A Datasets

The vision-language datasets are based on the dataset mixtures from Chen et al. [15] and Driess et al. [17]. The bulk of this data consists of the WebLI dataset, which is around 10B image-text pairs across 109 languages, filtered to the top 10% scoring cross-modal similarity examples to give 1B training examples. Many other captioning and vision question answering datasets are included as well, and more info on the dataset mixtures can be found in Chen et al. [15] for RT-2-PaLI-X, and Driess et al. [17] for RT-2-PaLM-E. When co-fine-tuning RT-2-PaLI-X, we do not use the Episodic WebLI dataset described by Chen et al. [16].

The robotics dataset is based on the dataset from Brohan et al. [1]. This consists of demonstration episodes collected with a mobile manipulation robot. Each demonstration is annotated with a natural language instruction from one of seven skills: "Pick `Object`", "Move `Object` Near `Object`", "Place `Object` Upright", "Knock `Object` Over", "Open `Drawer`", "Close `Drawer`", "Place `Object` into `Receptacle`", and "Pick `Object` from `Receptacle` and place on the counter". Further details can be found in Brohan et al. [1].

RT-2-PaLI-X weights the robotics dataset such that it makes up about 50% of the training mixture for co-fine-tuning. RT-2-PaLM-E weights the robotics dataset to be about 66% of the training mixture.

For the results on Language-Table in Table 2, our model is trained on the Language-Table datasets from Lynch et al. [78]. Our model is co-fine-tuned on several prediction tasks: (1) predict the action, given two consecutive image frames and a text instruction; (2) predict the instruction, given image frames; (3) predict the robot arm position, given image frames; (4) predict the number of timesteps between given image frames; and (5) predict whether the task was successful, given image frames and the instruction.

B Baselines

We compare our method to multiple state-of-the-art baselines that challenge different aspects of our method. All of the baselines use the exact same robotic data.

- **RT-1**: Robotics Transformer 1 [1] is a transformer-based model that achieved state-of-the-art performance on a similar suite of tasks when it was published. The model does not use VLM-based pre-training so it provides an important data point demonstrating whether VLM-based pre-training matters.
- **VC-1**: VC-1 [77] is a visual foundation model that uses pre-trained visual representations specifically designed for robotics tasks. We use pre-trained representations from the VC-1 ViT-L model. Since VC-1 does not include language conditioning, we add this by separately embedding the language command via Universal Sentence Encoder [81] to enable comparison to our method. In particular, we concatenate the resulting language embedding tokens to the image tokens produced by VC-1, and pass the concatenated token sequences through token learner [82]. The token sequences produced by token learner are then consumed by an RT-1 decoder-only transformer model to predict robot action tokens. We train the VC-1 baseline end-to-end and unfreeze the VC-1 weights during training, since this led to far better results than using frozen VC-1 weights.
- **R3M**: R3M [57] is a similar method to VC-1 in that R3M uses pre-trained visual-language representations to improve policy training. In this case the authors use Ego4D dataset [83] of human activities to learn the representation that is used by the policy. Both VC-1 and R3M test different state-of-the-art representation learning methods as an alternative to using a VLM. To obtain a language-conditioned policy from the R3M pretrained representation, we follow the same procedure as described above for VC-1, except we use the R3M ResNet50 model to obtain the image tokens, and unfreeze it during training.
- **MOO**: MOO [48] is an object-centric approach, where a VLM is first used to specify the object of interest in a form of a single, colored pixel in the original image. This pixel-modified image is then trained with an end-to-end policy to accomplish a set of manipulation tasks. This baseline corresponds to a situation where a VLM is used as a separate module that enhances perception but its representations are not used for policy learning.

C VLMs for RT-2

The PaLI-X model architecture consists of a ViT-22B [84] to process images, which can accept sequences of n images, leading to $n \times k$ tokens per image, where k is the number of patches per image. The image tokens passing over a projection layer is then consumed by an encoder-decoder backbone of 32B parameters and 50

layers, similar to UL2 [85], which jointly processes text and images as embeddings to generate output tokens in an auto-regressive manner. The text input usually consists of the type of task and any additional context (e.g., "Generate caption in ⟨lang⟩" for captioning tasks or "Answer in ⟨lang⟩: question" for VQA tasks).

