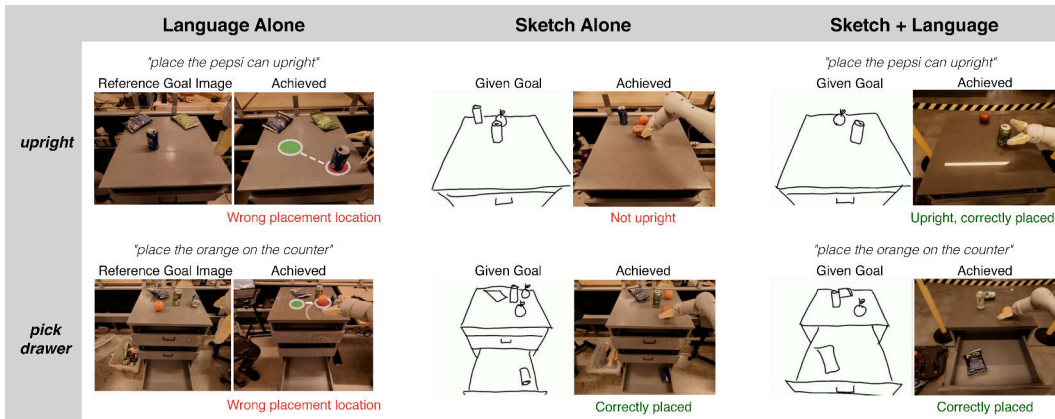


Figure 5: **Multimodal Goal Specification: Sketch+Language:** Empirically, we find that while a language-only policy can struggle with spatial precision, and a sketch-only policy can fail to interpret intended object orientations from a sketch alone, a multimodal policy is better able to address the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

535 **B Additional Results: Goal Alignment**

536 In addition to the goal alignment results reported in Fig. 3 which are based on average Likert ratings,
 537 we additionally conduct a non-parametric Mann-Whitney U (MWU) test with $\alpha = 0.05$ for H1-4

538 to evaluate the differences in goal alignment ratings across methods. This kind of statistical test is
 539 suitable for ordinal data and does not make specific assumptions on the normality or variance of the
 540 data distributions.

541 B.1 H1 Findings

542 The H1 experiments aim to evaluate how RT-Sketch compares to RT-1 and RT-Goal-Image on the
 543 standard RT-1 tabletop manipulation benchmark [1]. We conduct a MWU test under the null hy-
 544 pothesis that there is no difference in the goal alignment ratings from labelers across the methods.
 545 In Appendix Table 3 and Appendix Table 4, we report the pairs of methods for which the ratings
 546 yield a p-value of < 0.05 , rejecting the null hypothesis, along with their U -statistic.

Table 3: 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 4: 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}

547 We conclude that for 5 of 6 and 4 of 6 skills, the null hypothesis is confirmed for semantic and spa-
 548 tial alignment ratings, respectively, suggesting that there is no dropoff in performance with sketches
 549 compared to traditional modalities. We do observe that for the *upright* skill, the rating difference
 550 between RT-Sketch and RT-1 is significant, and RT-Sketch suffers a slight performance drop as re-
 551 orientation is particularly difficult to infer from a sketch alone. However, we have since addresses
 552 this challenge with a policy conditioned on both sketches and language, which performs reorienta-
 553 tion better than sketches-alone and with more spatial precision than language-alone (Section 4.2).

554 The highlighted rows above indicate when the goal alignment ratings for RT-Sketch compared to
 555 either RT-1 or RT-Goal-Image were found to be statistically significant. Notably, there are very few
 556 such findings, in alignment with H1. This is in accordance with what we observe Fig. 3: nearly
 557 no noticeable difference in performance between methods for most of the skills, and the slightly
 558 better performance of RT-1 compared to RT-Sketch (and the slightly better performance of RT-
 559 Sketch compared to RT-Goal-Image) for the *upright* skill.

Table 5: 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 6: **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	-	-

560 B.2 H2 Findings

561 For H2 experiments, we evaluate RT-Sketch’s robustness to the input specificity of the sketch. We
 562 find that across the 4 sketch types, the only pairings which garner statistically significant differences
 563 in ratings are free-hand sketches as compared to other types (Appendix Table 5 and Appendix Ta-
 564 ble 6). This is natural given the drastic perspective and geometric differences of free-hand sketches
 565 compared to those which are *traced* or derived from a transform of the goal image itself (edge
 566 detection).

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

Table 7: **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 8: **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}

571 B.3 H3 and H4 Findings

572 Finally, we conduct a MWU test over the semantic/spatial goal alignment ratings between RT-
 573 Sketch and RT-Goal-Image in the setting of visual distractors (H3, Appendix Table 7) as well as
 574 RT-Sketch and RT-1 in the setting of language ambiguity (H4, Appendix Table 8). We hypothe-
 575 size that RT-Sketch does indeed achieve *higher* ratings than baselines in these settings, as sketches
 576 are by nature 1) minimal, which may enable emergent robustness to distractors, and 2) agnostic to
 577 language.

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

580 B.4 Summary of Mann-Whitney U Findings

581 In short, the additional findings from conducting more thorough MWU testing over H1-4 align very
 582 closely with what we observe and report in Fig. 3 and suggest the merits of sketches across a range
 583 of scenarios.

