

$\pi_{0.7}$: a Steerable Generalist Robotic Foundation Model with Emergent Capabilities

Physical Intelligence

Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, Ury Zhilinsky

<https://pi.website/pi07>

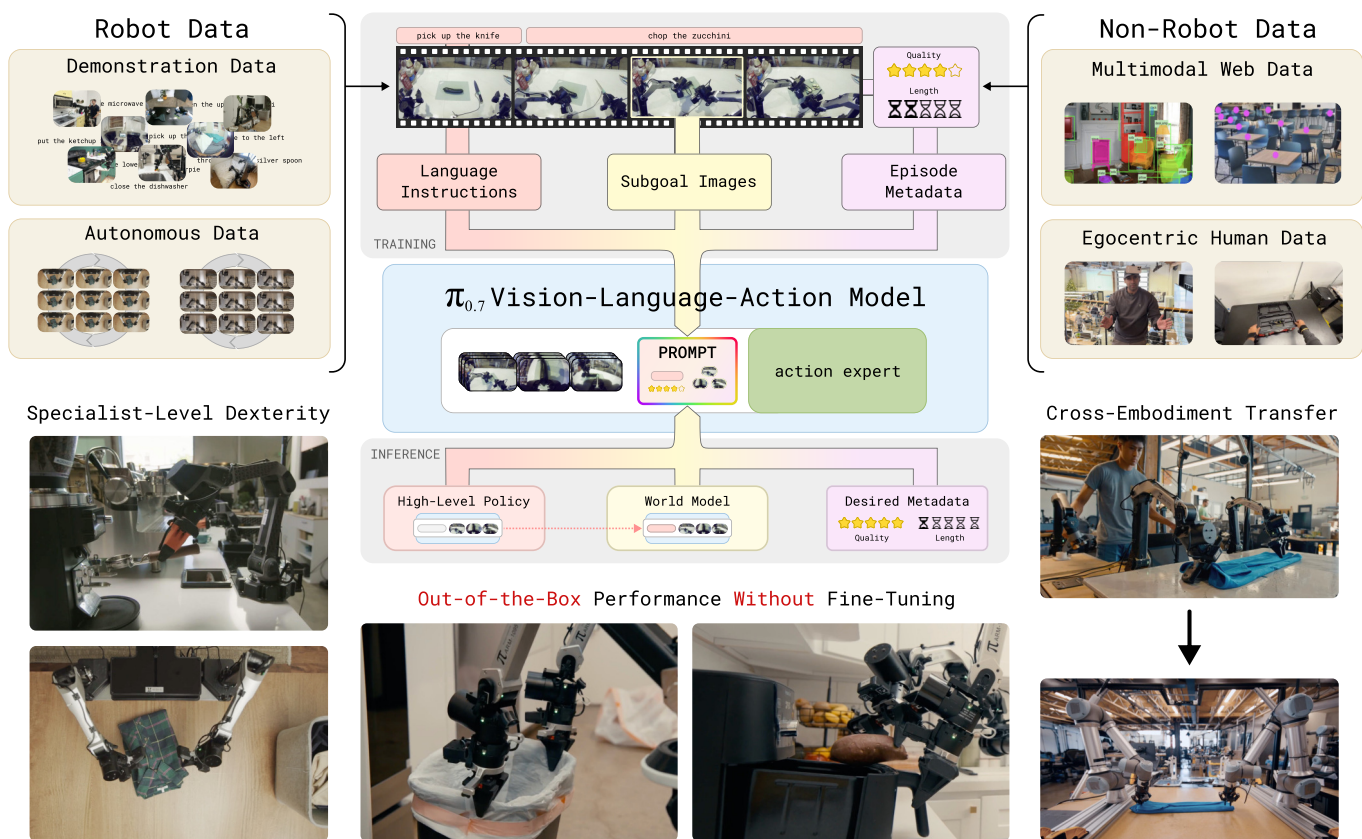


Fig. 1: We introduce $\pi_{0.7}$, a steerable generalist robot foundation model that can perform dexterous tasks across many tasks, environments, and robots. $\pi_{0.7}$ is trained with diverse prompts that contain not only the task description, but detailed language, generated subgoal images, and episode metadata. This provides richer context about not only *what* to do, but also *how* to do it, making it possible for $\pi_{0.7}$ to leverage broad set of both robot and non-robot data and compose the skills in this data in new ways to solve new tasks.

Abstract—We present a new robotic foundation model, called $\pi_{0.7}$, that can enable strong out-of-the-box performance in a wide range of scenarios. $\pi_{0.7}$ can follow diverse language instructions in unseen environments, including multi-stage tasks with various kitchen appliances, provide zero-shot cross-embodiment generalization, for example enabling a robot to fold laundry without seeing the task before, and perform challenging tasks such as operating an espresso machine out of the box at a level of performance that matches much more specialized

RL-finetuned models. The main idea behind $\pi_{0.7}$ is to use diverse context conditioning during training. This conditioning information, contained in the prompt, makes it possible to steer the model precisely to perform many tasks with different strategies. It is conditioned not just on a language command that describes what it should do, but on additional multimodal information that also describes the manner or strategy in which it should do it, including metadata about task performance and subgoal images. This enables $\pi_{0.7}$ to use very diverse data,

including demonstrations, potentially suboptimal (autonomous) data including failures, and data from non-robot sources. Our experiments evaluate $\pi_{0.7}$ across numerous tasks with multiple robot platforms, on tasks that require speed and dexterity, language following, and compositional task generalization.

I. INTRODUCTION

I am a part of all that I have met.

Alfred, Lord Tennyson, *Ulysses*

Foundation models work on the principle that generalist capabilities emerge from training on large and diverse datasets. For example, large language models can not only recall facts and semantic knowledge, but they can also compose that knowledge in new ways, solving problems that require unlikely connections, applying user-defined formats (e.g., JSON), and performing chain-of-thought reasoning. This kind of *compositional* generalization is arguably the cornerstone of generalist capabilities, but it has proven elusive in the domain of physical intelligence. While robotic foundation models such as vision-language-action models (VLAs) have advanced significantly in size and capability, their ability to generalize to new tasks or recombine skills in new ways has so far been limited. Unlike language models, which can compose different capabilities from their training data to solve new problems, prior VLAs not only lack the ability to solve new tasks, but often struggle to fluently perform all of the instructions they were trained on without task-specific fine-tuning.

In this paper, we present a new model, $\pi_{0.7}$, that exhibits strong signs of compositional generalization — enabling it to follow diverse language instructions, attain performance comparable to more specialized fine-tuned models on dexterous tasks, and even compose these behaviors in new ways. This is enabled by leveraging large and diverse datasets, including data from many robots with diverse strategies, suboptimal data from autonomous execution (including both data from RL post-trained agents as well as failures), and non-robot data from videos of humans performing tasks and general multimodal data from the internet. However, using such data naïvely does not lead to success: with a diversity of examples that differ in terms of both strategy and task performance, a naïve training process would lead to a model that averages together different modes in the dataset and produces sub-optimal results. In training $\pi_{0.7}$, we address this challenge by annotating the data with detailed context annotations that contain not only information about *what* to do but also *how* to do it and provide this knowledge to the model using a variety of multimodal conditioning signals. In this way, each episode teaches the robot about nuanced concepts and skills that it could use not only to perform the training tasks effectively, but also to compose in new ways to solve new tasks. Our proposed prompt structure includes detailed language labels, strategy metadata, and multimodal information such as subgoal images. This allows us to resolve the ambiguity in large and diverse datasets, learn from suboptimal behaviors without hurting performance, and obtain broad generalization across instructions, embodiments, and environments.

The idea that detailed prompts or context can improve the performance of foundation models has been explored in other fields. For example, models for image and video generation utilize *prompt expansion* to produce high-quality generations. Our approach has many parallels to such methods. However, in robotics, simply *captioning* the data with more detailed text is not enough — the details that determine task success and proficiency might be more subtle (e.g., information about the overall quality of the episode), or might simply be hard to express with language alone (e.g., the particular appearance of a cleanly folded t-shirt). Therefore, in addition to using more detailed text, our model adds a range of additional metadata to the prompt, as shown in Fig. 1, including information about episode quality (strategy metadata), the control modality used by the robot, and subgoal images. Some of this information can be provided or omitted at test time, but including it in training results in a model that can more effectively compose the concepts that it was trained on and exhibit a variety of emergent capabilities.

In our evaluation, we show that $\pi_{0.7}$ exhibits a number of capabilities that go beyond prior robotic foundation models:

- **Out-of-the-box performance:** $\pi_{0.7}$ can reliably perform highly dexterous, long-horizon tasks such as using an espresso machine, folding laundry, taking out a trash bag, folding a box, and peeling vegetables, without any task-specific post-training and in a variety of environments.
- **Instruction generalization:** $\pi_{0.7}$ can follow a diverse set of language instructions in unseen environments and demonstrates robust generalization to complex, unseen language references. For example, $\pi_{0.7}$ can follow a diverse set of open-ended instructions in entirely unseen kitchen and bedroom environments.
- **Cross-embodiment generalization:** $\pi_{0.7}$ can enable zero-shot cross-embodiment transfer, making it possible to transfer dexterous tasks such as folding a t-shirt to a robot that was never trained to perform any laundry folding task, matching the performance of expert operators teleoperating the robot on their initial attempts.
- **Compositional task generalization:** $\pi_{0.7}$ can be instructed to perform new tasks by composing skills in previously unseen ways. For example, we can prompt $\pi_{0.7}$ to use new kitchen appliances, such as loading a sweet potato into an air fryer, or prompt it to perform tasks in new ways.

Through ablation and scaling studies, we also empirically demonstrate that there is a strong synergy between diverse datasets and detailed contexts: our approach enables learning from mixed-quality data and non-standard data sources without hurting the model performance, and diverse data boosts the model performance when detailed context information is provided during training.

II. RELATED WORK

Generalist robot manipulation policies. There is a large body of work studying the development of generalist robot policies. These generalist policies are sometimes trained from

scratch [1–6], but are more commonly initialized using pre-trained vision-language models [7–23] or pre-trained video generation models [24–28]. Various works have developed architectural components of VLAs such as memory [29–37], hierarchy for long-horizon planning [13, 14, 38–40], and goal image conditioning [41]. We develop a VLA model that incorporates all three of these components in a single model, building on top of the $\pi_{0.6}$ -MEM architecture [37, 42]. While generalist policies are most often trained on robot demonstration data, prior works have shown how to derive benefits from incorporating web data [7], egocentric videos of humans [25, 43–49], and autonomous robot experience [50–52] into pretraining. We incorporate all of these data sources and find that the combination of diverse data with detailed prompting yields a model with strong signs of compositional generalization and performant out-of-the-box behavior.

Generalization across tasks and embodiments. Much prior work has aimed to learn robot policies that generalize not only to different environments, objects, and backgrounds, but also to entirely new tasks and embodiments. Often, this is done by leveraging human video data, either for general representation learning [53–58], by directly supervising with human motions [45, 59–65], or by extracting 2D point tracks [66–69]. Other work has aimed to improve generalization by directly leveraging Internet pre-trained foundation models during training or inference [70–76]. With the increasing availability of large, cross-embodiment robot datasets [77], there has also been work on explicitly improving cross-embodiment transfer between robots [78–84]. Rather than leveraging existing datasets, some works have proposed specialized hand-held devices that can be used to collect data that can then generalize to various robot embodiments [85, 86]. In this work, we find that the right prompting allows our model to leverage diverse robot, human, and Internet data to achieve strong generalization across tasks and embodiments.