The PaLI-3B model trained on Language-Table (Table 2) uses a smaller ViT-G/14 [86] (2B parameters) to process images, and UL2-3B [85] for the encoder-decoder network.

The PaLM-E model is based on a decoder-only LLM that projects robot data such as images and text into the language token space and outputs text such as high-level plans. In the case of the used PaLM-E-12B, the visual model used to project images to the language embedding space is a ViT-4B [15]. The concatenation of continuous variables to textual input allows PaLM-E to be fully multimodal, accepting a wide variety of inputs such as multiple sensor modalities, object-centric representations, scene representations and object entity referrals.

D Training Details

We perform co-fine-tuning on pre-trained models from the PaLI-X [16] 5B & 55B model, PaLI [15] 3B model and the PaLM-E [17] 12B model. For RT-2-PaLI-X-55B, we use learning rate 1e-3 and batch size 2048 and co-fine-tune the model for 80K gradient steps whereas for RT-2-PaLI-X-5B, we use the same learning rate and batch size and co-fine-tune the model for 270K gradient steps. For RT-2-PaLM-E-12B, we use learning rate 4e-4 and batch size 512 to co-fine-tune the model for 1M gradient steps. Both models are trained with the next token prediction objective, which corresponds to the behavior cloning loss in robot learning. For RT-2-PaLI-3B model used for Language-Table results in Table 2, we use learning rate 1e-3 and batch size 128 to co-fine-tune the model for 300K gradient steps.

E Evaluation Details

E.1 Evaluation Scenarios

For studying the emergent capabilities of RT-2 in a quantitative manner, we study various challenging semantic evaluation scenarios that aim to measure capabilities such as reasoning, symbol understanding, and human recognition. A visual overview of a subset of these scenes is provided in Figure 7, and the full list of instructions used for quantiative evalution is shown in Table 4.

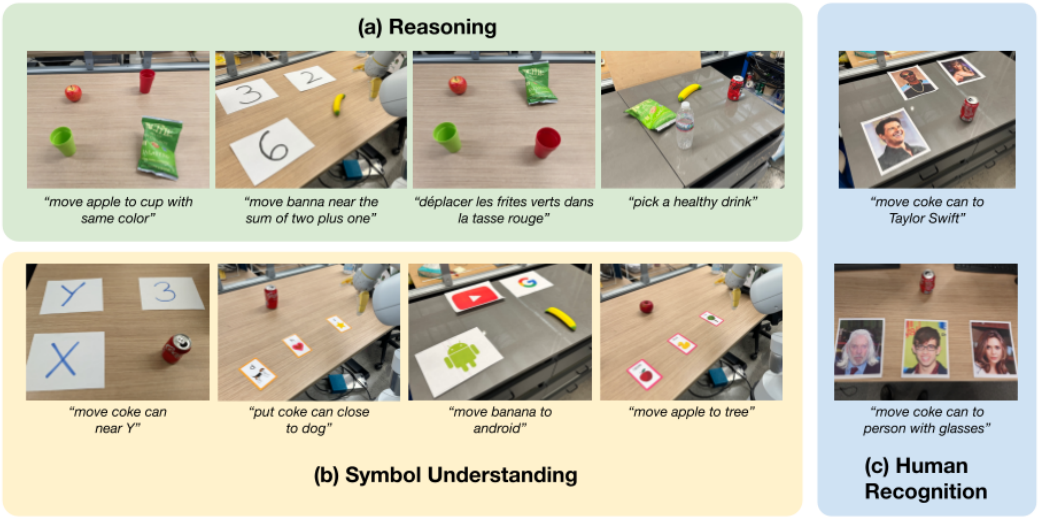

Figure 7: An overview of some of the evaluation scenarios used to study the emergent capabilities of RT-2. They focus on three broad categories, which are (a) reasoning, (b) symbol understanding, and (c) human recognition. The visualized instructions are a subset of the full instructions, which are listed in Appendix E.2.