584 C Future Directions

585 Learning a policy conditioned on view-invariant sketches can be an initial step before moving to
 586 even more abstract representations like schematics or diagrams for assembly tasks. Additionally,
 587 alternative ways to condition on sketches is a powerful avenue for future work. RT-Sketch currently
 588 only considers goal observations in sketch space, but projecting all observations to a sketch-based

589 or latent space is another underexplored but promising direction. Sketches are not without their
 590 own limitations, however, as ambiguity due to omitted details or poor quality sketches are persistent
 591 challenges. In the future, we are excited to continue exploring multimodal goal specification which
 592 can leverage the benefits of language, sketches, and other modalities to jointly resolve ambiguity
 593 from any single modality alone. This may include both end-to-end approaches that can jointly
 594 condition on multiple modalities, or hierarchical strategies that can leverage the spatial awareness
 595 of sketches and the summarization capabilities of VLMs to supplement ambiguous language with
 596 more informed descriptions derived from visual observations of a sketch. Lastly, exploring what
 597 combination of modalities humans prefer to use when providing goals, and how best they capture
 598 intent, is an important future direction not addressed in this work.

599 D Sketch Goal Representations

600 Since the main bottleneck to training a sketch-to-action policy like RT-Sketch is collecting a dataset
 601 of paired trajectories and goal sketches, we first train an image-to-sketch translation network \mathcal{T}
 602 mapping image observations o_i to sketch representations g_i , discussed in Section 3. To train \mathcal{T} , we
 603 first take a pre-trained network for sketch-to-image translation [39] trained on the ContourDrawing
 604 dataset of paired images and edge-aligned sketches (Fig. 6). This dataset contains $L^{(i)} = 5$ crowd-
 605 sourced sketches per image for 1000 images. By pre-training on this dataset, we hope to embed a
 606 strong prior in \mathcal{T} and accelerate learning on our much smaller dataset. Next, we finetune \mathcal{T} on a
 607 dataset of 500 manually drawn line sketches for RT-1 robot images. We visualize a few examples of
 608 our manually sketched goals in Fig. 7 under ‘Line Drawings’.

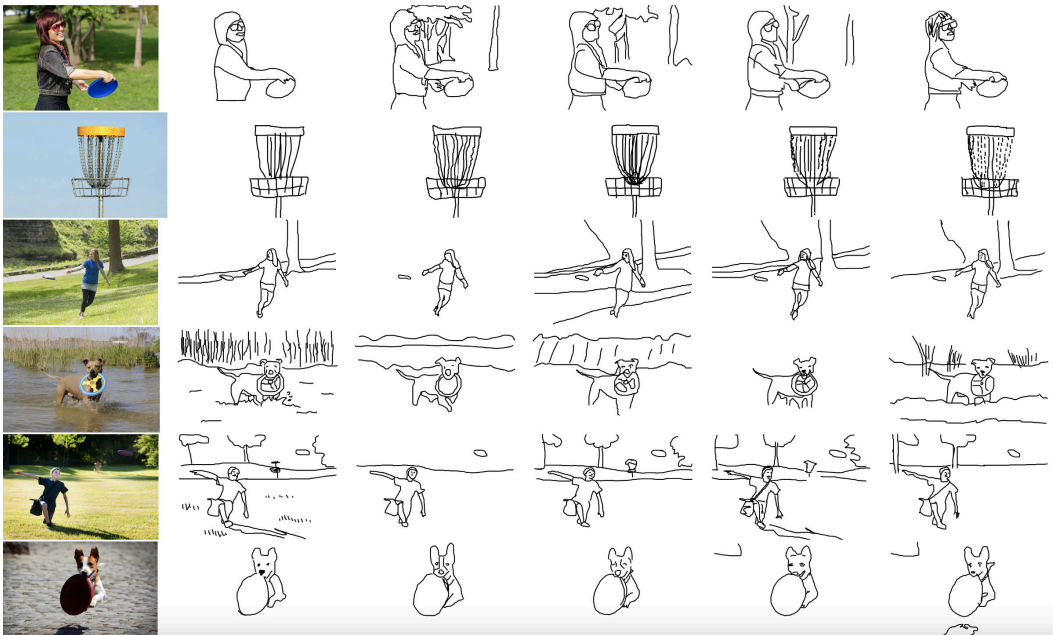


Figure 6: **ContourDrawing Dataset:** We visualize 6 samples from the ContourDrawing Dataset from [39]. For each image, 5 separate annotators provide an edge-aligned sketch of the scene by outlining on top of the original image. As depicted, annotators are encouraged to preserve main contours of the scene, but background details or fine-grained geometric details are often omitted. Li et al. [39] 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.

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

616 Although RT-Sketch is only trained on generated line sketches, colorized line sketches, edge-
 617 detected images, and goal images, we find that it is able to handle sketches of even greater diversity.

