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

Priya Sundaresan[1,3], Quan Vuong[2], Jiayuan Gu[2], Peng Xu[2], Ted Xiao[2], Sean Kirmani[2], Tianhe Yu[2], Michael Stark[3], Ajinkya Jain[3], Karol Hausman[1,2]
Dorsa Sadigh*[1,2], Jeannette Bohg*[1], Stefan Schaal*[3]

*Equal advising, alphabetical order
[1]Stanford University, [2]Google DeepMind, [3][Google] Intrinsic

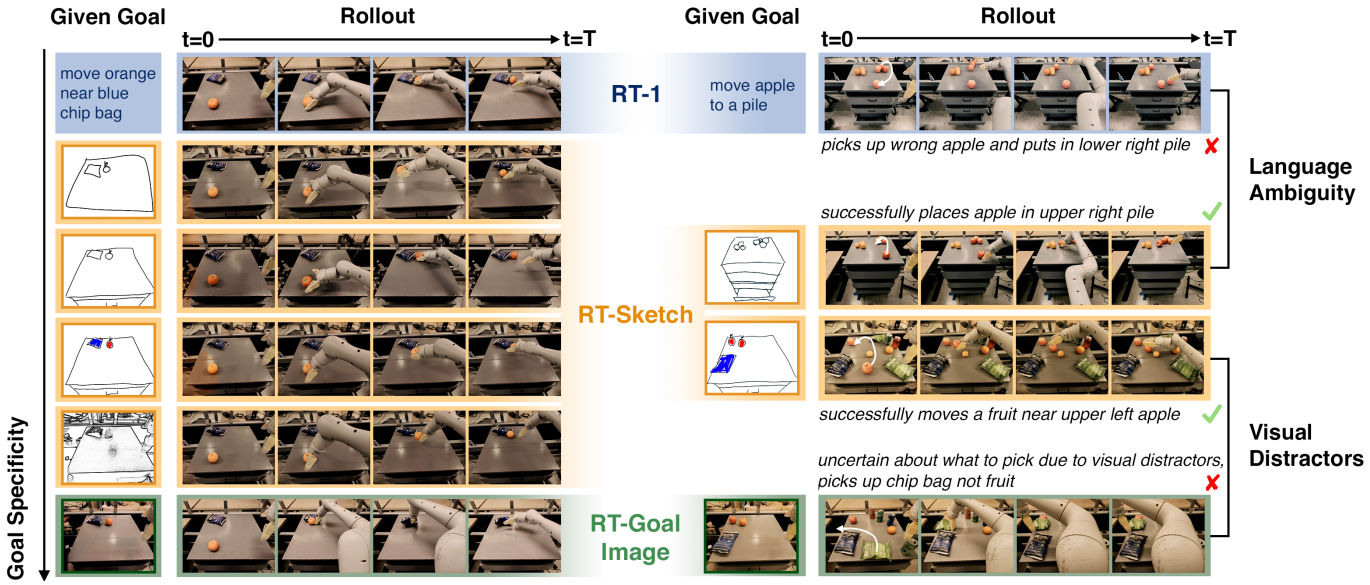

Fig. 1: Rollouts showing RT-Sketch's robustness to sketch detail, ambiguous language, and visual distractors.

***Abstract*—Natural language and images are commonly used as goal representations in goal-conditioned imitation learning (IL). However, natural language can be ambiguous and images can be over-specified. In this work, we study hand-drawn sketches as a modality for goal specification. Sketches are easy for users to provide on the fly like language, but similar to images they can also help a downstream policy to be spatially-aware and even go beyond images to disambiguate task-relevant from task-irrelevant objects. We present RT-Sketch, a goal-conditioned policy for manipulation that takes a hand-drawn sketch of the desired scene as input, and outputs actions. We train RT-Sketch on a dataset of trajectories paired with synthetically generated goal sketches. We evaluate this approach on six manipulation skills involving tabletop object rearrangements on an articulated countertop. Experimentally we find that RT-Sketch is able to perform on a similar level to image or language-conditioned agents in straightforward settings, while achieving greater robustness when language goals are ambiguous or visual distractors are present. Additionally, we show that RT-Sketch has the capacity to interpret and act upon sketches with varied levels of specificity, ranging from minimal line drawings to detailed, colored drawings.**

I. Introduction

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

Instruction-following robots attempt to address this problem using the intuitive interface of natural language commands as inputs to language-conditioned imitation learning policies [7, 8, 22, 27, 28]. For instance, imagine asking a household robot to set the dinner table. A language description such as *"put the utensils, the napkin, and the plate on the table"* is under-specified or ambiguous. It is unclear how exactly the utensils should be positioned relative to the plate or the napkin, or whether their distances to each other matter or not. To achieve this higher level of precision, a user may need to give lengthier descriptions such as *"put the fork 2cm to the right of the plate, and 5cm to the leftmost edge of the table."*, or even online corrections (*"no, you moved too far to the right, move back a bit!"*) [14, 28]. While intuitive, the qualitative nature and ambiguity of language can make it both inconvenient for humans to provide without lengthy instructions or corrections, and for robot policies to interpret for downstream precise manipulation.