Prompting robots with subgoal images. A core architectural component of our model relative to $\pi_{0.6}$ -MEM is to allow the model to be prompted using goal images, including generated subgoal images. Conditioning robot manipulation policies on goal images and videos has been explored in a large body of work. Some of these works utilize user-provided images [87–90], while others condition the policy on generated goal images from a separate model [91–98] or in a chain-of-thought fashion [41]. Alternatively, image and video generation can be integrated into policy training objectives [24–26, 99–101] to improve policy representations and produce more generalizable actions. We view our contribution as complementary to these works: we do not aim to propose a new architecture or model design, so much as a methodology for enabling VLAs to utilize more diverse data sources, together with an empirical analysis showing that it leads to strong indications of compositional generalization. To our knowledge, our empirical results go significantly beyond the quantitative improvements reported in prior works, showing zero-shot transfer of dexterous skills like laundry folding to a different robot and generalization to novel object interactions such as operating an air fryer.

III. FLOW-BASED VISION-LANGUAGE-ACTION MODELS

VLAs are trained by starting from a pre-trained vision-language model (VLM) backbone, and adapting it for robotic control. The training dataset \mathcal{D} contains robot trajectories, which are sequences of observations \mathbf{o}_t and actions \mathbf{a}_t . The observations $\mathbf{o}_t = [\mathbf{I}_t^1, \dots, \mathbf{I}_t^n, \mathbf{q}_t]$ consist of n camera images \mathbf{I}_t^i and the joint configuration of the robot \mathbf{q}_t , while the actions \mathbf{a}_t consist of joint or end-effector commands.

VLAs are typically trained to predict an *action chunk*, corresponding to a short trajectory of future actions $\mathbf{a}_{t:t+H}$, based on a recent history of observations $\mathbf{o}_{t-T:t}$ (often a shorter horizon of actions, $\hat{H} < H$, is executed). The action chunk can be generated by an “action expert”, a smaller transformer that attends to the VLM backbone and enables fast inference at runtime. The action expert typically uses a flow matching [102] (or diffusion) objective that captures the multi-modality of the robot actions. To learn effective representations, our model also uses the knowledge insulation (KI) training recipe [103]: the VLM backbone is supervised with FAST tokens [104], and while the action expert attends to all of the activations in the VLM backbone, gradients from the action expert do not flow into the VLM backbone, such that the VLM is trained via the comparatively stable discrete cross-entropy loss.

In addition to the observation and action, each training example for the VLA is accompanied by a *prompt* or *context*, which we denote with \mathcal{C}_t . Conventionally this corresponds to a language instruction ℓ_t , such that $\mathcal{C}_t = (\ell_t)$, provided by a human annotator (e.g., “clean up the kitchen”).

In designing $\pi_{0.7}$, we explore how additional information added to the context for each training example can enable learning from diverse and heterogeneous datasets (including suboptimal behaviors and failures). As we will show, training with this data leads to a model with greater robustness and dexterity, and makes it possible for the model to generalize more broadly. The training objective for the VLA π_θ corresponds to an approximate log-likelihood given by

$$\max_{\theta} \mathbb{E}_{\mathcal{D}} [\log \pi_{\theta}(\mathbf{a}_{t:t+H} \mid \mathbf{o}_{t-T:t}, \mathcal{C}_t)]. \quad (1)$$

Note that a flow matching action expert optimizes an approximate lower bound rather than a closed form log-likelihood [10]. The dataset \mathcal{D} typically consists of high-quality human demonstration trajectories; however, as mentioned, we use a broader dataset, which includes failed episodes and suboptimal autonomous rollouts, as well as other data sources such as egocentric human video data. We will show how using a sufficiently detailed and informative context \mathcal{C}_t makes it possible to incorporate such diverse data and, perhaps surprisingly, even results in better policy performance and generalization.

IV. $\pi_{0.7}$ OVERVIEW

$\pi_{0.7}$ is our newest robotic foundation model that builds on the existing VLA architecture from $\pi_{0.6}$ [42] and the MEM memory system [37] and extends it with multi-modal context conditioning. The model consists of a VLM backbone initialized from the Gemma3 4B-parameter VLM [106]

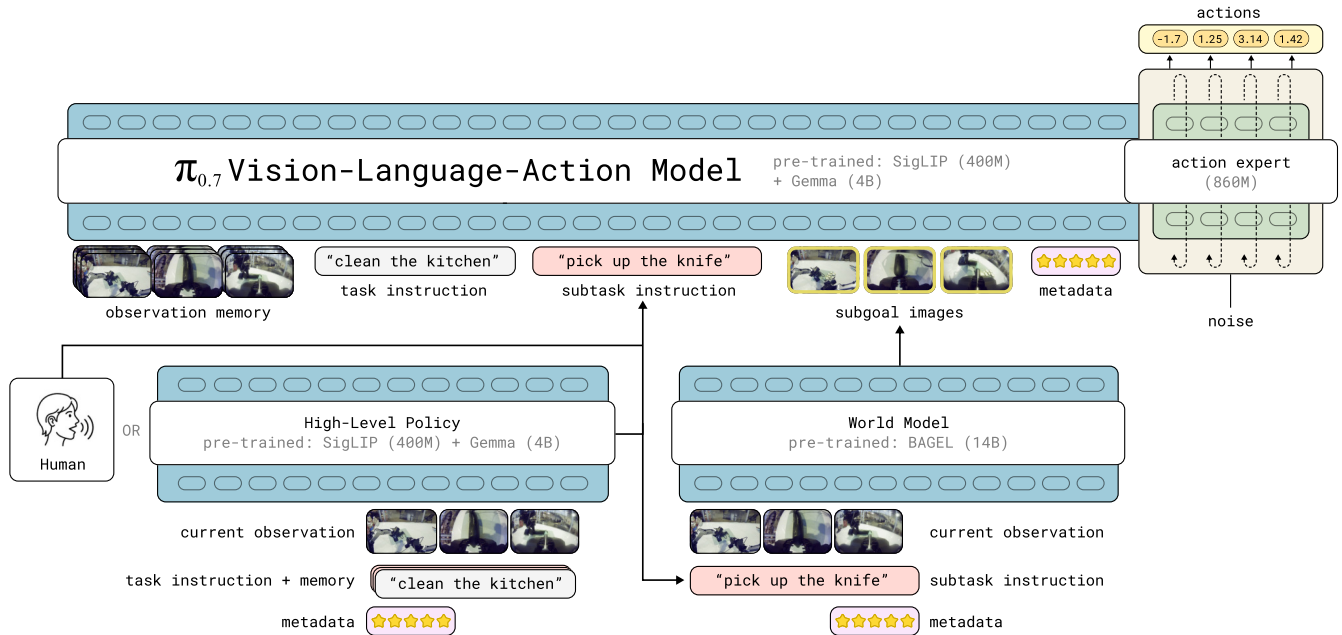


Fig. 2: **Architecture overview.** The $\pi_{0.7}$ model is a 5B-parameter VLA consisting of a 4B VLM backbone, a MEM-style video history encoder and a 860M parameter action expert. The model’s context includes multiple distinct modalities, including language commands, episode metadata that describes the data quality and strategy, and multimodal inputs such as subgoal images. At runtime, the language commands are produced by a high-level semantic policy based on the same architecture, and the subgoal images are produced by a lightweight world model based on the BAGEL image generation model [105].

(including a 400M-parameter vision encoder), and a flow matching action expert with 860M parameters. The model has about 5B total parameters. The vision encoder is also initialized from Gemma3 and follows the design of the MEM video history encoder [37], applying both temporal and spatial compression over history observations and outputting a fixed number of tokens for any number of history frames. An overview of the model architecture is provided in Fig. 2, and Sec. VI-B describes the architecture in more detail.

Our previous models, π_0 , $\pi_{0.5}$, and $\pi_{0.6}$, use a short textual description of the task as the context. In training $\pi_{0.7}$, we expand the context to include additional information and modalities: more expressive language commands, episode metadata, and subgoal images, making it possible to train on diverse and potentially suboptimal data.

V. DIVERSIFYING THE PROMPT

In this section, we describe each part of the prompt contained in the context \mathcal{C}_t used by $\pi_{0.7}$. The model is trained to handle prompts that contain each of these components, though it is trained with each component randomly dropped out so that it can also handle any subset, providing flexibility at test time.

A. Subtask instructions

Following $\pi_{0.5}$ [14], we include intermediate, higher-level text that captures the **next semantic subtask** as part of the prompt in addition to the overall textual task description ℓ_t (e.g., “clean the kitchen”). We denote this intermediate text by $\hat{\ell}_t$ (e.g., “open the fridge door”). During inference, $\hat{\ell}_t$ may be produced by a learned high-level policy or a human (or

be omitted) and may change over time. We collect data from a diverse set of tasks and scenarios, and then annotate the segments with detailed textual descriptions.

Conditioning the model on the semantic subtask also enables us to **verbally coach** the model step-by-step. Since the model is trained to follow diverse language instructions, it can follow the live instructions from the human in a new task, e.g., loading a sweet potato into an air fryer (Fig. 14). After coaching we can take the verbal coaching data to finetune $\pi_{0.7}$ as a high-level policy that maps the robot observations, task specification, and history of past subtask instructions to the new subtask instruction (Fig. 2 bottom left). This high-level policy then guides the robot to perform the task fully autonomously.

B. Subgoal images

While subtask instructions are effective at conveying the high-level intent of the task, they may lack details that matter for execution — e.g., “open the fridge door” does not specify how the robot arm should grasp the handle. Subgoal images address this by depicting the desired near-future state of the scene in images, providing a richer specification of *what the world should look like* after successful progress of the task.

We consider **multi-view subgoals** $\mathbf{g}_t = [G_t^1, \dots, G_t^n]$, where G_t^i is the desired near-future image for camera i . Multi-view subgoals simultaneously specify environment- and object-centric outcomes (often easiest in the base view) and arm/gripper outcomes (often easiest in wrist views), improving spatial grounding for control.

At runtime, the subgoal images are produced by a

lightweight world model, which takes in the same subtask instruction $\hat{\ell}_t$ as the main model, but benefits from web-scale pre-training on videos and image editing tasks and is thus capable of generalizing to diverse tasks and scenarios. Generated subgoal images that are grounded in the robot’s current observation can often more clearly disambiguate the objective for the policy than a language instruction, resulting in improvements in language following and generalization. We denote this model as g_ψ and it is trained with the objective

$$\max_{\psi} \mathbb{E}_{\mathcal{D}_g} \left[\mathcal{L}_{\text{CFM}} \left(\mathbf{g}_t^*, g_\psi(\mathbf{o}_t, \hat{\ell}_t, m) \right) \right],$$

where \mathcal{L}_{CFM} is the standard flow matching loss [102], \mathbf{g}_t^* is the future subgoal image, m is the episode metadata from Sec. V-C, and the dataset \mathcal{D}_g is a subset of segments from Sec. V-A that are annotated with especially high-quality subtask labels $\hat{\ell}_t$. The image frames at the end of the segments serve as the ground-truth subgoal, i.e., $\mathbf{g}_t^* = \mathbf{o}_{t_{\text{end}}}$.