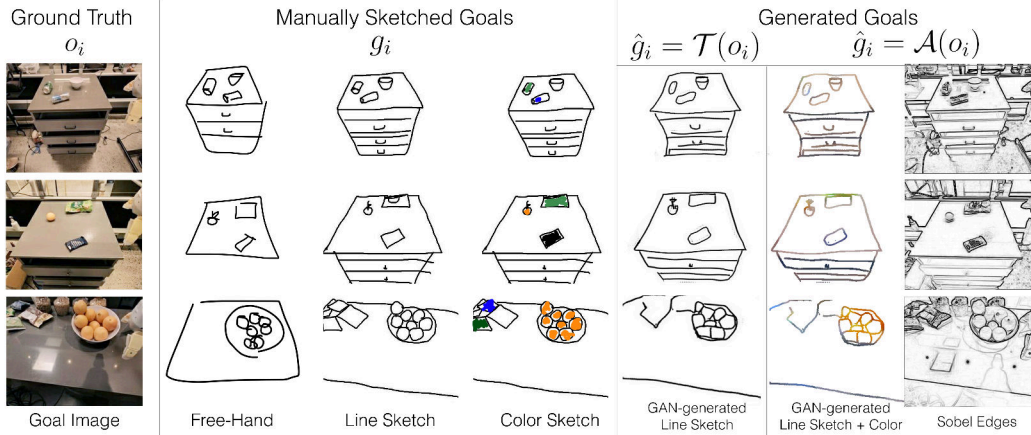


Figure 7: **Visual Goal Diversity**: RT-Sketch is capable of handling a variety of visual goals at both train and test time. RT-Sketch is trained on generated and augmented images like those shown on the right below 'Generated Goals'. But it can also interpret free-hand, line sketches, and colored sketches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

618 This includes non-edge aligned free-hand sketches and sketches with color infills, like those shown
 619 in Fig. 7.

620 D.1 Alternate Image-to-Sketch Techniques

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

623 Recently, two recent works, CLIPAsso [36] and CLIPAScene [37] explore methods for automatically
 624 generating a sketch from an image. These works pose sketch generation as inferring the parameters
 625 of Bezier curves representing "strokes" in order to produce a generated sketch with maximal CLIP-
 626 similarity to a given input image. These methods perform a per-image optimization to generate a
 627 plausible sketch, rather than a global batched operation across many images, limiting their scalabil-
 628 ity. Additionally, they are fundamentally more concerned with producing high-quality, aesthetically
 629 pleasing sketches which capture a lot of extraneous details.



Figure 8: **Alternate Image-to-Sketch Techniques**

630 We, on the other hand, care about producing a minimal but reasonable-quality sketch. The second
 631 technique we explore is trying the pre-trained Photosketching GAN [39] on internet data of paired
 632 images and sketches. However, this model output does not capture object details well, likely due
 633 to not having been trained on robot observations, and contains irrelevant sketch details. Finally, by
 634 finetuning this PhotoSketching GAN on our own data, the outputs are much closer to real, hand-
 635 drawn human sketches that capture salient object details as minimally as possible. We visualize
 636 these differences in Fig. 8.

637 E Evaluation Visualizations

638 To further interpret RT-Sketch's performance, we provide visualizations of the precision metrics
 639 and experimental rollouts. In Fig. 9, we visualize the degree of alignment RT-Sketch achieves,
 640 as quantified by the pixelwise distance of object centroids in achieved vs. given goal images. In
 641 Fig. 10, Fig. 11, Fig. 12, and Fig. 14, we visualize each policy's behavior for H1, H2, H3 and H4,
 642 respectively. Fig. 13 visualizes the four tiers of difficulty in language ambiguity that we analyze for
 643 H4.

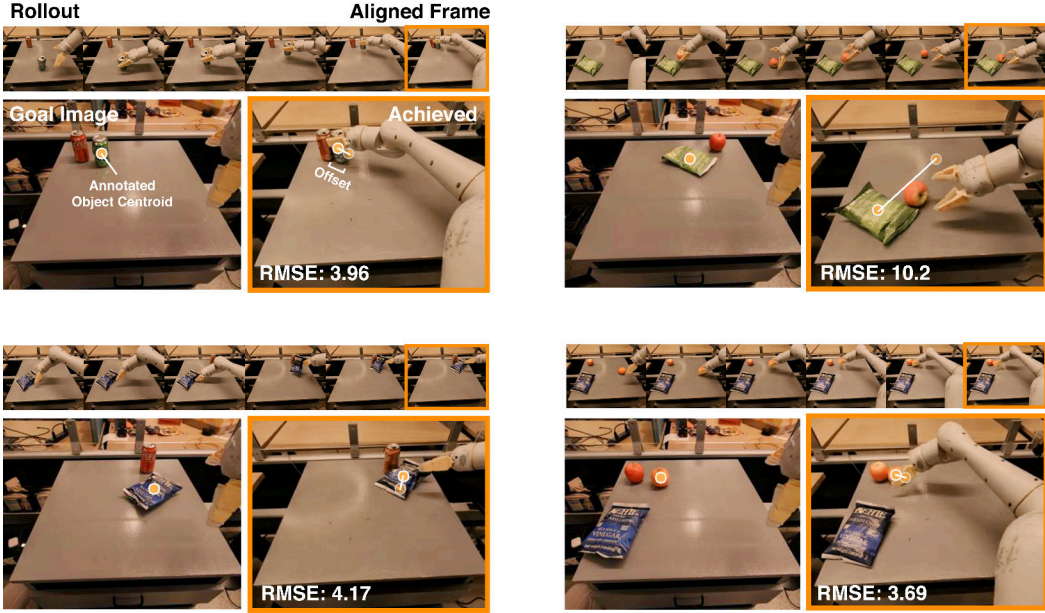


Figure 9: **Spatial Precision Visualization:** We visualize four trials of RT-Sketch on the Move Near skill, along with the measured spatial precision in terms of RMSE. To evaluate spatial precision, we have a human annotator annotate the frame that is visually most aligned, and then keypoints for the 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 1.