Using a goal image (i.e. an image of the scene in its final desired state) to specify objectives and train goal-conditioned imitation learning policies has shown to be quite successful in

recent years, with or without language [20, 21, 34]. However, this has its own shortcomings: access to a goal image is a strong prior assumption, and a pre-recorded goal image is tied to a particular environment, making it difficult to reuse for generalization. To summarize: while natural language is highly flexible, it can also be highly ambiguous or require lengthy descriptions. This quickly becomes difficult in long-horizon tasks or those requiring spatial awareness. Meanwhile, goal images over-specify goals in unnecessary detail, leading to the need for internet-scale data for generalization.

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

In this work, we present RT-Sketch, a goal-conditioned policy for manipulation that takes a hand-drawn sketch of the desired scene as input, and outputs actions. The novel architecture of RT-Sketch modifies the original RT-1 language-to-action Transformer architecture [8] to consume visual goals rather than language, allowing for flexible conditioning on sketches, images, or any other visually representable goals. To enable this, we concatenate a goal sketch and history of observations as input before tokenization, omitting language. We train RT-Sketch on a dataset of 80K trajectories paired with synthetic goal sketches, generated by an image-to-sketch stylization network trained from a few hundred image-sketch pairs.

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

II. Related Work

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

a) Goal-Conditioned Imitation Learning: Reinforcement learning (RL) is not easily applicable in our scenario, as it is nontrivial to define a reward objective which accurately quantifies alignment between a provided scene sketch and states achieved by an agent. We instead focus on IL techniques, particularly the goal-conditioned setting [16]. Goal-conditioned IL has proven useful in settings where a policy needs to handle different variations of the same task [2]. Examples include moving objects into different arrangements [7, 8, 28, 29, 34], kitting [45], folding of deformable objects into different configurations [17], and search for different target objects in clutter [15]. However, these approaches tend to condition on either language [8, 22, 27, 28, 38], or images [15] to specify goals. Follow-up work enabled multimodal conditioning on either goal images and language [20], in-prompt images [21], or image embeddings [17, 29, 45]. All of these representations are ultimately derived from raw images or language, which overlooks the potential for more abstract goals like sketches.

Beyond inflexible goal representations, goal-conditioned IL tends to overfit to demonstration data and fails to handle even slight distribution shifts [36]. For language, this can encompass ambiguous or novel phrasing or unseen objects [8, 20]. Goal-image conditioning is similarly susceptible to out-of-distribution visual shifts, such as lighting variations or unseen object and background appearances [4, 10]. Instead, sketches are minimal enough to combat visual distractors, yet expressive enough to provide unambiguous goals. Prior work, including [3] and [31], have shown the utility of sketches over pure language for navigation and limited manipulation. However, the sketches explored in these works are largely intended to guide low-level motion at the joint-level for manipulation, or provide explicit directional cues for navigation. [13] considers sketches amongst other modalities as an input for goal-conditioned manipulation, but does not explicitly train a policy conditioned on sketches. They thus came to the conclusion that the scene image is better than the sketch at goal specification. Our result is different and complementary, in that policies trained to take sketches as input outperform a scene image conditioned policy, by 1.63x and 1.5x in terms of Likert ratings for perceived spatial and semantic alignment, subject to visual distractors. Gu et al. [18] propose goal-conditioning on *hindsight-trajectory* sketches. Here, sketches represent 2D paths drawn over an image to indicate the intended robot *motion*. While this approach treats sketches as a *motion-centric* representation, the sketches in our work are *scene-centric*, representing the desired visual goal state rather than the desired robot actions.

b) Image-Sketch Conversion: Sketches have been studied within the computer vision community for object detection [5, 6, 11], visual question answering [24, 32], and scene understanding [12], either in isolation or in addition to text and images. When considering how best to incorporate sketches in IL, an important design choice is whether to take sketches into account (1) at test time (by converting a sketch to another

modality compatible with a pre-trained policy), or (2) at train time (by explicitly training a policy conditioned on sketches). For (1), one could first convert a given sketch to a goal image, and then roll out a vanilla goal-image conditioned policy. Existing frameworks tackle sketch-to-image conversion, such as ControlNet [46], GAN-style approaches [23], or text-to-image synthesis, such as InstructPix2Pix [9] or Stable Diffusion [35]. While these models can produce photorealistic visuals, they do not jointly handle image generation and style transfer, making it unlikely for generated images to match the style of agent observations. These approaches are also susceptible to hallucinated artifacts, introducing distribution shifts [46].