Following SuSIE [93], our world model is initialized using an off-the-shelf image generation and editing model with web-scale pre-training. We initialize from BAGEL [105], a 14B mixture-of-transformers model capable of image understanding, editing, and generation. By augmenting our world model training with web data, non-robot data sources such as egocentric human videos, and other video data, we can acquire semantic and physical concepts from these other data sources and then transfer them into $\pi_{0.7}$ via subgoal images. Implementation details are in Appendix C.

C. Episode metadata

A key goal in expanding the context provided to the model is to train on a broader, more diverse dataset of trajectories. Instead of just using high-quality demonstration data, $\pi_{0.7}$ leverages lower quality demonstrations (including failures) and even autonomous data from prior models. Since we still want $\pi_{0.7}$ to perform the task as well as possible at test time, we need to appropriately label these diverse trajectories with information about *how* the task was performed so that the model can correctly contextualize them. To this end, we add a variety of “episode metadata” information to the context with attributes of the given training episode. We denote the set of metadata m , which may contain various labels including

- **Overall speed:** the length of the episode in timesteps. We discretize the values in an interval of 500 steps, i.e., values between 1750 and 2250 are binned to “2000 steps”. Often faster speed also corresponds to higher quality, e.g., the episode has fewer mistakes.
- **Overall quality:** task execution quality expressed as a score between 1 and 5, with 5 being the highest quality.
- **Mistake:** label indicating whether the robot made a mistake within a given action segment (e.g., failing to grasp an object or performing the wrong subtask). These labels are provided by humans coarsely annotating our data.

The $\pi_{0.7}$ model is thus trained with ground-truth episode speed and manual annotations of the episode quality and mistake segments from a diverse data mixture. The data diversity

(e.g., episodes of varying speed) provides the necessary signals for the model to learn to correlate such metadata with the target action. At runtime, the model can then be instructed to perform the task at high speed, with high quality, and without mistakes, through metadata prompting.

D. Control mode

We also consider using different control modes for the low-level action execution. Specifically, we include both **joint-level and end-effector** actions during training and use a text identifier $c \in \{\text{joint}, \text{ee}\}$ to designate the control mode in the prompt. Then at runtime, we can pick the control mode depending on the task.

E. Full prompt and training details

```

<Multi-view observation><Multi-view subgoals>
Task: peel vegetables. Subtask: pick up the
peeler. Speed: 8000. Quality: 5. Mistake:
false. Control Mode: joint.<Proprioception>
```

Combining all of the context information together, the example above illustrates a potential prompt that may be provided to the model.

During training, we randomly drop out each part of the prompt, which provides $\pi_{0.7}$ with the flexibility to use any subset of the prompt components at test time (e.g., running with or without subgoal images). First, we find that the model trains significantly faster when given the subgoal images — the action prediction task essentially becomes an “inverse dynamics” problem inferring the robot action between the current and future frames. Thus we only add the visual subgoal images to 25% of the examples in each batch in training. Among the examples with subgoal images we also drop out the subtask instruction $\hat{\ell}_t$ 30% of the time as often visual subgoal can substitute the equivalent textual subtask description in richer details. For episode metadata, we drop it entirely 15% of the time, and additionally each component (overall speed, overall quality, and mistake label) is dropped with 5% probability individually. We do not apply dropout for the control mode.

VI. THE $\pi_{0.7}$ MODEL AND TRAINING RECIPE

We now discuss how we incorporate the different context in $\pi_{0.7}$ model by training on diverse data, as well as the details about the model architecture, training, and inference.

A. Training datasets

The training dataset for $\pi_{0.7}$ consists of demonstration data for a wide range of tasks with many different robot platforms (both static and mobile, with single arm or bimanual) in diverse environments (in-house lab-like and home-like environments, and in-the-wild home environments), autonomous data from a large amount of policy evaluations, human interventions within policy rollouts, open-source robot datasets, egocentric human video data, and auxiliary non-robot data sources from the web, including object localization and attribute prediction, visual question answering, and text-only

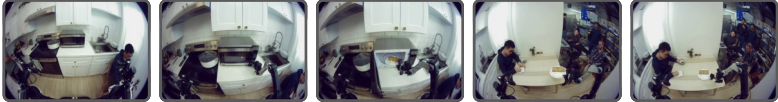


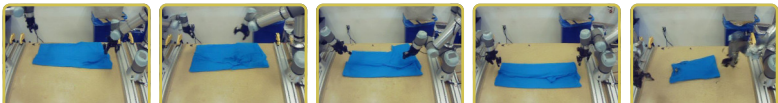
Task: "put food on table"	Observation	
	Subgoal Image	
	Subtask Instruction	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid #ccc; padding: 5px; width: 15%;">"push the open button on the microwave"</div> <div style="border: 1px solid #ccc; padding: 5px; width: 15%;">"pick up the plate of food in the microwave"</div> <div style="border: 1px solid #ccc; padding: 5px; width: 15%;">"move to the dining table"</div> <div style="border: 1px solid #ccc; padding: 5px; width: 15%;">"put the plate with food on the dining table"</div> <div style="border: 1px solid #ccc; padding: 5px; width: 15%;">"close the microwave"</div> </div>
	Metadata	<div style="border: 1px solid #ccc; padding: 5px; display: flex; justify-content: space-between;"> Quality: 5/5 Mistake: false </div>
Task: "fold the shirt"	Observation	
	Subgoal Image	
	Metadata	<div style="border: 1px solid #ccc; padding: 5px; display: flex; justify-content: space-between;"> Quality: 5/5 Speed: 2000 Mistake: false </div>

Fig. 3: **Prompt overview.** $\pi_{0.7}$ uses diverse modalities of context in the prompt, including: subtask instructions, subgoal images, and episode metadata. We train the model with dropout for each component, and then prompt the model flexibly combining modalities. For example, when using the UR5e bimanual manipulator to fold a shirt, we use subgoal image and metadata prompting.

prediction. We also include video-language tasks including video captioning of in-house robot data and from the web.

In a significant departure from classic VLA training pipelines, we make heavy use of suboptimal robot data in training. This includes both lower quality demonstrations (failure episodes or success episodes with a substantial amount of mistakes) and data collected by prior versions of our models during model evaluation experiments¹. For example, we use data collected by the $\pi_{0.6}^*$ model during RL training as additional examples, effectively allowing $\pi_{0.7}$ to distill their behavior. Incorporating the episode metadata into the context allows our model to effectively use all of this evaluation data and, as we will see in Sec. IX-A, enables it to attain similar performance as models that are specialized for high performance on individual tasks with RL. This corresponds to a kind of “distillation” process, where the generalist $\pi_{0.7}$ model can inherit the capabilities of RL-trained specialists. Suboptimal data also diversifies the possible states and scenarios in a given task and leads to stronger robustness, enabling the model to even sometimes outperform RL-trained or generally, single-task post-trained policies, in highly dexterous tasks.

B. Model architecture

The major architectural modifications of $\pi_{0.7}$ model compared to previous $\pi_{0.5}$ and $\pi_{0.6}$ models include the use of

¹We exclude autonomous data collected in any generalization-focused evaluation task (including ones in Sec. IX) from training.

the history vision encoder from MEM [37] and visual subgoal images in the context. The model takes as input up to four camera images (front view, two wrist views, and optionally rear view), each with up to six history frames, and up to three subgoal images (omitting the rear view). The history frames are processed through the vision encoder and compressed to the same number of tokens as a single frame; subgoal images are processed through the same encoder. Both the camera observations and subgoal images are first resized to 448x448 pixels. For sampling history frames we use a stride of 1 second, and the entire history frames are dropped out entirely with probability 0.3. The rear view image (when available) is dropped out with probability 0.3 as well.

We employ a block-causal masking scheme, such that the observation tokens and the subgoal image tokens use bidirectional attention within themselves, and goal-image tokens can additionally attend the observations. The following text tokens use causal attention (see attention mask visualization in the appendix). We also feed the proprioceptive state \mathbf{q}_t (including the history states) of the robot into the model backbone. Unlike $\pi_{0.6}$ that uses discretized text tokens to represent \mathbf{q}_t , $\pi_{0.7}$ follows MEM and embeds the state using a linear projection that maps the state dimension to the backbone dimension. Each history state is treated as an individual token; if the history frame is dropped out, the corresponding state token is masked out as well.

The more lightweight “action expert” is a 860M-parameter

transformer that is trained to predict continuous actions using flow matching objective. We use adaptive RMSNorm to inject timestep information for flow matching. The number of action tokens processed by the action expert is fixed at 50, representing an action chunk of 50 steps. The 50 tokens attend bidirectionally to each other and can also attend to the VLM backbone activations.

$\pi_{0.7}$ also employs the training-time version of real-time action chunking (RTC) [107, 108] for generating smooth action trajectories in the presence of inference delay. During training, we simulate delays of 0 to 12 timesteps, corresponding to a maximum inference latency of 240ms on a 50Hz robot.

C. Training with subgoal images

When training $\pi_{0.7}$ to handle subgoal images, we need the model to accommodate goals with different delays and different levels of image quality, including images generated by our world model. This requires carefully selecting which subgoals are provided as context to the model when training. We train on a combination of real images from future timesteps of the training trajectory and generated images. We found the following sampling scheme to be effective for selecting the timesteps for the real images: with probability 0.25, we sample the end-of-segment images (consistent with the prediction target for the world model), and with probability 0.75 we sample future images uniformly from 0–4 seconds ahead of the current timestep. In addition to these real images, we mitigate the train-test mismatch between real and generated images by also sampling a large number of subgoal images from the world model, and constructing additional training examples with these *generated* images added into the context of $\pi_{0.7}$ instead of the real future images.

VII. PROMPTING $\pi_{0.7}$ AT RUNTIME

At runtime, we configure $\pi_{0.7}$ to run with different forms of context depending on the desired behavior **without any task-specific post-training**. For any task we always prompt the model with the control mode and episode metadata. For choosing the episode metadata, we follow

- Overall speed: set per-task to the 15th percentile of the episode length from the task.
- Overall quality: always set to 5, which is the highest score.
- Mistake: always set to false, meaning no mistake.

The subtask instruction $\hat{\ell}_t$ is provided either by a learned high-level language policy or by a human supervisor for coaching (see Sec. V-A). When the subgoal images are used, we refresh the subgoal images whenever the semantic intent changes (i.e., new $\hat{\ell}_t$), or after $\Delta = 4$ seconds have elapsed since the last subgoal image was produced, whichever happens first. See Algorithm 1 for the full workflow. We apply asynchronous inference: the visual subgoal and subtask instruction generation happens in separate threads and the VLA inference always uses the latest ones available.