E.2 Evaluation Instructions

Table 3 lists natural language instructions used in model evaluations for unseen objects, backgrounds, and environments. Each instruction was run between 1-5 times, depending on the number of total instructions in that evaluation set.

Table 4 lists natural language instructions used to evaluate quantitative emergent evals. Each instruction was run 5 times.

Task Group	Tasks
Unseen Objects (Easy)	pick banana, move banana near coke can, move orange can near banana, pick oreo, move oreo near apple, move redbull can near oreo, pick pear, pick coconut water, move pear near coconut water, move pepsi can near pear
Unseen Objects (Hard)	pick cold brew can, pick large orange plate, pick chew toy, pick large tennis ball, pick bird ornament, pick fish toy, pick ginger lemon kombucha, pick egg separator, pick wrist watch, pick green sprite can, pick blue microfiber cloth, pick yellow pear, pick pretzel chip bag, pick disinfectant wipes, pick pineapple hint water, pick green cup, pick pickle snack, pick small blue plate, pick small orange rolling pin, pick octopus toy, pick catnip toy
Unseen Backgrounds (Easy)	pick green jalapeno chip bag, pick orange can, pick pepsi can, pick 7up can, pick apple, pick blue chip bag, pick orange, pick 7up can, move orange near sink, pick coke can, pick sponge, pick rxbar blueberry
Unseen Backgrounds (Hard)	pick wrist watch, pick egg separator, pick green sprite can, pick blue microfiber cloth, pick yellow pear, pick pretzel chip bag, pick disinfectant wipes, pick pineapple hint water, pick green cup, pick pickle snack, pick small blue plate, pick small orange rolling pin, pick octopus toy, pick catnip toy, pick swedish fish bag, pick large green rolling pin, pick black sunglasses
Unseen Environments (Easy)	pick coke can, pick apple, pick rxbar blueberry, move apple near coke can, move rxbar blueberry near apple, move coke can near rxbar blueberry, pick blue plastic bottle, pick sponge, pick blue chip bag, move sponge near blue plastic bottle, move blue chip bag near sponge, move blue plastic bottle near blue chip bag, move coke can near white mug, move sponge near white mug, move coke can near yellow bowl, move sponge near yellow bowl, move coke can near green cloth, move sponge near green cloth, move coke can near plate, move sponge near plate, move coke can near spoon, move sponge near spoon, move coke can near orange cup, move sponge near orange cup, pick white mug, pick yellow bowl, pick green cloth, move white mug near sponge, move yellow bowl near sponge, move green cloth near sponge, pick plate, pick spoon, pick orange cup, move plate near sponge, move spoon near sponge, move orange cup near sponge, put coke can into sink, drop coke can into sink, push coke can into sink, put sponge into sink, drop sponge into sink, push sponge into sink, put green cloth into sink, drop green cloth into sink, push green cloth into sink
Unseen Environments (Hard)	pick coke can, pick apple, pick rxbar blueberry, move apple near coke can, move rxbar blueberry near apple, move coke can near rxbar blueberry, move coke can near stapler, move apple near stapler, move coke can near keyboard, move apple near keyboard, move coke can near tissue box, move apple near tissue box, move coke can near papers, move apple near papers, move coke can near mouse, move apple near mouse, move coke can near book, move apple near book, pick marker, pick stapler, pick mouse, move marker near apple, move stapler near apple, move mouse near apple, push coke can to the left, push coke can to the right, push sponge to the left, push sponge to the right, push tissue box to the left, push tissue box to the right, point at coke can, point at sponge, point at tissue box

Table 3: Natural language instructions used for evaluations testing controlled distribution shifts along the dimension of novel objects, novel environments, and novel backgrounds. For each category, we introduce evaluation settings with smaller distribution shifts as well as larger distribution shifts. A visualization of these scenarios if shown in Figure 3.