Thus, we instead opt for (2), and consider image-to-sketch conversion techniques for hindsight relabeling of demonstrations. Recently, Vinker et al. [43, 44] propose networks for predicting Bezier curve-based sketches of input images, supervised by a CLIP-based alignment metric. While these approaches generate visually compelling sketches, test-time generation takes on the order of minutes, which does not scale to the typical size of robot learning datasets with hundreds to thousands of trajectories. Meanwhile, conditional generative adversarial networks (cGANs) such as Pix2Pix [19] have proven useful for scalable image-to-image translation. Most related to our work is that of Li et al. [25], which trains a Pix2Pix model to produce sketches from given images on a large crowd-sourced dataset of $5K$ paired images and line drawings. We build on this work to fine-tune an image-to-sketch model that maps robot observations to sketches, with which to train an IL policy.

III. Sketch-Conditioned Imitation Learning

Problem Statement We first formalize the problem of learning a manipulation policy conditioned on a goal *sketch* of the desired scene state and a history of interactions. We denote such a policy 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 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 inference time, the policy takes a given goal sketch along with a history of D previous RGB image observations, and outputs an action. To train such a policy, we assume access to a dataset $\mathcal{D}_{\text{sketch}} = \{g^n, \{(o_t^n, a_t^n)\}_{t=1}^{T^{(n)}}\}_{n=1}^{N}$ of N successful demonstrations, where $T^{(n)}$ refers to the length of the n^{th} trajectory in timesteps. Each episode of the dataset consists of a given goal sketch and a corresponding demonstration trajectory, with images recorded at each timestep. Our goal is to thus learn the sketch-conditioned imitation policy $\pi_{\text{sketch}}(a_t|g, \{o_j\}_{j=1}^{t})$ trained on $\mathcal{D}_{\text{sketch}}$.

A. Image-to-Sketch Translation

Training a sketch-conditioned policy requires a dataset of robot trajectories, each paired with a goal sketch. Collecting both demonstration trajectories and manually drawn sketches at scale is impractical. Thus, we instead aim to learn an image-to-sketch translation network $\mathcal{T}(g|o)$ that takes an image observation o and outputs the corresponding goal sketch g. This network can be used to post-process an existing dataset of demonstrations $\mathcal{D} = \{\{(o_t^n, a_t^n)\}_{t=1}^{T^{(n)}}\}_{n=1}^{N}$ with image observations by appending a synthetically generated goal sketch to each demonstration. This produces a 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 large-scale dataset of VR-teleoperated robot demonstrations from prior work, which included skills such as object pick and place, placing cans and bottles upright or sideways, and opening and closing cabinets [8]. Prior work previously trained a language-conditioned IL policy RT-1 from this data, but we extend this policy architecture to accommodate sketches, detailed in Section III-B.

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

b) Sketch Dataset Generation: To train an image-to-sketch translation network $\mathcal{T}$, we collect a new dataset $\mathcal{D}_{\mathcal{T}} = \{(o_i, g_i^1, \ldots, g_i^{L^{(i)}})\}_{i=1}^{M}$ consisting of M image observations o_i each paired with a set of goal sketches $g_i^1, \ldots, g_i^{L^{(i)}}$. Those represent $L^{(i)}$ *different* representations of the same image o_i, in order to account for the fact that there are multiple, valid ways of sketching the same scene. To collect $\mathcal{D}_{\mathcal{T}}$, we take 500 randomly sampled terminal images from demonstration trajectories in the RT-1 dataset, and manually draw sketches with black lines on a white background capturing the tabletop, drawers, and relevant objects visible on the table. While we personally annotate each robot observation with just one single sketch, we add this data to an existing, much larger non-robotic dataset of paired images and sketches [25]. This dataset captures inter-sketch variation via multiple crowdsourced sketches per image. We do not include the robot arm in our manual sketches, as we find a minimal representation to be most natural. Empirically, we find that our policy can handle such sketches despite actual goal configurations likely having the arm in view. We collect these drawings using a custom digital stylus drawing interface where user draws an edge-aligned sketch over the original image (Appendix Fig. 17) by *tracing outlines*. The final recorded sketch includes the user's strokes in black on a white canvas.

c) Image-to-Sketch Training: We implement the image-to-sketch translation network $\mathcal{T}$ with the Pix2Pix conditional generative adversarial network (cGAN) architecture, which is composed of a generator $G_{\mathcal{T}}$ and a discriminator $D_{\mathcal{T}}$ [19]. The generator $G_{\mathcal{T}}$ takes an input image o, a random noise vector z, and outputs a goal sketch g. The discriminator $D_{\mathcal{T}}$ is trained to discriminate amongst artificially generated versus ground