For all experiments we use 5 denoising steps to generate the 50-step action chunks and execute $\hat{H} \in \{15, 25\}$ steps out of the chunk. Since each prompt component is trained with

Algorithm 1 Prompting $\pi_{0.7}$ at test time

- 1: **Input:** initial observation \mathbf{o}_0 , task instruction ℓ , episode metadata m , control mode c
 - 2: Initialize subtask $\hat{\ell}$ (from high-level policy or coaching)
 - 3: $\mathbf{g}^* \sim p_\psi(\mathbf{g}^* | \mathbf{o}_0, \hat{\ell}, m)$
 - 4: $\mathcal{C} = \{\ell, \hat{\ell}, \mathbf{g}^*, m, c\}$
 - 5: $\mathbf{a}_{t:t+H} \sim \pi_\theta(\mathbf{a} | \mathbf{o}_{t-T:t}, \mathcal{C})$ ▷ Optional: CFG
 - 6: **for** $t = 0, 1, 2, \dots$ **do**
 - 7: **if** $\hat{\ell}$ changed **or** Δ -second timer elapsed **then**
 - 8: $\mathbf{g}^* \sim p_\psi(\mathbf{g}^* | \mathbf{o}_t, \hat{\ell}, m)$ ▷ Non-blocking (async)
 - 9: $\mathcal{C} = \{\ell, \hat{\ell}, \mathbf{g}^*, m, c\}$
 - 10: **end if**
 - 11: **if** \hat{H} steps elapsed since last inference **then**
 - 12: $\mathbf{a}_{t:t+H} \sim \pi_\theta(\mathbf{a} | \mathbf{o}_{t-T:t}, \mathcal{C}, \mathbf{a}_{t:})$ ▷ Async w/ RTC
 - 13: **end if**
 - 14: Execute \mathbf{a}_t
 - 15: **end for**
-

dropout, $\pi_{0.7}$ can also be used with classifier-free guidance (CFG) [109] for any part of the prompt, for example to guide the generated actions toward higher speeds. Concretely, each action denoising step follows

$$\nabla_{\mathbf{a}} \log \pi_\theta(\mathbf{a}_{t:t+H} | \mathbf{o}_t, \mathcal{C}_t) + \beta (\nabla_{\mathbf{a}} \log \pi_\theta(\mathbf{a}_{t:t+H} | \mathbf{o}_t, \mathcal{C}_t) - \nabla_{\mathbf{a}} \log \pi_\theta(\mathbf{a}_{t:t+H} | \mathbf{o}_t, \mathcal{C}_t^{\text{uncond}})),$$

where $\mathcal{C}_t^{\text{uncond}}$ denotes the set of context used in “unconditional” mode and β is the CFG weight. While any part of the context could be dropped out, we apply CFG on the episode metadata to elicit strong performance in dexterous tasks. We use moderate values of $\beta \in \{1.3, 1.7, 2.2\}$.

VIII. ROBOT SYSTEM DETAILS

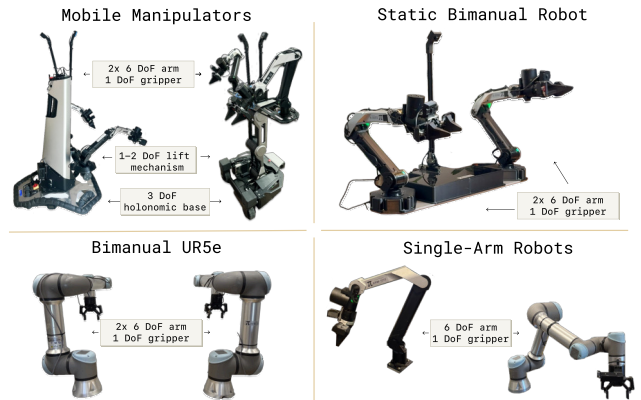


Fig. 4: **Illustrations of some of the robots in our experiments.** We evaluate $\pi_{0.7}$ on a variety of robots, including bimanual mobile manipulators (left), static bimanual robots (middle), and a bimanual UR5e setup (right) that we use for cross-embodiment experiments.

We deploy $\pi_{0.7}$ in a variety of robot platforms (Fig. 4), including bimanual mobile manipulators with two 6 DoF arms, static bimanual manipulators with lightweight 6 DoF arms (“BiPi”), and a bimanual UR5e system with Robotiq

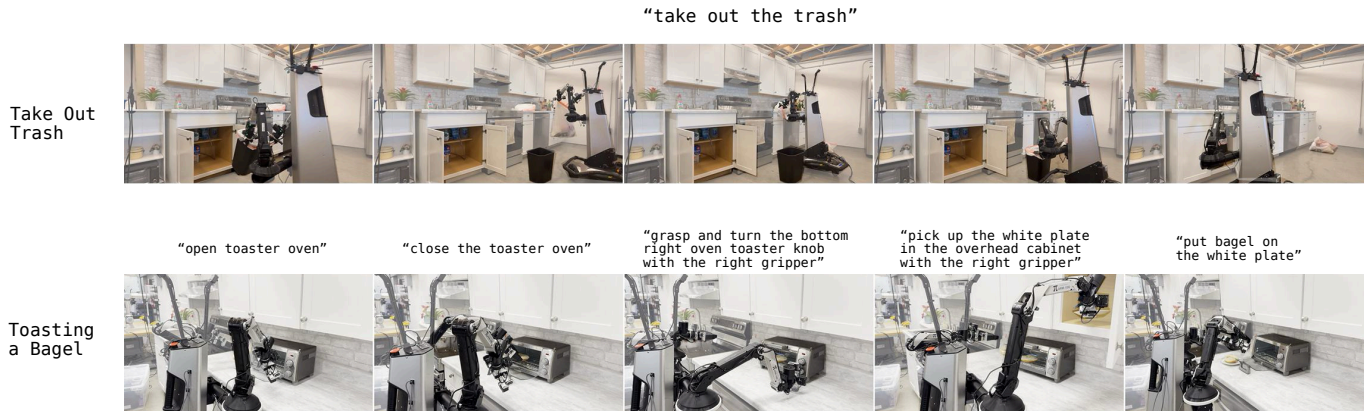


Fig. 5: **Illustration of selected evaluation tasks.** We evaluate $\pi_{0.7}$ on a number of tasks, and two of the more longer-horizon ones are visualized here. For some tasks such as “Take Out Trash”, we provide a coarse instruction like “take out the trash” and $\pi_{0.7}$ performs the full long-horizon task. For other tasks which do not appear in the training data for $\pi_{0.7}$ such as “Toasting a Bagel”, we can leverage the strong language following capabilities of $\pi_{0.7}$ to coach it to perform the task with a series of detailed instructions that break down the task step-by-step.

grippers, which we use for cross-embodiment experiments. Additional generalization and language following experiments use a single-arm 6 DoF system, using the same arms as the BiPi platform. Note that while a large fraction of our data is collected with arms that resemble the BiPi platform, the UR5e arms that we use for cross-embodiment testing are significantly longer, have a different morphology, and are much heavier. In practice, the UR5e arms need to employ a different manipulation strategy due to the shape of the arms, their positioning over the table (on the sides rather than at one edge), and the shape of the gripper and fingers, making cross-embodiment transfer to this platform a significant challenge. All manipulators use parallel-jaw grippers. The UR5e robots run at 20 Hz, while all other robots run at 50 Hz. Each robot has a front-facing camera as well as wrist camera on each arm, and the mobile robots also have a rear-facing camera. The action output of the $\pi_{0.7}$ model is applied on each robot using a simple PD controller. For commanding end-effector movement, we apply numerical inverse kinematics to convert target end-effector poses into target joint positions.

IX. EXPERIMENTAL EVALUATION

In our experiments, we evaluate how well $\pi_{0.7}$ can leverage diverse data sources to enable strong out-of-the-box performance, broad generalization, and more effective transfer, leveraging a variety of context modalities. Specifically, we study how well $\pi_{0.7}$ can perform complex and dexterous tasks out of the box, particularly in comparison with more specialized RL-finetuned models (Sec. IX-A), evaluate its ability to flexibly follow instructions to do a variety of different tasks (Sec. IX-B), study its transfer capabilities across embodiments (Sec. IX-C), and test its ability to compose skills in previously unseen ways to do new tasks (Sec. IX-D). Finally, we perform controlled experiments to study how the performance of $\pi_{0.7}$ scales with increased task and context diversity in our robot datasets (Sec. IX-E).

A. Out-of-the-box performance on challenging tasks

$\pi_{0.7}$ achieves high performance on dexterous tasks without task-specific post-training. In our first set of experiments, we study how well $\pi_{0.7}$ can master dexterous tasks that were seen in the training data, but where the goal is to perform these tasks as robustly and efficiently as possible. This is surprisingly difficult for prior robotic foundation models: often the best-performing policies are fine-tuned for specific downstream tasks, even if they use generalist pre-training [42, 50]. We aim to answer: can the general-purpose $\pi_{0.7}$ model match the performance of task-specific fine-tuned models on a variety of dexterous manipulation tasks?

We use the tasks shown in Fig. 6. These include the espresso making, box building, and laundry folding tasks that we previously used to evaluate the RL-trained $\pi_{0.6}^*$ models [50], where we can directly compare the speed and robustness of the single general-purpose $\pi_{0.7}$ model to the individual RL-finetuned specialist $\pi_{0.6}^*$ models. We also study a number of other dexterous tasks, including some tasks from our previous “Robot Olympics” experiments (making a peanut butter sandwich, turning a shirt inside-out, and driving through a door) and a number of additional dexterous tasks, such as fully slicing up a zucchini, peeling a few fruits and vegetables (zucchini, cucumbers, and carrots), and a long-horizon task that involves replacing a trash bag in a trash can. We find that $\pi_{0.7}$ achieves performance that is competitive with the RL specialists used in the $\pi_{0.6}^*$ release [50] for all of the tasks considered in the paper directly out of the box (Fig. 6, first row), and even outperform the specialists in throughput in the difficult laundry and box building tasks. Additionally, we compare $\pi_{0.7}$ to SFT specialists trained on top of $\pi_{0.6}$ for a number of other dexterous tasks, and find that $\pi_{0.7}$ is again able to closely match the performance of all specialist policies (Fig. 6, second row).

To understand how the training recipe of $\pi_{0.7}$ affects performance, we additionally compare $\pi_{0.7}$ with two ablations on the tasks from the $\pi_{0.6}^*$ release: $\pi_{0.7}$ (no eval data), which holds out all autonomous evaluation episodes from training