F Example Failure Cases

In Fig. 8 we provide examples of a notable type of failure case in the Language Table setting, with the RT-2 model not generalizing to *unseen object dynamics*. In these cases, although the model is able to correctly attend to the language instruction and move to the first correct object, it is not able to control the challenging dynamics of these objects, which are significantly different than the small set of block objects that have been seen in this environment [78]. Then pen simply rolls off the table (Fig. 8, left), while the banana's center-of-

Task Group	Tasks
Symbol Understanding: Symbol 1	move coke can near X, move coke can near 3, move coke can near Y
Symbol Understanding: Symbol 2	move apple to tree, move apple to duck, move apple to apple, move apple to matching card
Symbol Understanding: Symbol 3	put coke can close to dog, push coke can on top of heart, place coke can above star
Reasoning: Math	move banana to 2, move banna near the sum of two plus one, move banana near the answer of three times two, move banana near the smallest number
Reasoning: Logos	move cup to google, move cup to android, move cup to youtube, move cup to a search engine, move cup to a phone
Reasoning: Nutrition	get me a healthy snack, pick a healthy drink, pick up a sweet drink, move the healthy snack to the healthy drink, pick up a salty snack
Reasoning: Color and Multilingual	move apple to cup with same color, move apple to cup with different color, move green chips to matching color cup, move apple to vaso verde, Bewegen Sie den Apfel in die rote Tasse, move green chips to vaso rojo, mueve la manzana al vaso verde, déplacer les frites verts dans la tasse rouge
Person Recognition: Celebrities	move coke can to taylor swift, move coke can to tom cruise, move coke can to snoop dog
Person Recognition: CelebA	move coke can to person with glasses, move coke can to the man with white hair, move coke can to the brunette lady

Table 4: Natural language instructions used for quantitative emergent evalutions.

mass is far from where the robot makes contact (Fig. 8, right). We note that pushing dynamics are notoriously difficult to predict and control [87]. We hypothesize that greater generalization in robot-environment interaction dynamics may be possible by further scaling the datasets across diverse environments and objects – for example, in this case, datasets that include similar types of more diverse pushing dynamics [33].

In addition, despite RT-2's promising performance on real world manipulation tasks in qualitative and quantitative emergent evaluations, we still find numerous notable failure cases. For example, with the current training dataset composition and training method, RT-2 seemed to perform poorly at:

- Grasping objects by specific parts, such as the handle
- Novel motions beyond what was seen in the robot data, such as wiping with a towel or tool use
- Dexterous or precise motions, such as folding a towel
- Extended reasoning requiring multiple layers of indirection

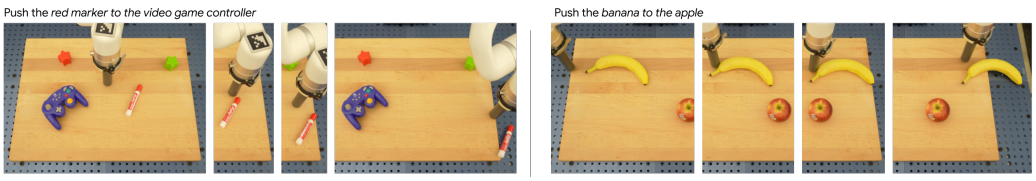

Figure 8: Qualitative example failure cases in the real-world failing to generalize to *unseen object dynamics*.

G Quantitative Emergent Evaluation

Table 5 lists all of our quantitative emergent evaluation results. We find that RT-2 performs 2x to 3x better than RT-1 on these new instructions, without any additional robotic demonstrations. This showcases how our method allows us to leverage capabilities from pretraining on web-scale vision-language datasets.

Model	Symbol Understanding				Reasoning					Person Recognition			Average
	Symbol 1	Symbol 2	Symbol 3	Average	Math	Logos	Nutrition	Color/Multilingual	Average	Celebrities	CelebA	Average	
VC-1 [77]	7	25	0	11	0	8	20	13	10	20	7	13	11
RT-1 [1]	27	20	0	16	5	0	32	28	16	20	20	20	17
RT-2-PaLI-X-55B (ours)	**93**	**60**	**93**	**82**	25	52	**48**	**58**	**46**	**53**	**53**	**53**	**60**
RT-2-PaLM-E-12B (ours)	67	20	20	36	**35**	**56**	44	35	43	33	**53**	43	40

Table 5: Performance of RT-2 and baselines on quantitative emergent evaluations.