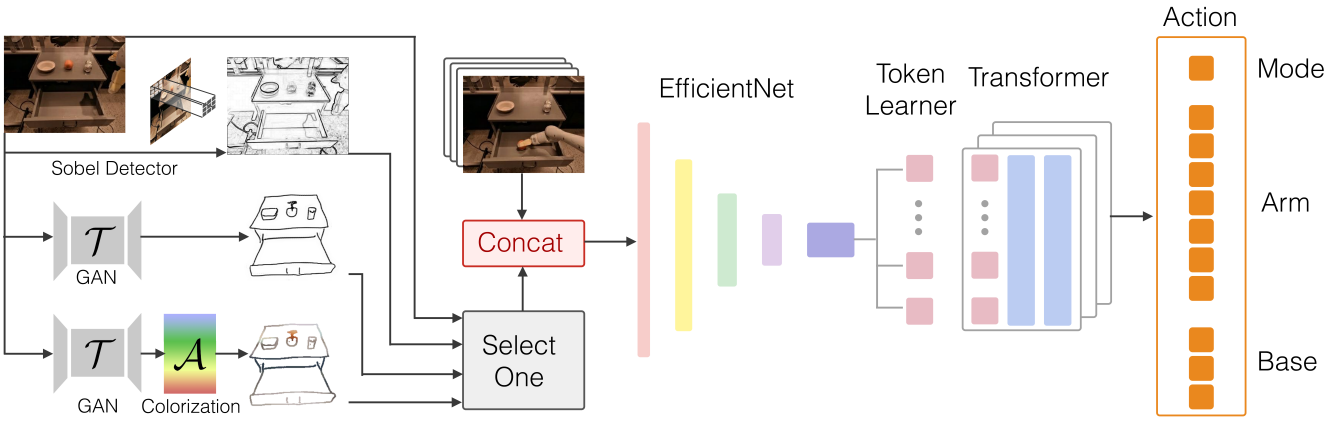

Fig. 2: Architecture of RT-Sketch allowing different kinds of visual input. RT-Sketch adopts the Transformer [42] architecture with EfficientNet [40] tokenization at the input, and outputs bucketized actions.

truth sketches. We utilize the standard cGAN supervision loss to train both [19, 25]:

$$\mathcal{L}_{\text{cGAN}} = \min_{G_\mathcal{T}} \max_{D_\mathcal{T}} \mathbb{E}_{o,g}[\log D_\mathcal{T}(o,g)] + \mathbb{E}_{o,g}[\log(1 - D_\mathcal{T}(o, G_\mathcal{T}(o,g))] \tag{1}$$

We also add the $\mathcal{L}_1$ loss to encourage the produced sketches to align with ground truth sketches as in [25]. To account for the fact that there may be multiple valid sketches for a given image, we only penalize the minimum $\mathcal{L}_1$ loss incurred across all $L^{(i)}$ sketches provided for a given image as in Li et al. [25]. This is to prevent wrongly penalizing $\mathcal{T}$ for producing a valid sketch that aligns well with one example but not another simply due to stylistic differences in the ground truth sketches. The final objective is a λ-weighted combination of the average cGAN loss and the minimum alignment loss:

$$\mathcal{L}_\mathcal{T} = \frac{\lambda}{L^{(i)}} \sum_{k=1}^{L^{(i)}} \mathcal{L}_{\text{cGAN}}(o_i, g_i^{(k)}) + \min_{k \in \{1,\ldots,L^{(i)}\}} \mathcal{L}_1(o_i, g_i^{(k)}) \tag{2}$$

In practice, we supplement the 500 manually drawn sketches from $\mathcal{D}_\mathcal{T}$ by leveraging the existing larger-scale Contour Drawing Dataset [25]. We refer to this dataset as $\mathcal{D}_{\text{CD}}$, which contains 1000 examples of internet-scraped images containing objects, people, animals from Adobe Stock, paired with $L^{(i)} = 5$ crowd-sourced black and white outline drawings per image collected on Amazon Mechanical Turk (see Appendix Fig. 6 for examples). We first take a pre-trained image-to-sketch translation network $\mathcal{T}_{\text{CD}}$ [25] trained on $\mathcal{D}_{\text{CD}}$, with $L^{(i)} = 5$ sketches per image. Then, we fine-tune $\mathcal{T}_{\text{CD}}$ on $\mathcal{D}_\mathcal{T}$, with only $L^{(i)} = 1$ manually drawn sketch per robot observation, to obtain our final image-to-sketch network $\mathcal{T}$. Visualizations of sketches generated by $\mathcal{T}$ are available in Fig. 7.