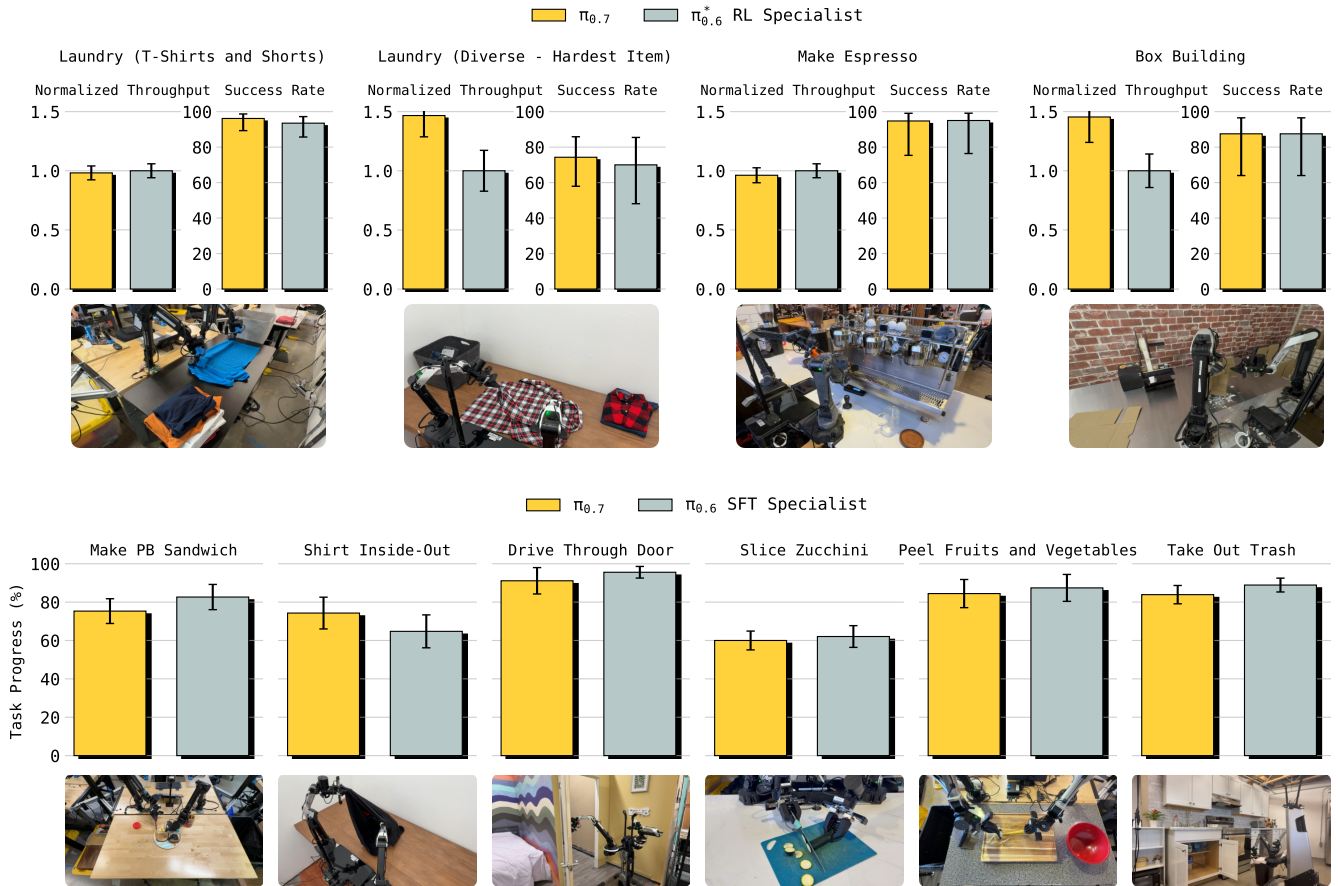


Fig. 6: **Out-of-the-box dexterity:** $\pi_{0.7}$ can perform a wide range of highly dexterous tasks directly out of the box. We consider tasks from $\pi_{0.6}^*$ [50] (top row) and a number of other dexterous tasks including ones from the “Robot Olympics” experiments (bottom row). For the tasks from $\pi_{0.6}^*$, we report success rate and normalized throughput (relative to the specialist model; raw throughput means successes per hour), while for other tasks we report task progress. We find that the same $\pi_{0.7}$ model can match the performance of the task-specific post-trained specialist policy from $\pi_{0.6}^*$ or $\pi_{0.6}$ for each of these tasks, and even achieve higher throughput than the RL specialists in diverse laundry folding and box building.

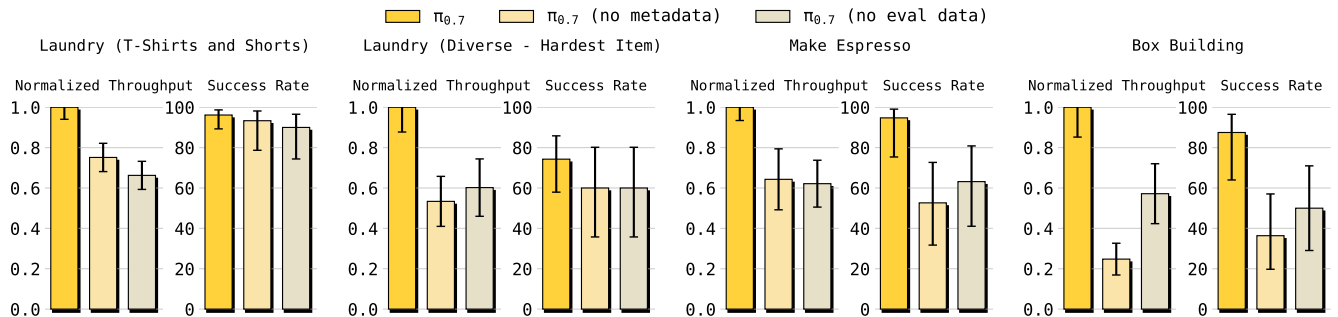


Fig. 7: **Impact of prompt composition and evaluation data on out-of-the-box performance:** We compare $\pi_{0.7}$ with two ablations: one that does not include episode metadata in the context, $\pi_{0.7}$ (no metadata), and another that does not include data from autonomous evaluation episodes during training, $\pi_{0.7}$ (no eval data). We find that $\pi_{0.7}$ outperforms both $\pi_{0.7}$ (no metadata) and $\pi_{0.7}$ (no eval data) across the board, with the gap most prominent in throughput. Throughput (successes/hour) here is normalized relative to $\pi_{0.7}$.

(and thus cannot benefit from distilling agent rollouts from strong, e.g. RL trained, policies), and $\pi_{0.7}$ (no metadata), which omits episode metadata from the context as shown in Fig. 7 for the tasks used in the $\pi_{0.6}^*$ release. Results suggest that $\pi_{0.7}$ significantly outperforms both $\pi_{0.7}$ (no eval data) and

$\pi_{0.7}$ (no metadata) on all tasks. Since policy evaluation data can vary widely in quality, training on this data, combined with rich metadata to disambiguate high and low quality behaviors, is critical for $\pi_{0.7}$'s strong performance on all of these challenging tasks.

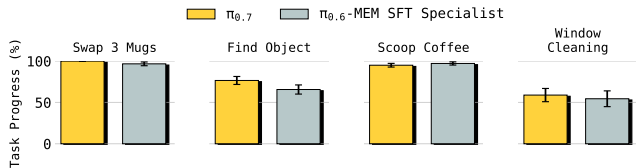


Fig. 8: **Tasks that require memory:** $\pi_{0.7}$ can also perform tasks that require explicitly keeping track of prior context, achieving similar or better performance compared to the specialist policies with memory fine-tuned to some of the tasks in the MEM paper [37].

$\pi_{0.7}$ **achieves high performance on tasks that require memory without fine-tuning.** In these experiments, we study how well $\pi_{0.7}$ can perform tasks that require explicitly keeping track of previous observations [37]. We compare the same single $\pi_{0.7}$ model out of the box to the task-specific fine-tuned versions of $\pi_{0.6}$ with memory used in Torne et al. [37], and find that $\pi_{0.7}$ can achieve similar or better performance to the fine-tuned specialists on all of these tasks (Fig. 8).

B. Instruction following

In the next set of experiments, we study how well $\pi_{0.7}$ can follow language instructions, including its ability to carry out a variety of different contexts and follow referential instructions that differ systematically from the training data. Our experiments focus specifically on performing tasks in messy environments where there are many possible tasks that the robot could perform, requiring $\pi_{0.7}$ to pay careful attention to the provided instruction in order to succeed. We find that $\pi_{0.7}$ displays instruction following capabilities that significantly improve upon our prior models $\pi_{0.5}$ [14] and $\pi_{0.6}$ [42].

$\pi_{0.7}$ **can be flexibly prompted to perform a wide variety of different tasks.** Language following has presented a notorious challenge for robotic foundation models, particularly with open-vocabulary instructions that do not correspond directly to ones seen in training. In these experiments, we aim to study the breadth of $\pi_{0.7}$ ’s capabilities, to answer: can $\pi_{0.7}$ better handle a wider variety of language instructions than prior models?

We evaluate $\pi_{0.7}$ on a diverse set of instructions across 4 unseen kitchens and 2 unseen bedrooms that were not present in the training data (Fig. 9). Each experiment tests whether the robot can follow a 3 to 6 step sequence of instructions to achieve a specific goal. The tasks require a variety of different realistic tasks in kitchen and bedroom environments, including re-arranging and tidying items, interacting with furniture, and cleaning up spills. This combination of new test environments and diverse instructions presents a major challenge for robotic foundation models, which can struggle to follow even simple instructions in seen environments. We find that $\pi_{0.7}$ is able to significantly outperform $\pi_{0.5}$ and $\pi_{0.6}$, with a high overall instruction following success rate.

$\pi_{0.7}$ **can handle out-of-distribution referential instructions.** Because of the breadth and diversity of our training data, it is generally difficult for us to quantify how *novel* the test instructions are. In the next set of experiments, we intentionally designed a set of instructions that are unusual, refer to objects in unconventional ways, or require understanding

spatial relations. In Fig. 10, we compare $\pi_{0.7}$ and prior models on a set of object re-arrangement instructions, broken up into *standard* and *complex* instructions. The standard instructions are phrased in a similar way to language instructions in the training data. The complex instructions use unusual language or complex spatial references, such as “pick up an object I would use to eat soup” or “pick up the fruit on the largest plate.” In these experiments, we also evaluate $\pi_{0.7}$ with and without subgoal images, which are generated by a lightweight world model as discussed previously. We can see that $\pi_{0.7}$ improves over prior models on the complex instructions, and the use of subgoal images further boosts its performance, importing semantic understanding from the world model.

$\pi_{0.7}$ **can follow instructions that go against dataset biases.** Dataset bias presents a major challenge for instruction following: if the robot always does the same thing in a given scene, a model trained on such data will often ignore language in these scenes, blindly copying the behavior seen in the data. In the next experiment, we construct scenarios that suffer from this problem, and test if we can prompt $\pi_{0.7}$ to go *against* the natural bias in the dataset. We constructed two tasks: “Reverse Bussing” and “Reverse Fridge to Microwave.” In our dataset, the “bussing” task involves putting trash in a trash bin and dishes in a bussing bin. For the “Reverse Bussing” task, the robot is asked to do the opposite: put the trash in the bussing bin, and the dishes in the trash. The “Fridge to Microwave” task requires taking food out of the fridge and putting it in the microwave, and we did not collect data going the other way. At test time, in the “reverse” version of this task (“Reverse Fridge to Microwave”), we prompt the robot to take the food from the microwave to the fridge, violating the bias in the dataset.

The results in Fig. 11 show that $\pi_{0.7}$ significantly improves over prior models on these tasks. This suggests that $\pi_{0.7}$ has significantly better language following capabilities, and pays enough attention to the instructions to overcome the bias in the data for these tasks. On the “Reverse Fridge to Microwave” task, conditioning on generated subgoal images ($\pi_{0.7}$ (GC)) is critical for success, as the world model can generate subgoals based on textual instructions effectively by leveraging web-scale image generation pre-training.

C. Cross-embodiment transfer

While a number of models have used data from multiple different robot embodiments [3, 5, 8, 80], zero-shot transfer of complex tasks from a source embodiment to a target robot that has never seen the task presents a major challenge. In these experiments, we aim to study whether cross-embodiment transfer is an *emergent* property of $\pi_{0.7}$. Namely, can $\pi_{0.7}$ directly transfer capabilities to robot embodiments where no task-specific data was ever collected?