644 F RT-Sketch Failure Modes and Limitations

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

648 In Fig. 16, we visualize the failure modes of RT-Sketch. One failure mode we see with RT-Sketch is
 649 occasionally re-trying excessively, as a result of trying to align the scene as closely as possible. For
 650 instance, in the top row, Rollout Image 3, the scene is already well-aligned, but RT-Sketch keeps
 651 shifting the chip bag which causes some misalignment in terms of the chip bag orientation. Still,
 652 this kind of failure is most common with RT-Goal-Image (Table 1), and is not nearly as frequent
 653 for RT-Sketch. We posit that this could be due to the fact that sketches enable high-level spatial
 654 reasoning without over-attending to pixel-level details.

655 One consequence of spatial reasoning at such a high level, though, is an occasional lack of precision.
 656 This is noticeable when RT-Sketch orients items incorrectly (second row) or positions them slightly
 657 off, possibly disturbing other items in the scene (third row). This may be due to the fact that sketches
 658 are inherently imperfect, which makes it difficult to reason with such high precision.

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

667 Finally, the least frequent failure mode is manipulating the wrong object to the wrong target location
 668 (i.e. opening the wrong drawer handle). This is most frequent when the input is a free-hand sketch,
 669 and could be mitigated by increasing sketch detail (Table 2).

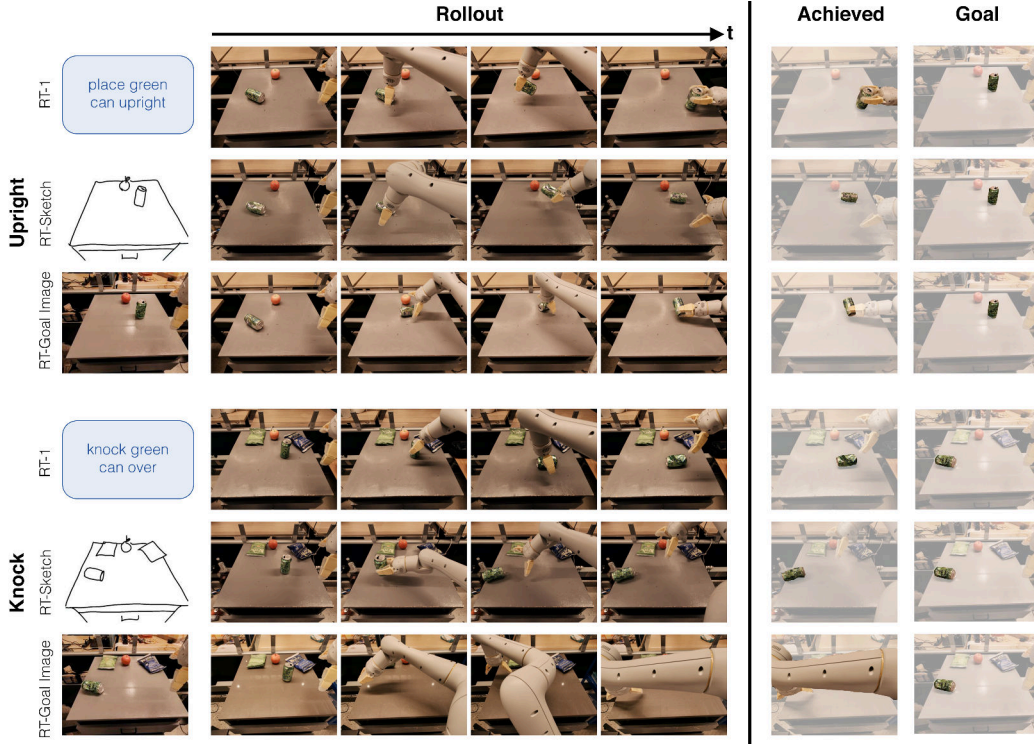


Figure 10: **H1 Rollout Visualization:** We visualize the performance of RT-1, RT-Sketch, and RT-Goal-Image on two skills from the RT-1 benchmark (*upright* and *knock*). For each skill, we visualize the goal provided as input to each policy, along with the policy rollout. We see that for both skills, 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 1).