B. RT-Sketch

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

a) 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 episodes with a paired natural language instruction i and demonstration trajectory $\{(o_t^n, a_t^n)\}_{t=1}^{T^n}$. We can automatically hindsight-relabel such a dataset with goal images instead of language goals [1]. Let us denote the last step of a trajectory n as $T^{(n)}$. Then the new dataset with image 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 observation of the trajectory $o_{T^{(n)}}^n$ as the goal g^n. To produce a dataset for π_{sketch}, we can simply replace $o_{T^{(n)}}^n$ with $\hat{g}^n = \mathcal{T}(o_{T^{(n)}}^n)$ such that $\mathcal{D}_{\text{sketch}} = \{\hat{g}^n, \{(o_t^n, a_t^n)\}_{t=1}^{T^{(n)}}\}_{n=1}^N$.

To encourage the policy to afford different levels of input sketch specificity, we in practice produce goals by $\hat{g}^n = \mathcal{A}(o_{T^{(n)}}^n)$, where $\mathcal{A}$ is a randomized augmentation function. $\mathcal{A}$ chooses between simply applying $\mathcal{T}$, $\mathcal{T}$ with colorization during postprocessing (e.g., superimposing a blurred version of the ground truth RGB image over the binary sketch), a Sobel operator [39] for edge detection, or an identity operation, which preserves the original image (Fig. 2). By co-training on all representations, we intend for RT-Sketch to handle a spectrum of specificity going from binary sketches; colorized sketches; edge detected images; and goal images (Appendix Fig. 7).

b) RT-Sketch Model Architecture: In our setting, we consider goals provided as sketches rather than language as was done in RT-1. The original RT-1 policy relies on a Transformer architecture backbone [42]. RT-1 first passes a history of $D = 6$ images through an EfficientNet-B3 model [40] producing image embeddings, which are tokenized, and separately extracts textual embeddings and tokens via FiLM [30] and a Token Learner [37]. The tokens are then fed into a Transformer which outputs bucketed actions: a 7-DoF output for the end-effector (x, y, z, roll, pitch, yaw, gripper width), 3-DoF for the mobile base, (x, y, yaw), and 1 mode-switching flag (base movement, arm movement, and termination). To accommodate our change in the input, we omit the FiLM language tokenization altogether. Instead, we

concatenate a given visual goal with the history of images as input to EfficientNet, and extract tokens from its output, leaving the rest of the policy architecture unchanged. We train two policies using this architecture (Fig. 2): RT-Sketch refers to our policy trained from sketches, and RT-Goal-Image is a baseline policy trained from goal images.

c) Training RT-Sketch: We now train π_{sketch} on $\mathcal{D}_{\pi_{\text{sketch}}}$ from scratch (rather than finetuning an existing backbone) using the same procedure as in RT-1 [8], with the above architectural changes. We fit the policy using the behavioral cloning objective that minimizes the negative log-likelihood of an action [41]: $J(\pi_{\text{sketch}}) = \sum_{n=1}^{N} \sum_{t=1}^{T^{(n)}} \log \pi_{\text{sketch}}(a_t^n | g^n, \{o_j\}_{j=1}^t)$

IV. Experiments

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

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

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

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

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

A. Experimental Setup

a) Policies: We compare RT-Sketch to the original language-conditioned agent RT-1 [8], and a goal image-conditioned agent RT-Goal-Image. All policies are trained on a multi-task dataset of $\sim$ 80K real-world trajectories manually collected via VR teleoperation using the setup from Brohan et al. [8]. These trajectories span 6 common household object rearrangement tasks: *move X near Y*, *place X upright*, *knock X over*, *open the X drawer*, *close the X drawer*, and *pick X from Y*.

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

c) Metrics: Defining a standardized evaluation protocol for goal alignment is non-trivial when binary task success is too coarse and image-similarity metrics like CLIP [33] can be brittle. We first measure performance by quantifying the policy precision as the pixel distance between object centroids in achieved and ground truth goal states, using manual keypoint annotations (see Fig. 9 in Appendix for examples). Although leveraging out-of-the box object detectors to detect object centroids is a possibility, we want to avoid conflating errors in detection (imprecise/wrong bounding box, etc.) with manipulation policy errors. Second, we gather human-provided assessments of perceived goal alignment via 2 Likert questions [26], rated from 1-7 (Strongly Disagree - Strongly Agree):

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

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

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

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

B. Experimental Results

In this section, we present our findings related to the hypotheses of Section IV by quantifying precision (Table I, Table II) and goal alignment (Fig. 3)) across policies.