H How does the generalization vary with parameter count and other design decisions?

For this comparison, we use RT-2-PaLI-X model because of its flexibility in terms of the model size (due to the nature of PaLM-E, RT-2-PaLM-E is restricted to only certain sizes of PaLM and ViT models). In particular, we compare two different model sizes, 5B and 55B, as well as three different training routines: training a model from scratch, without using any weights from the VLM pre-training; fine-tuning a pre-trained model using robot action data only; and co-fine-tuning (co-training with fine-tuning), the primary method used in this work where we use both the original VLM training data as well as robotic data for VLM fine-tuning. Since we are mostly interested in the generalization aspects of these models, we remove the *seen tasks* evaluation from this set of experiments.

The results of the ablations are presented in Table 6. First, we observe that training a very large model from scratch results in a very poor performance even for the 5B model. Given this result, we decide to skip the evaluation of an even bigger 55B PaLI-X model when trained from scratch. Second, we notice that co-fine-tuning a model (regardless of its size) results in a better generalization performance than simply fine-tuning it with robotic data. We attribute this to the fact that keeping the original data around the fine-tuning part of training, allows the model to not forget its previous concepts learned during the VLM training. Lastly, somewhat unsurprisingly, we notice that the increased size of the model results in a better generalization performance.

Model	Size	Training	Unseen Objects		Unseen Backgrounds		Unseen Environments		Average
			Easy	Hard	Easy	Hard	Easy	Hard	
RT-2-PaLI-X	5B	from scratch	0	10	46	0	0	0	9
RT-2-PaLI-X	5B	fine-tuning	24	38	79	50	36	23	42
RT-2-PaLI-X	5B	co-fine-tuning	60	38	67	29	44	24	44
RT-2-PaLI-X	55B	fine-tuning	60	62	75	38	57	19	52
RT-2-PaLI-X	55B	co-fine-tuning	70	62	96	48	63	35	63

Table 6: Ablations of RT-2 showcasing the impact of the parameter count and the training strategy on generalization.

I Additional Chain-Of-Thought Reasoning Results

We present additional examples of chain-of-thought reasoning rollouts accomplished with RT-2-PaLM-E described in Sec. 4.4.

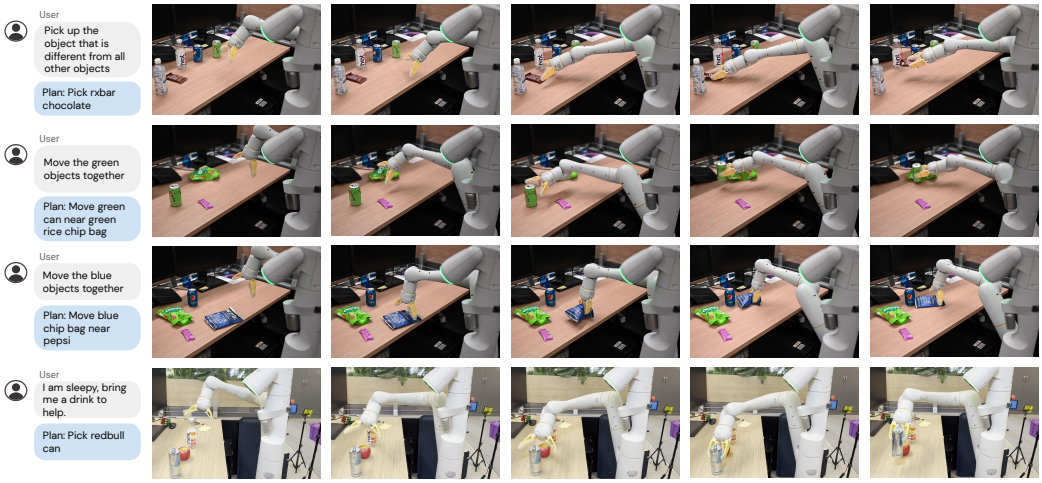

Figure 9: Additional examples of RT-2 with chain-of-thought reasoning