670 **G Evaluation and Assessment Interfaces**

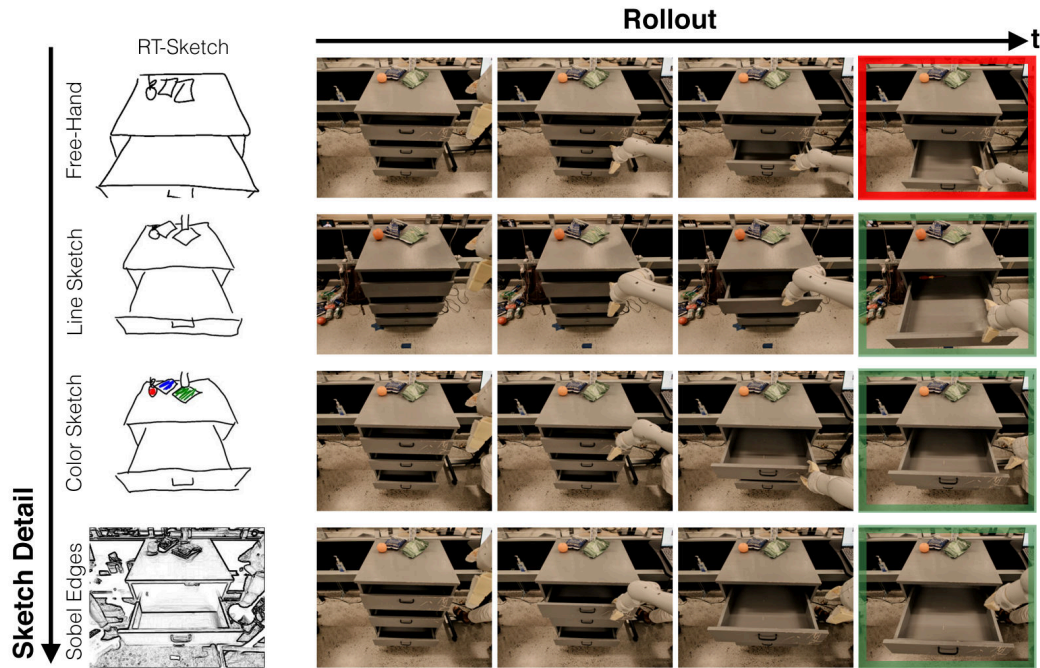


Figure 11: **H2 Rollout Visualization:** For the *open drawer* skill, we visualize four separate rollouts of RT-Sketch operating from different input types. Free-hand sketches are drawn without outlining over the original image, such that they can contain marked perspective differences, partially obscured 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.

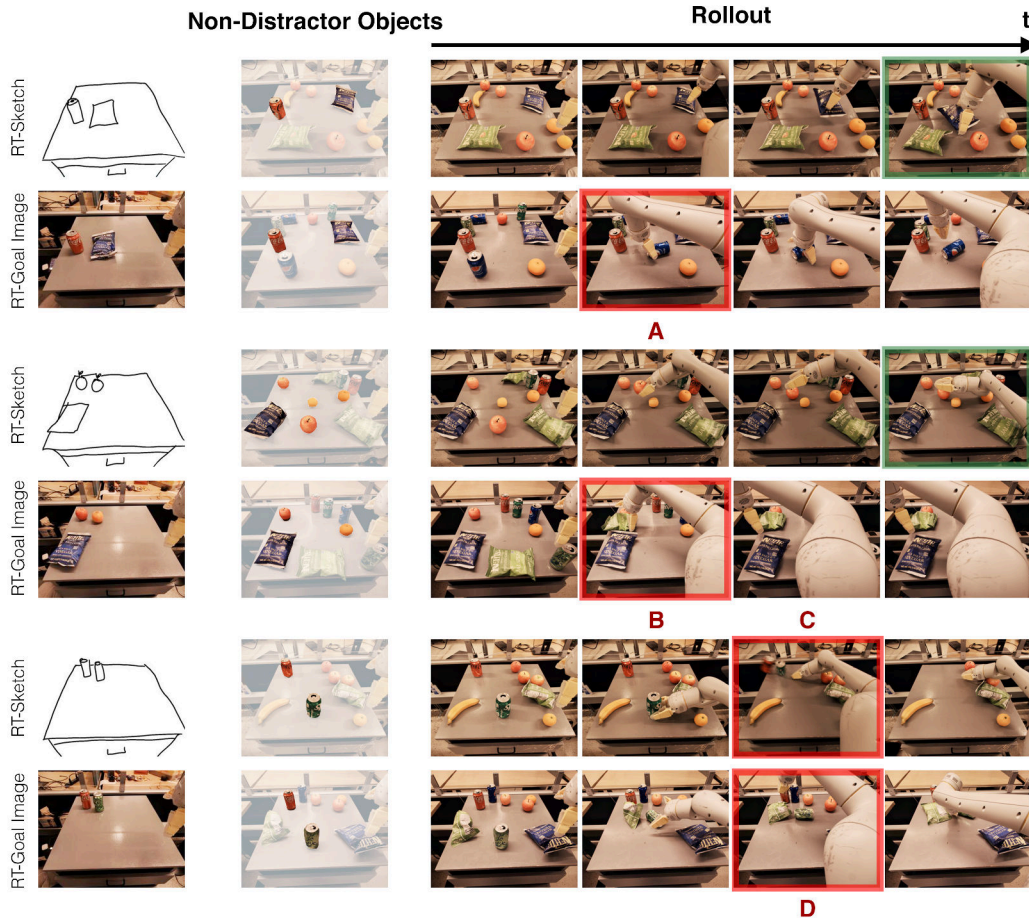


Figure 12: **H3 Rollout Visualization:** We visualize qualitative rollouts for RT-Sketch and RT-Goal-Image for 3 separate trials of the *move near* skill subject to distractor objects. In Column 2, we highlight the relevant non-distractor objects that the policy must manipulate in order to achieve the 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).