H1: We evaluate all policies on each of the 6 skills on 15 different evaluation catalog scenarios per skill, varying objects (16 unique in total) and their placements. Overall, RT-Sketch performs comparably to RT-1 and RT-Goal-Image in both semantic (**Q1**) and spatial alignment (**Q2**), achieving average ratings from 'Agree' to 'Strongly Agree' for nearly all skills (Fig. 3 (top)). The exception is *upright*; both RT-Sketch and RT-Goal-Image tend to *position* cans or bottles appropriately, without realizing the need for *reorientation* (Appendix Fig. 10). This results in low semantic alignment

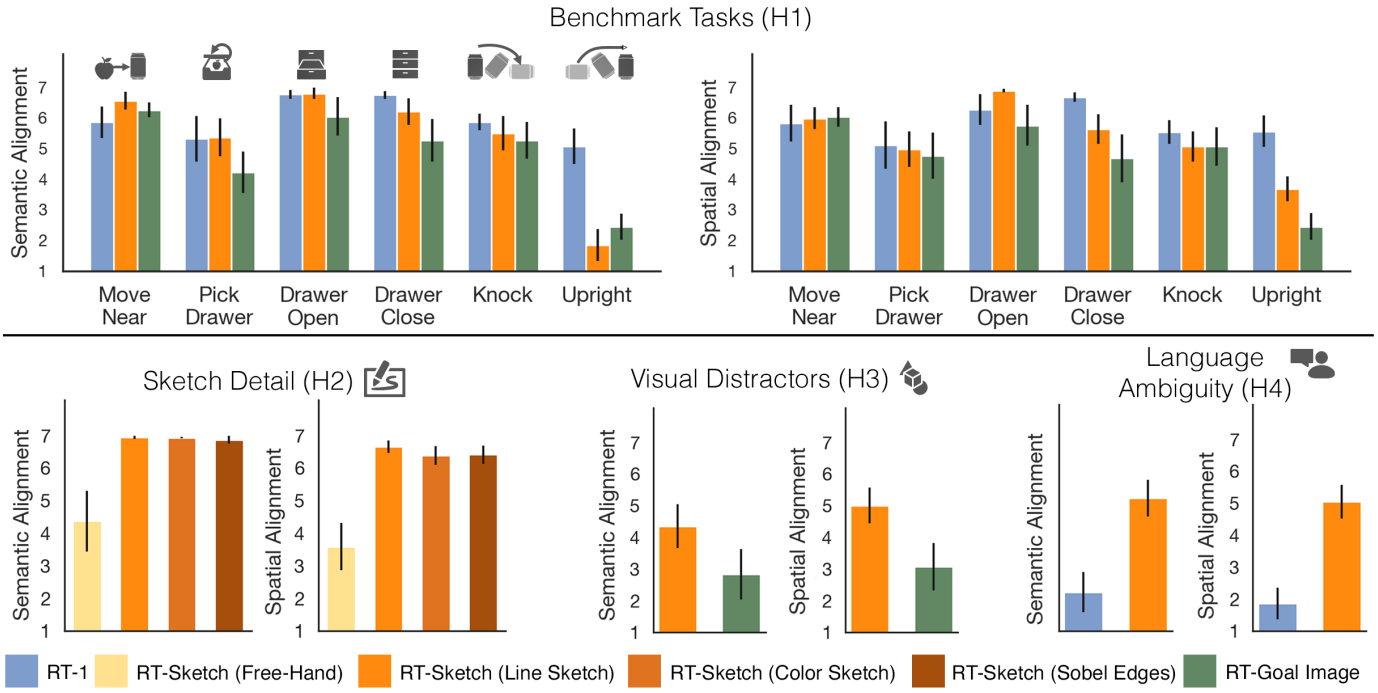

Fig. 3: **Goal Alignment Results:** Average Likert scores for different policies rating perceived semantic alignment (**Q1**) and spatial alignment (**Q2**) to a provided goal. Error bars indicate standard error. To back up the visual insights from these barplots, we report additional findings on statistically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

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

TABLE I: **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 across policies (right, bold=least failure-prone).

but somewhat higher spatial alignment (Fig. 3 (top), darker gray in Table I (left)). RT-1, on the other hand, reorients cans and bottles successfully, but at the expense of higher spatial error (Appendix Fig. 10, light color in Table I (left)). With RT-Goal-Image in particular, we also observe the occurrence of *excessive retrying behavior*, in which a policy attempts to align the current scene with a given goal with retrying actions that inadvertently disturb the scene, knocking objects off the table or undoing task progress. In Table I, we report the proportion of rollouts in which this occurs (via manual inspection) across all policies. RT-Goal-Image is most susceptible, as a result of over-attending to pixel-level details, while RT-Sketch and RT-1 are far less vulnerable, given the higher-level goal abstractions that sketches and language offer.