We find that for several tasks, $\pi_{0.7}$ succeeds on target embodiments entirely out of the box, on tasks for which the target robot has no training data. For more modest embodiment differences, the $\pi_{0.5}$ and $\pi_{0.6}$ models also show some amount of emergent cross-embodiment transfer. However, as the gap

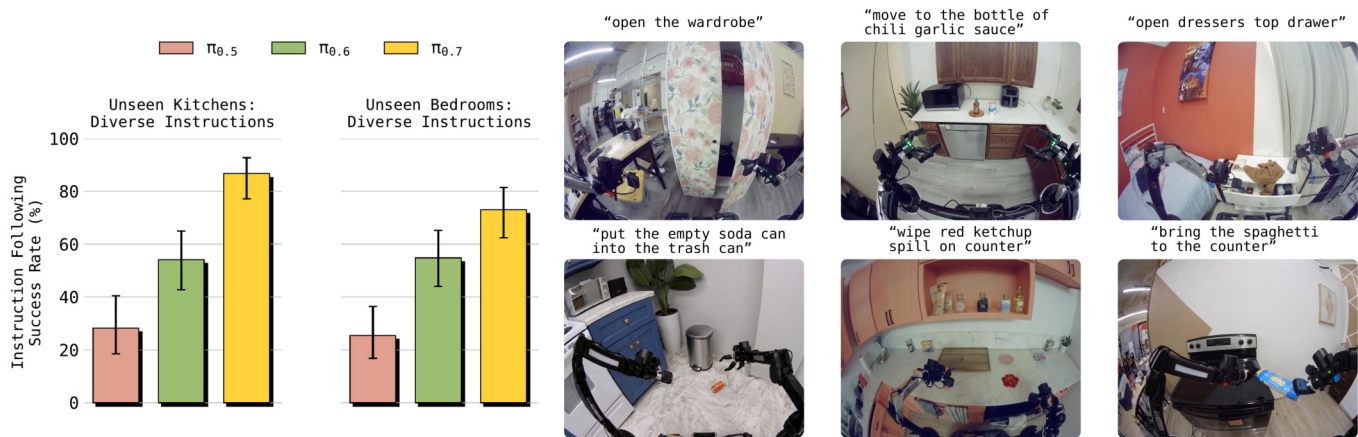


Fig. 9: **Broad instruction following in novel environments:** We evaluate $\pi_{0.7}$ on 14 instruction following scenarios, each of which involve following a sequence of 3-6 open-ended instructions, across 4 unseen kitchen and 2 unseen bedroom environments. We report the instruction following success rate, the percentage of total instructions that were correctly followed across all evaluations. We find that $\pi_{0.7}$ significantly outperforms $\pi_{0.5}$ and $\pi_{0.6}$ across the board, attaining high absolute success rates.

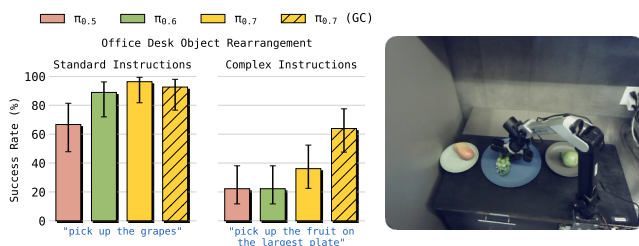


Fig. 10: **Following complex referential instructions:** $\pi_{0.7}$ and prior models all succeed on the simpler re-arrangement instructions (instructions include “pick up the spoon”, “put the spoon to the left of the fork” and “put the spoon to the right of the fork”), but $\pi_{0.7}$ performs significantly better on the complex and unusual instructions (instructions include “pick up the largest bowl on the table”, “pick up the object I would use to eat soup” and “pick up the fruit on the largest plate”). Including subgoal images generated by a lightweight world model ($\pi_{0.7}$ (GC)) further boosts instruction following performance, making $\pi_{0.7}$ significantly more capable at following complex instructions.

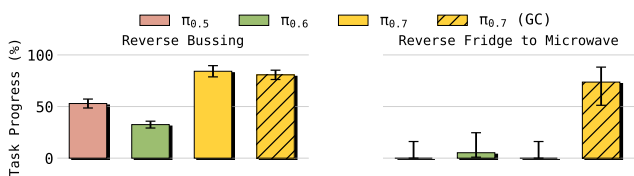


Fig. 11: **Breaking dataset biases by following instructions:** the improved language-following performance of $\pi_{0.7}$ enables it to break strong dataset biases. Prior models struggle on these data bias challenge tasks, which require following instructions that contradict the pattern in the data (e.g., putting dishes in the trash and trash in the dish bin). $\pi_{0.7}$, however, can follow instructions sufficiently well that it can break these strong biases to still perform the task. Notably, for the “Reverse Fridge to Microwave” task, including subgoal images from the world model in the context ($\pi_{0.7}$ (GC)) is critical for success.

in the robot morphology increases, effectively transferring complex skills requires more significant changes in strategy. In these cases, $\pi_{0.7}$ significantly outperforms the prior models, and even matches the “zero-shot” performance of human teleoperators for the shirt folding task, as we discuss below. Experimental results below apply joint-space control, as we

find that end-effector control does not yield noticeable gains in performance with our prior models (Appendix E).

Zero-shot cross-embodiment transfer for object rearrangement tasks. We first study a set of simpler object rearrangement tasks, where we test $\pi_{0.7}$ on a robot other than the one that was used to collect data for that task. The results are presented in Fig. 12. For the first task, “Table Setting,” the data was collected with several different robot types, including mobile, static, and single-arm systems. This is the most favorable setting for cross-embodiment transfer, since the models can infer the common structure of the task from multiple different robots. When evaluated on a static bimanual robot system, we find that all methods show strong signs of cross-embodiment transfer. However, when we increase the embodiment gap more significantly and test policies on the smaller static bimanual platform when all data was collected on the larger UR5e bimanual platform (“Bag In Backpack” and “Organize Tupperware” tasks), we find that the performance of $\pi_{0.5}$ degrades significantly, while both $\pi_{0.6}$ and $\pi_{0.7}$ still are able to achieve strong performance. We then increase the embodiment gap even more, studying transfer for a task where data was collected on the smaller static bimanual platform and evaluated on a *single-arm* UR5e platform (“Shirt Bagging” task). This necessitates a significantly different strategy, as the target robot has only one arm, is much larger, and also heavier. Here, $\pi_{0.7}$ significantly outperforms the prior models.

Notably, successful transfer often requires the policy to discover new manipulation strategies suited to the target morphology rather than simply replicate the source behavior. For example, the shorter static bimanual robot must use one arm to hold the bag open while the other performs insertion, whereas the taller UR5e arm can accomplish the same task with a single-arm pick-and-place (Fig. 13 (a)). Despite this significant morphological gap, the model applies the appropriate strategy for each embodiment, demonstrating cross-embodiment transfer that goes beyond mimicking the original robot’s motion. **Zero-shot cross-embodiment transfer for dexterous tasks.** Dexterous tasks, such as laundry folding, present a bigger

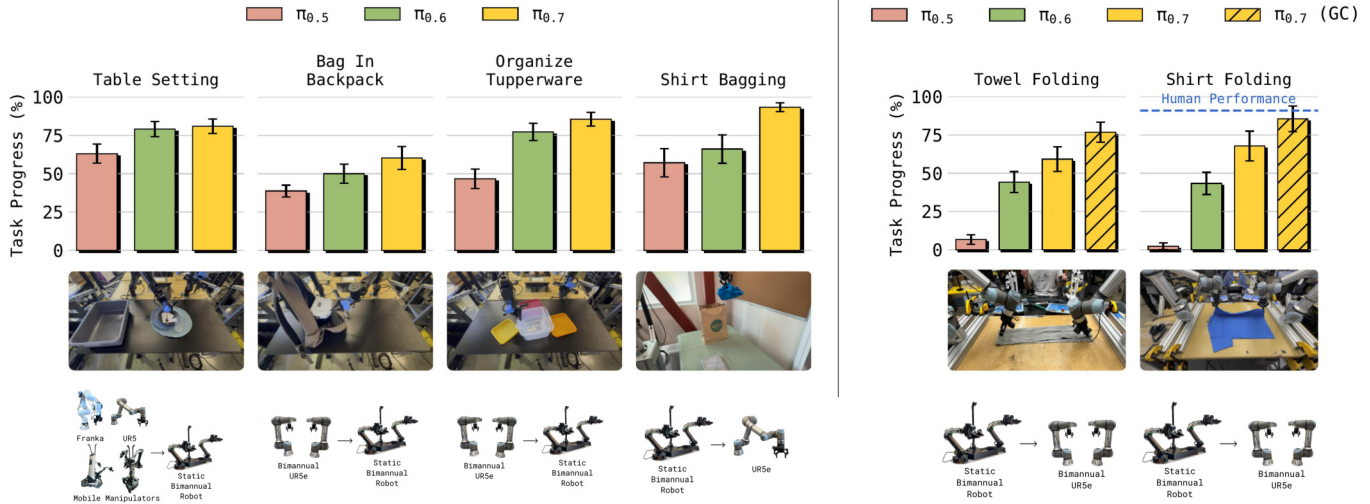


Fig. 12: **Cross-embodiment transfer**: Left: Both $\pi_{0.7}$ and prior models achieve strong cross-embodiment transfer directly out of the box on simpler rearrangement or repositioning style tasks. For example, for the “Table Setting” task, data was collected with a variety of different robots, and the task was tested on the static bimanual robot. All of the models performed well. For tasks that required transfer from the bimanual UR5e robot to the smaller static bimanual robot (“Bag In Backpack” and “Organize Tupperware”), the embodiment gap is larger, as there is only one source robot that is bigger and heavier than the target robot. Here, $\pi_{0.5}$ performed very poorly, but $\pi_{0.6}$ still performs quite well. Transferring from the smaller static bimanual robot to the UR5e for the “Shirt Bagging” task introduces the largest embodiment gap, and here the $\pi_{0.7}$ model significantly outperforms prior models. Right: for the more dexterous tasks that require folding towels and t-shirts, the embodiment gap poses an even greater challenge. Data for these tasks was collected with the smaller static bimanual robot, and the task was evaluated on the larger bimanual UR5e platform. $\pi_{0.7}$ was able to transfer this task successfully, and improved even more when using visual subgoal images generated by our lightweight world model. In fact, task progress matches the “zero-shot” performance of our most experienced human teleoperators, who have operated a diverse set of robots and attempted this task on the UR5e for the first time.

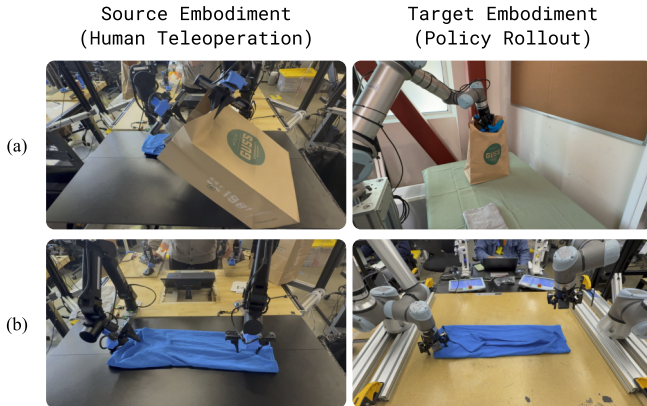


Fig. 13: **Cross-embodiment transfer produces emergent strategies adapted to the target embodiment**. (a) On the source robot, human teleoperators use one arm to hold the bag open while the other performs insertion. On the UR5e target robot, $\pi_{0.7}$ instead discovers a single-arm pick-and-place strategy suited to the robot’s greater reach. (b) Human teleoperators approach the shirt with a tilted end-effector on the source robot, while $\pi_{0.7}$ produces vertical grasps on the UR5e, which is more suitable for the larger robot’s arm placement. In both cases, the policy goes beyond replicating source behavior, discovering manipulation strategies for the task that are better suited to the target embodiment.

challenge for cross-embodiment transfer. Such tasks require more precise manipulation skills than just grasping and repositioning objects. Successfully folding a t-shirt requires a sequence of precise grasps and placements, and the correct grasp angle might vary with the robot’s reachable workspace and gripper orientation. Most of our folding data was collected with lightweight static bimanual robots (see Fig. 4). We

did not collect any laundry folding data with the bimanual UR5e system. The morphology of this system, its reachable workspace, and its dynamics (e.g., higher internal inertia) differ significantly from the robots with which we collected laundry folding data. We found the UR5e generally harder to teleoperate and less suitable for very precise grasps, suggesting that folding laundry with this robot requires a change in the manipulation strategy.