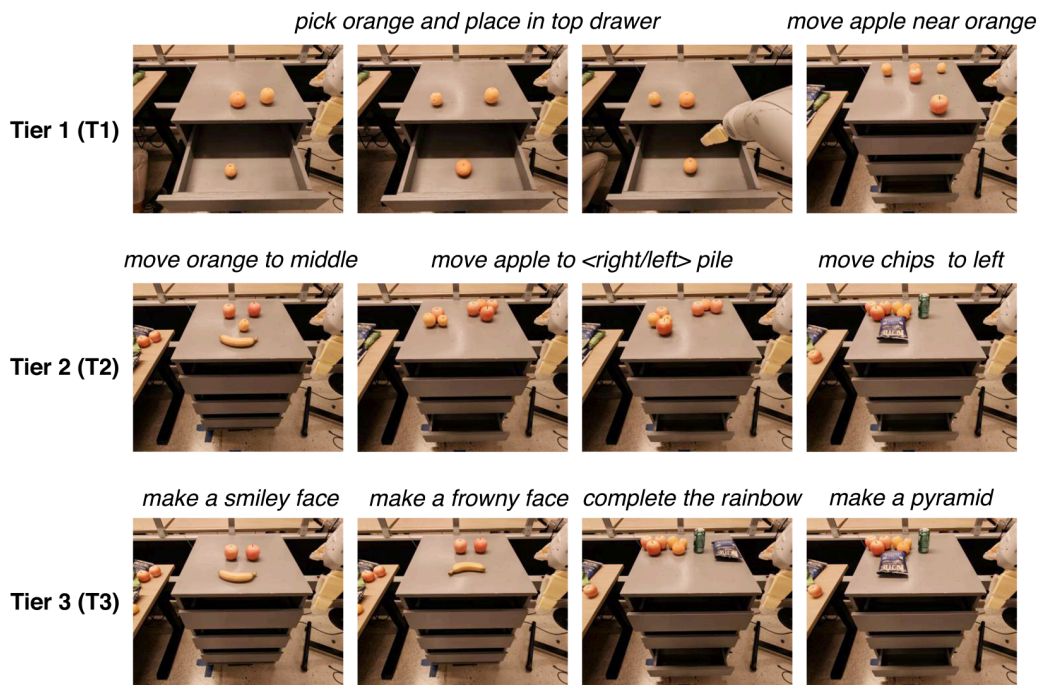


Figure 13: **H4 Tiers of Difficulty**: To test **H4**, we consider language instructions that are either ambiguous due the presence of multiple similar object instances (**T1**), are somewhat out-of-distribution for RT-1 (**T2**), or are far out-of-distribution and difficult to specify concretely without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