H2: Next, we assess RT-Sketch's ability to handle varying levels of sketch detail. Across 5 trials of the *move near* and *open drawer* skills, we see in Table II that many different

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

TABLE II: **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).

sketch types result in reasonable levels of spatial precision, particularly: free-hand sketches drawn completely free-form on a blank canvas, line sketches drawn by tracing an image, line sketches with color shading, and edge-detected images. Appendix Fig. 17 shows the interface used to sketch, and a detailed breakdown of the differences. As expected, Sobel edge-detected images incur the least error, but they are impractical and merely represent an upper-bound in terms of sketch detail. Even free-hand sketches, which do not necessarily preserve perspective projection, and line sketches, which are far sparser in detail, are not far behind in terms of precision or alignment

ratings. This is reflected in the Likert ratings (Fig. 3 (left, bottom)) of free-hand sketches (around 4 on average), and line sketches (nearly 7 – "Strongly Agree" on average). Adding color to line sketches does not further improve performance, but leads to interesting behavioral differences (see Appendix Fig. 11). In Appendix B, we also evaluate RT-Sketch on sketches drawn by 6 different individuals whose sketches were never seen during training and observe little-to-no policy performance drop-off compared to in-distribution sketches.

H3: Next, we compare the robustness of RT-Sketch and RT-Goal-Image to the presence of visual distractors. On 15 *move X near Y* trials from the evaluation catalog, we introduce $5-9$ distractor objects into the initial visual scene, replicating the setup of the RT-1 generalization experiments referred to as *medium-high* difficulty [8]. In Table I (left, bottom), we see that RT-Sketch exhibits far lower spatial errors on average, while producing higher semantic and spatial alignment scores over RT-Goal-Image (Fig. 3 (middle, bottom)). RT-Goal-Image is easily confused by the distribution shift introduced by distractor objects, and often cycles between picking up and putting down the wrong object. RT-Sketch, on the other hand, ignores task-irrelevant objects not captured in a sketch and completes the task in most cases (see Appendix Fig. 12).

H4: Finally, we evaluate whether sketches as a representation are favorable when language goals alone are ambiguous. On 15 evaluation catalog scenarios, we consider 3 types of language ambiguity: instance (**T1**) (e.g., *move apple near orange* when multiple orange instances are present), somewhat out-of-distribution (OOD) phrasing (**T2**) (e.g., *move left apple near orange*), and highly OOD phrasing (**T3**) (e.g., *complete the rainbow*) (see Appendix Fig. 13). Directional cues (i.e. 'left') should intuitively help resolve ambiguities, but were unseen during RT-1 training [8], and hence are out-of-distribution. In these scenarios, RT-Sketchachieves nearly half the error of RT-1 (Table I (left, bottom)), and a 2.33-fold and 2.71-fold score increase for semantic and spatial alignment, respectively (Fig. 3 (right, bottom)). For **T1** and **T2** scenarios, RT-1 often tries to pick up an instance of any object mentioned in the task string, but fails to make further progress (Appendix Fig. 14). This suggests the utility of sketches to express new, unseen goals with minimal overhead, when language can easily veer out of distribution (Appendix Fig. 15).

a) Towards Multimodal Goal Specification: For cases in which one modality alone is still ambiguous, we provide initial demonstrations showing that a multimodal (sketch-and-language conditioned) policy can be favorable to either alone, especially for tasks involving repositioning and reorientation (see Appendix C).

C. Limitations and Failure Modes

Firstly, the image-to-sketch generation network used in this work is fine-tuned on a dataset of sketches provided by a single human annotator. Although we empirically show that despite this, RT-Sketch can handle sketches drawn by other annotators (Appendix B), we have yet to investigate the effects of training RT-Sketch at scale with sketches produced by different people. Secondly, we note that RT-Sketch shows some inherent biases towards performing certain skills it was trained on (i.e. performing directional movements that are more represented in the demonstration trajectories). For a detailed breakdown of RT-Sketch's limitations and failure modes, please see Appendix H).

V. Conclusion

We propose RT-Sketch, a goal-conditioned policy for manipulation that takes a hand-drawn scene sketch as input, and outputs actions. We do so by developing a scalable way to generate paired sketch-trajectory training data via an image-to-sketch translation network, and modifying the existing RT-1 architecture to take visual information as an input. Empirically, RT-Sketch not only performs comparably to existing language or goal-image conditioning policies for a number of manipulation skills, but is amenable to different degrees of sketch fidelity, and more robust to visual distractors or ambiguities. Our rigorous evaluations comprise 400 cumulative robot rollouts, evaluated across 62 annotators (over 8 cumulative hours). Future work will focus on multimodal goal specification and moving towards even more abstract goal representations, detailed in Appendix G.

References

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

[2] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57 (5):469–483, 2009.

[3] Christine M Barber, Robin J Shucksmith, Bruce MacDonald, and Burkhard C Wünsche. Sketch-based robot programming. In *2010 25th International Conference of Image and Vision Computing New Zealand*, pages 1–8. IEEE, 2010.

[4] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Data quality in imitation learning. *arXiv preprint arXiv:2306.02437*, 2023.