$\pi_{0.7}$ was able to successfully fold both towels and shirts on the bimanual UR5e system (Fig. 12, right). On the source robot (the static bimanual robot in Fig. 4), human operators often approach the cloth with a tilted end-effector to secure the fabric against the table before lifting. On the UR5e, $\pi_{0.7}$ instead uses a vertical grasp that is better suited to the arm’s kinematics — a strategy that is different from the training data on the source robot but more suitable for the target embodiment (Fig. 13 (b)). We also find that using subgoal image generation with our world model significantly improves performance (labeled $\pi_{0.7}$ (GC) in the plots), as the world model can more effectively construct visual analogies between the source and target robots. The generated subgoals predict what kind of grasps and clothing configurations would be reasonable for the target robot, and the model incorporates this additional context as a hint for selecting better actions.

To contextualize these results, we conducted a human subject study with 10 experienced teleoperators (mean 375 hours of teleoperation experience across all robots, all within the top 2% by experience). Like the policy, these operators had extensive experience on the source embodiment, but had never



Fig. 14: **Example of language coaching:** We can “teach” a new task to $\pi_{0.7}$ by providing step-by-step verbal instructions. Because of its language following ability, $\pi_{0.7}$ can perform new tasks successfully under user instruction, and these instructions can then be used to train a high-level policy that prompts $\pi_{0.7}$ so that it can perform the task fully autonomously.

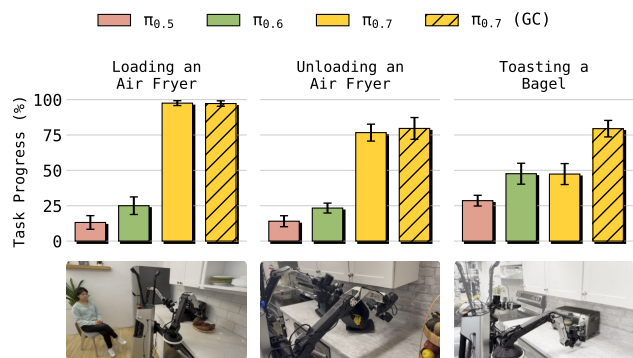


Fig. 15: **Coaching to perform new long-horizon tasks:** Because $\pi_{0.7}$ can follow language instructions effectively, even for unfamiliar skills, it can be “coached” to perform a number of unseen, longer horizon tasks both when conditioned on language and generated subgoal images ($\pi_{0.7}$ (GC)). Prior models generally lack the language following ability needed to follow the coaching commands, and thus achieve very poor performance.

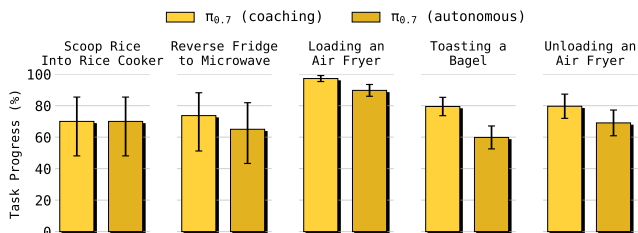


Fig. 16: **Acquiring new autonomous capabilities with coaching:** We can use the coaching episodes collected for a number of different unseen tasks to train a high-level policy to automatically prompt $\pi_{0.7}$ in accordance with the coaching episodes. This allows us to create fully autonomous policies for these tasks ($\pi_{0.7}$ (autonomous)) that closely match the performance of the policy with live human coaching ($\pi_{0.7}$ (coaching)) *without collecting any additional data with teleoperation or any other kind of low-level actions.*

attempted shirt folding on the UR5e bimanual system, making this a zero-shot cross-embodiment transfer setting for both humans and the policy. The human operators achieved 90.9% task progress and an 80.6% success rate, while $\pi_{0.7}$ achieved 85.6% task progress and an 80% success rate, demonstrating performance comparable to these expert operators. The strong cross-embodiment transfer performance highlighted by this comparison is scientifically exciting, and it has a practical implication: dexterous skills may be transferred *from* lightweight, low-cost platforms that are easy to teleoperate *to* high-payload industrial arms, where collecting human demonstration data is

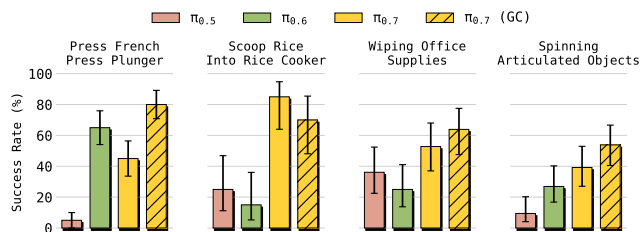


Fig. 17: **Performing new short-horizon tasks:** $\pi_{0.7}$ can perform a number of new short-horizon tasks directly out of the box, including scooping rice into a rice cooker, spinning various objects such as a gear set and desk fan, and wiping down objects with a cloth, such as a ruler and headphones, despite no data being collected for any of these tasks. $\pi_{0.7}$ shows roughly equally strong performance when conditioned directly on language instructions ($\pi_{0.7}$) or generated image goals ($\pi_{0.7}$ (GC)).

substantially more expensive and difficult. More details on the human study setup and results are provided in Appendix F.

D. Compositional task generalization

In the next set of experiments, we study how well $\pi_{0.7}$ can perform new tasks by compositionally generalizing over the skills seen in training. This has been viewed as a kind of “grand challenge” for robotic foundation models: while prior models have demonstrated generalization over semantic concepts (e.g., reaching for an object with an unseen textual label), performing new tasks has proven elusive.

We found that for some short-horizon tasks, $\pi_{0.7}$ can work well completely out of the box, despite no data being collected explicitly for these tasks. These tasks involve manipulating unfamiliar objects in new ways, such as wiping down a pair of headphones with a cloth, or spinning a desk fan (Fig. 17). For longer-horizon, more complex tasks, we find that we can actually use the instruction following capabilities of $\pi_{0.7}$ to “coach” it through the task with language. This provides an exciting new way to teach $\pi_{0.7}$ new tasks without collecting any additional training data.

$\pi_{0.7}$ can perform new short-horizon tasks out of the box.

We find that $\pi_{0.7}$ can perform short-horizon tasks such as pressing the plunger on a french press, scooping rice into a rice cooker, wiping down common office objects, and spinning various articulated items directly out of the box, despite no robot data being collected specifically for any of these tasks (Fig. 17). This is quite exciting — since $\pi_{0.7}$ can flexibly compose a wide variety of skills, we can often get it to perform

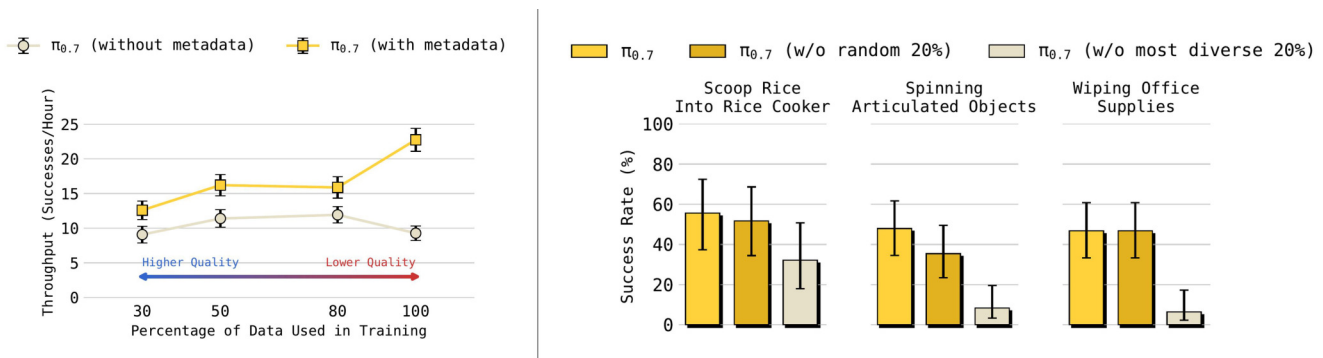


Fig. 18: **Scaling of generalization performance with diverse context and data:** Left: We find that $\pi_{0.7}$ (with metadata) can continuously improve its performance when it is trained on larger datasets, even when the average quality of the data actually decreases. By contrast, without training on rich conditioning information, we find that $\pi_{0.7}$ (without metadata) actually can degrade in performance as more lower quality data is introduced. Right: When $\pi_{0.7}$ is trained without our robot data with the highest task diversity, its performance degrades substantially. This suggests that $\pi_{0.7}$ is able to utilize the task diversity in our robot data to drive substantial improvements in compositional task generalization.

new, simple tasks by just prompting it to get the desired behaviors.

$\pi_{0.7}$ can be coached to perform new longer horizon tasks purely with language. While $\pi_{0.7}$ can be directly prompted to perform new short-horizon tasks, simply asking the robot to perform an unseen long-horizon task like cooking a sweet potato would not work: although $\pi_{0.7}$ exhibits a high level of out-of-the-box generalization, these tasks are simply too complex, requiring up to 5 minutes of interaction with multiple stages. However, $\pi_{0.7}$ ’s language following abilities provide us with an exciting new path to teach the model such tasks: instead of providing demonstration data for each complex skill that we want the robot to learn, we can instead “coach” the robot to perform the new task with language, a bit like how one might teach the task to a person (Fig. 14). We set up several realistic multi-stage kitchen tasks: (1) “Loading an Air Fryer”: using an air fryer to cook a sweet potato; (2) “Unloading an Air Fryer”: dumping items out of an air fryer, and (3) “Toasting a Bagel”: using a toaster to toast up a bagel. In each case, our robot data did not contain any training episodes for this task, although similar appliances were seen in different contexts in human data and external datasets. However, a person can walk the robot through task with step-by-step instructions, such as “pick up the sweet potato” and “open the air fryer.” We present the results of coaching $\pi_{0.7}$ and comparisons in Fig. 15. Critically, none of the models have any *action-level* data of these particular tasks, and in the “coaching” episodes, the environment and task are entirely unseen. We find that $\pi_{0.7}$ can be coached much more effectively than prior methods to perform all of these tasks, and can perform even more effectively when conditioned on generated subgoal images.