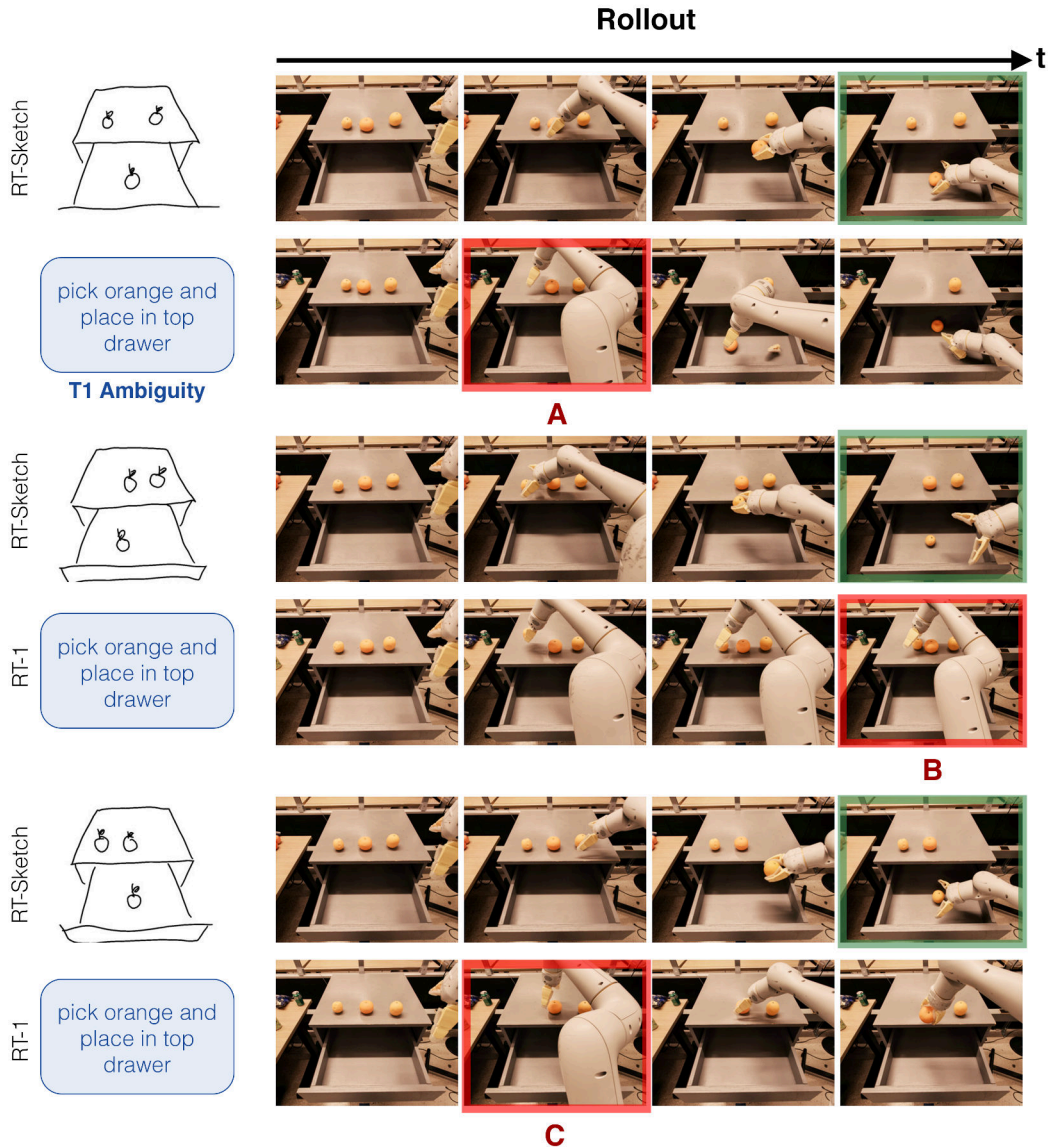


Figure 14: **H4 Rollout Visualization (T1 as visualized in Fig. 13)**: One source of ambiguity in language descriptions is mentioning an object for which there are multiple instances present. For example, we can easily illustrate three different desired placements of an orange in the drawer via a sketch, 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.

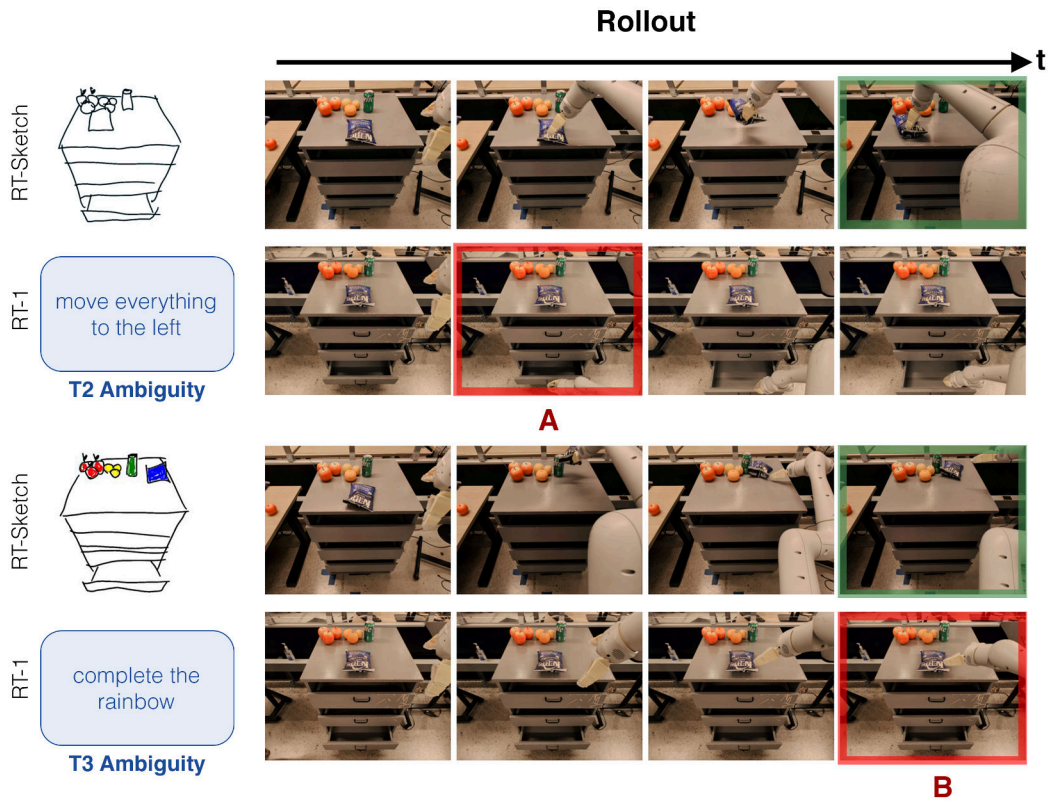


Figure 15: **H4 Rollout Visualization (T2-3 as visualized in Fig. 13)**: For **T2**, we consider language with spatial cues that intuitively should help the policy disambiguate in scenarios like the oranges in Fig. 14. However, we find that RT-1 is not trained to handle such spatial references, and this 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).