[5] Ayan Kumar Bhunia, Viswanatha Reddy Gajjala, Subhadeep Koley, Rohit Kundu, Aneeshan Sain, Tao Xiang, and Yi-Zhe Song. Doodle it yourself: Class incremental learning by drawing a few sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2293–2302, 2022.

[6] Ayan Kumar Bhunia, Subhadeep Koley, Amandeep Kumar, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Sketch2saliency: Learning to detect salient objects from human drawings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2733–2743, 2023.

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding,

Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

[8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S Ryoo, Grecia Salazar, Pannag R Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan H Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics Transformer for Real-World Control at Scale. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.025.

[9] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[10] Kaylee Burns, Tianhe Yu, Chelsea Finn, and Karol Hausman. Robust manipulation with spatial features. In *CoRL 2022 Workshop on Pre-training Robot Learning*, 2022.

[11] Pinaki Nath Chowdhury, Ayan Kumar Bhunia, Aneeshan Sain, Subhadeep Koley, Tao Xiang, and Yi-Zhe Song. What can human sketches do for object detection? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15083–15094, 2023.

[12] Pinaki Nath Chowdhury, Ayan Kumar Bhunia, Aneeshan Sain, Subhadeep Koley, Tao Xiang, and Yi-Zhe Song. Scenetrilogy: On human scene-sketch and its complementarity with photo and text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10972–10983, 2023.

[13] Yuchen Cui, Scott Niekum, Abhinav Gupta, Vikash Kumar, and Aravind Rajeswaran. Can foundation models perform zero-shot task specification for robot manipulation? In *Learning for Dynamics and Control Conference*, pages 893–905. PMLR, 2022.

[14] Yuchen Cui, Siddharth Karamcheti, Raj Palleti, Nidhya Shivakumar, Percy Liang, and Dorsa Sadigh. No, to the right: Online language corrections for robotic manipulation via shared autonomy. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pages 93–101, 2023.

[15] Michael Danielczuk, Andrey Kurenkov, Ashwin Balakrishna, Matthew Matl, David Wang, Roberto Martín-Martín, Animesh Garg, Silvio Savarese, and Ken Goldberg. Mechanical search: Multi-step retrieval of a target object occluded by clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1614–1621. IEEE, 2019.

[16] Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. Goal-conditioned imitation learning. *Advances in neural information processing systems*, 32, 2019.

[17] Aditya Ganapathi, Priya Sundaresan, Brijen Thananjeyan, Ashwin Balakrishna, Daniel Seita, Jennifer Grannen, Minho Hwang, Ryan Hoque, Joseph E Gonzalez, Nawid Jamali, et al. Learning dense visual correspondences in simulation to smooth and fold real fabrics. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11515–11522. IEEE, 2021.

[18] Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *arXiv preprint arXiv:2311.01977*, 2023.

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[20] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[21] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2022.

[22] Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023.

[23] Subhadeep Koley, Ayan Kumar Bhunia, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Picture that sketch: Photorealistic image generation from

abstract sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6850–6861, 2023.

[24] Zixing Lei, Yiming Zhang, Yuxin Xiong, and Siheng Chen. Emergent communication in interactive sketch question answering. *arXiv preprint arXiv:2310.15597*, 2023.

[25] Mengtian Li, Zhe Lin, Radomir Mech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring contour drawings from images. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1403–1412. IEEE, 2019.

[26] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 1932.

[27] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020.

[28] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[29] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. In *The International Symposium of Robotics Research*, pages 132–157. Springer, 2019.

[30] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[31] David Porfirio, Laura Stegner, Maya Cakmak, Allison Sauppé, Aws Albarghouthi, and Bilge Mutlu. Sketching robot programs on the fly. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '23, page 584–593, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399647. doi: 10.1145/3568162.3576991. URL https://doi.org/10.1145/3568162.3576991.

[32] Shuwen Qiu, Sirui Xie, Lifeng Fan, Tao Gao, Jungseock Joo, Song-Chun Zhu, and Yixin Zhu. Emergent graphical conventions in a visual communication game. *Advances in Neural Information Processing Systems*, 35:13119–13131, 2022.

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[34] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *Robotics: Science and Systems (RSS)*, 2023.

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[36] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[37] Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. *Advances in Neural Information Processing Systems*, 34:12786–12797, 2021.

[38] Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[39] Irwin Sobel. An isotropic 3x3 image gradient operator. *Presentation at Stanford A.I. Project 1968*, 1968.

[40] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[41] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[43] Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. Clipascene: Scene sketching with different types and levels of abstraction. *arXiv preprint arXiv:2211.17256*, 2022.

[44] Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching. *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022.

[45] Kevin Zakka, Andy Zeng, Johnny Lee, and Shuran Song. Form2fit: Learning shape priors for generalizable assembly from disassembly. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9404–9410. IEEE, 2020.

[46] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.