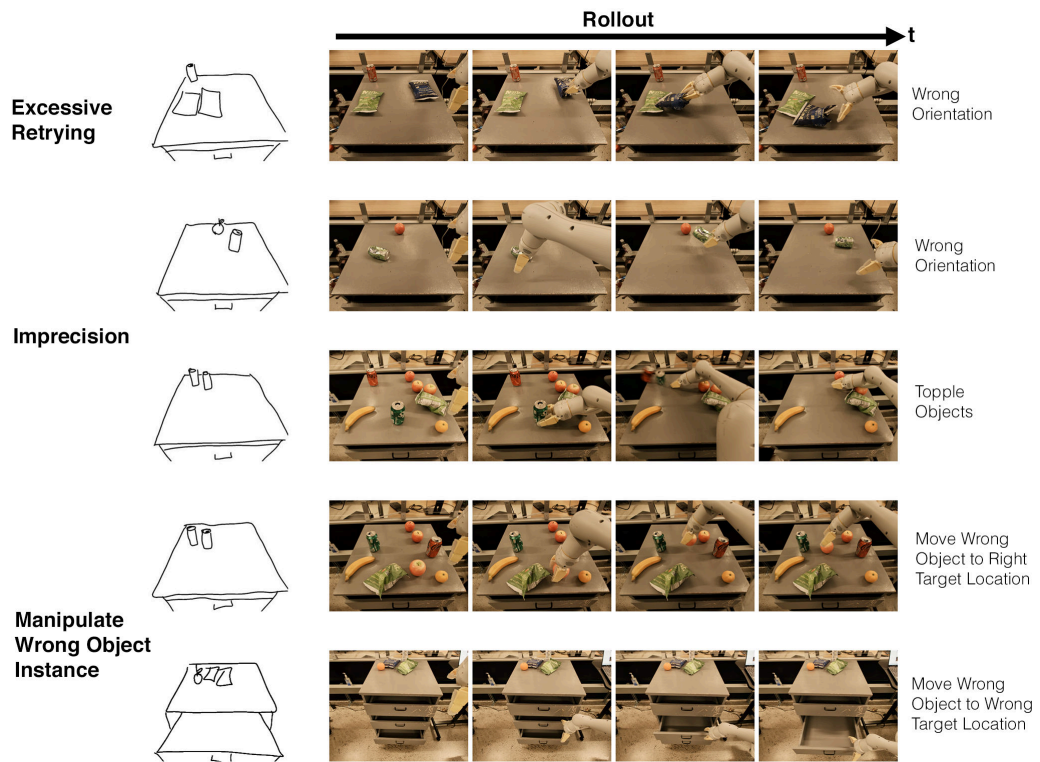


Figure 16: RT-Sketch Failure Modes

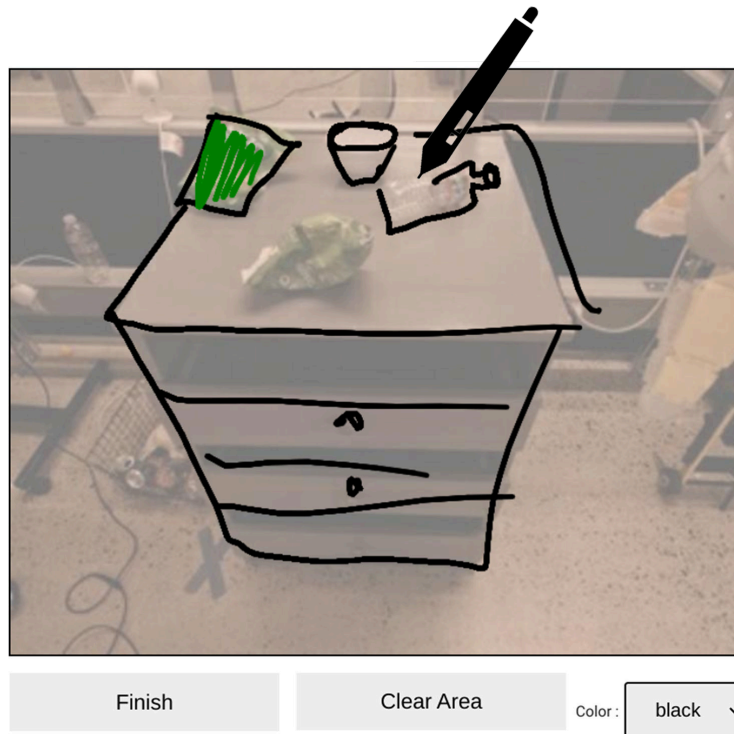



Figure 17: **Sketching UI**: We design a custom sketching interface for manually collecting paired robot images and sketches with which to train \mathcal{T} , and for sketching goals for evaluation. The interface visualizes the current robot observation, and provides the ability to draw on a digital 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

move coke can near green can



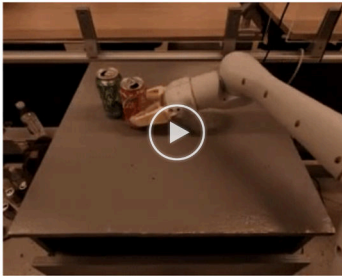

The robot achieves **semantic alignment** with the goal during the rollout. *

1 2 3 4 5 6 7

Strongly Disagree 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 Strongly Agree

Figure 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.