Coaching data can endow $\pi_{0.7}$ with new capabilities. Since $\pi_{0.7}$ can be coached to perform new tasks, we can actually use the step-by-step instructions in the coaching data to train a high-level language policy which can prompt $\pi_{0.7}$ with the appropriate language instructions as it performs the task. This gives $\pi_{0.7}$ the ability to perform fully unseen, long-horizon tasks without collecting any additional teleoperation data. We show the results of this experiment in Fig. 16 for five different

tasks. For all of these tasks, we find that we can successfully train an autonomous policy ($\pi_{0.7}$ (autonomous)) that can roughly match the performance of the coaching episodes ($\pi_{0.7}$ (coaching)) that were collected by prompting the model to perform the task.

E. Can $\pi_{0.7}$ learn effectively from diverse and mixed-quality data?

In our last set of experiments, we perform a set of controlled ablation studies to understand whether $\pi_{0.7}$ can effectively leverage large and diverse datasets, and whether its performance improves with dataset diversity. These questions are difficult to answer definitively, since the performance of such a model depends on a large number of factors, and very large datasets are very difficult to slice cleanly so as to ablate “diversity”. To provide some understanding of these questions, we first study whether $\pi_{0.7}$ continues to improve on seen tasks when trained on increasingly larger but more mixed-quality datasets. Then, we study whether $\pi_{0.7}$ can leverage datasets with high task diversity to drive improvements in generalization.

$\pi_{0.7}$ can effectively learn from mixed-quality data: Effectively learning from diverse robot data has so far been a significant challenge in training robotic policies. Designers will often carefully filter or curate the data to pull high-quality datasets with consistent strategies [110, 111]. However, data filtering is laborious, task-specific, and ends up throwing away a lot of valuable information. In these experiments, we aim to answer: can $\pi_{0.7}$ learn more from data with diverse manipulation strategies?

To study this question, we consider the “Laundry (T-Shirts and Shorts)” task that we studied in Fig. 6. We annotated the data collected by human operators based on fold quality and speed, and split our dataset into 4 buckets consisting of (1) the top 30% by quality and speed, (2) the top 50%, (3) the top 80%, and (4) all of the data. We then train new $\pi_{0.7}$ models from scratch for data from each of the four buckets, using episode metadata or without it (8 models in total). We find that while $\pi_{0.7}$ (without metadata) can actually get worse when trained on larger, mixed-quality datasets, $\pi_{0.7}$ (with metadata)

is able to continuously improve as we train on more data, even though the dataset size increases corresponds to a decrease in average data quality (Fig. 18, left). This suggests that our diverse prompting method effectively makes the model design *more scalable*, in the sense that it benefits more from larger datasets, even when these larger datasets actually consist of lower quality data that harm models trained in the usual way. Episode metadata effectively disambiguates the different data quality and strategies within the large-scale datasets during $\pi_{0.7}$ training, and enables prompting for the desired mode of behavior at test time.

Can $\pi_{0.7}$ translate increased dataset diversity into better generalization performance?

To study this question, we compare $\pi_{0.7}$ with the following ablations on some of the short-horizon unseen tasks from Fig. 17.

- $\pi_{0.7}$ (w/o most diverse 20%): $\pi_{0.7}$ but with the 20% of our data with the highest task diversity removed.
- $\pi_{0.7}$ (w/o random 20%): $\pi_{0.7}$ with a randomly sampled 20% of our data removed to serve as a data-controlled comparison to $\pi_{0.7}$ (w/o most diverse 20%)

The comparison between $\pi_{0.7}$ (w/o most diverse 20%) and $\pi_{0.7}$ (w/o random 20%) allows us to understand the impact of data with high task diversity in a controlled way, as both models are trained on the same quantity of data. We find that across all tasks, $\pi_{0.7}$ and $\pi_{0.7}$ (w/o random 20%) significantly outperform $\pi_{0.7}$ (w/o most diverse 20%), demonstrating that $\pi_{0.7}$ is able to effectively ingest data with high task diversity and translate that data into performance improvements on short-horizon, unseen tasks (Fig. 18, right).

X. DISCUSSION

We presented $\pi_{0.7}$, a general-purpose robot foundation model that exhibits out-of-the-box compositional generalization, effective language following, and task performance that is competitive with more specialized models that are fine-tuned to individual dexterous tasks. At the core of $\pi_{0.7}$ is a diverse prompting strategy, inspired by prompt expansion, where additional information about the episode is provided to the model during training and, optionally, at test time. This additional information includes more detailed language, episode metadata, and subgoal images. Our experiments show that by using diverse prompting and a larger and more diverse dataset, $\pi_{0.7}$ can represent policies of many different qualities, and distill specialist performance back into one pre-trained model. $\pi_{0.7}$ acquires a number of emergent capabilities, such as the ability to transfer skills across robots, follow complex language commands, and generalize compositionally, recombining skills in new ways to solve new tasks.

While $\pi_{0.7}$ generalizes broadly, the success rate for zero-shot generalization is (unsurprisingly) lower than in-distribution tasks: while seen tasks often have success rates in excess of 90%, unseen tasks or unseen task-robot combinations have success rates in the 60-80% range. An exciting direction for future work is to leverage the high steerability of $\pi_{0.7}$ to efficiently learn from data in the test task, for example with

more detailed language coaching or even with autonomous reinforcement learning.

A perhaps surprising limitation of our experiments is that it’s practically difficult when training on such large and diverse datasets to definitively determine which tasks are truly “seen” or “unseen”: while some of our generalization experiments (e.g., those in Sec. IX-D) use tasks for which we did not deliberately collect data, our dataset contains so many different scenes and behaviors that potentially related skills may well be present elsewhere in the data, either with a different label, or incidentally as part of performing other tasks. In many ways this mirrors the challenge of understanding generalization with large language models: determining what is truly novel becomes difficult, and the model may well be achieving generalization primarily by “remixing” skills and behaviors from other situations. However, we would posit that this in fact is the essence of *compositional generalization*. Practically, whether the behaviors are truly new or merely novel combinations of seen parts, the ramifications are similar: instead of deliberating collecting targeted data for each new task that we want the robot to solve, a model that provides compositional generalization would allow the user to simply *prompt* it to do the desired task. Models that can enable such compositional generalization at scale would transform how we approach robotic learning, making it possible to prompt, coach, and explain things to a robot rather than needing to collect additional action data.

ACKNOWLEDGEMENTS

We thank our robot operators for data collection, evaluations, logistics, and video recording, and our technicians for robot maintenance and repair. See Appendix A for a full contributions statement.

REFERENCES

- [1] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 3
- [2] Scott Reed, Konrad Żołna, Emilio Parisotto, et al. A generalist agent. *Transactions on Machine Learning Research (TMLR)*, 2022.
- [3] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 10
- [4] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *International Conference on Learning Representations (ICLR)*, 2025.

- [5] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *NeurIPS*, 2024. 10
- [6] TRI LBM Team, Jose Barreiros, Andrew Beaulieu, Aditya Bhat, Rick Cory, Eric Cousineau, Hongkai Dai, Ching-Hsin Fang, Kunimatsu Hashimoto, Muhammad Zubair Irshad, Masha Itkina, Naveen Kuppaswamy, Kuan-Hui Lee, Katherine Liu, Dale McConachie, Ian McMahan, Haruki Nishimura, Calder Phillips-Grafflin, Charles Richter, Paarth Shah, Krishnan Srinivasan, Blake Wulfe, Chen Xu, Mengchao Zhang, Alex Alspach, Maya Angeles, Kushal Arora, Vitor Campagnolo Guizilini, Alejandro Castro, Dian Chen, Ting-Sheng Chu, Sam Creasey, Sean Curtis, Richard Denitto, Emma Dixon, Eric Dusel, Matthew Ferreira, Aimee Goncalves, Grant Gould, Damrong Guoy, Swati Gupta, Xuchen Han, Kyle Hatch, Brendan Hathaway, Allison Henry, Hillel Hochshtein, Phoebe Horgan, Shun Iwase, Donovan Jackson, Siddharth Karamcheti, Sedrick Keh, Joseph Masterjohn, Jean Mercat, Patrick Miller, Paul Mitiguy, Tony Nguyen, Jeremy Nimmer, Yuki Noguchi, Reko Ong, Aykut Onol, Owen Pfannenstiehl, Richard Poyner, Leticia Pribe Mendes Rocha, Gordon Richardson, Christopher Rodriguez, Derick Seale, Michael Sherman, Mariah Smith-Jones, David Tago, Pavel Tokmakov, Matthew Tran, Basile Van Hoorick, Igor Vasiljevic, Sergey Zakharov, Mark Zolotas, Rares Ambrus, Kerri Fetzer-Borelli, Benjamin Burchfiel, Hadas Kress-Gazit, Siyuan Feng, Stacie Ford, and Russ Tedrake. A careful examination of large behavior models for multitask dexterous manipulation, 2025. URL <https://arxiv.org/abs/2507.05331>. 3
- [7] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023. 3
- [8] Open X-Embodiment Collaboration, Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, Antonin Raffin, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Brian Ichter, Cewu Lu, Charles Xu, Chelsea Finn, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Chuer Pan, Chuyuan Fu, Coline Devin, Danny Driess, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Federico Ceola, Fei Xia, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Giulio Schiavi, Hao Su, Hao-Shu Fang, Haochen Shi, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jaehyung Kim, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jiajun Wu, Jialin Wu, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jitendra Malik, Jonathan Tompson, Jonathan Yang, Joseph J. Lim, João Silvério, Junhyek Han, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Zhang, Keyvan Majd, Krishan Rana, Krishnan Srinivasan, Lawrence Yunliang Chen, Lerrel Pinto, Liam Tan, Lionel Ott, Lisa Lee, Masayoshi Tomizuka, Maximilian Du, Michael Ahn, Mingtong Zhang, Mingyu Ding, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Norman Di Palo, Nur Muhammad Mahi Shafullah, Oier Mees, Oliver Kroemer, Pannag R Sanketi, Paul Wohlhart, Peng Xu, Pierre Sermanet, Priya Sundaresan, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Sherry Moore, Shikhar Bahl, Shivin Dass, Shuran Song, Sichun Xu, Siddhant Haldar, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Sudeep Dasari, Suneel Belkhale, Takayuki Osa, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xuanlin Li, Yao Lu, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yueh hua Wu, Yujin Tang, Yuke Zhu, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zhuo Xu, and Zichen Jeff Cui. Open X-Embodiment: Robotic learning datasets and RT-X models, 2023. 10
- [9] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
- [10] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Motukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 3
- [11] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *arXiv preprint*

- arXiv:2409.12514*, 2024.
- [12] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.
- [13] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025. 3
- [14] Kevin Black, Noah Brown, James Darphinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *9th Annual Conference on Robot Learning*, 2025. 3, 4, 10
- [15] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.
- [16] Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea open-world dataset and g0 dual-system vla model. *arXiv preprint arXiv:2509.00576*, 2025.
- [17] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. *International Conference on Learning Representations (ICLR)*, 2024.
- [18] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.
- [19] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model. *Robotics: Science and Systems (RSS)*, 2025.
- [20] Johan Bjorck et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.
- [21] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.
- [22] AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.
- [23] Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, and Yi Xu. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. *arXiv preprint arXiv:2502.14420*, 2025. 3
- [24] Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *International Conference on Learning Representations (ICLR)*, 2026. 3
- [25] Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas, 2025. URL <https://arxiv.org/abs/2512.15692>. 3
- [26] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. 3
- [27] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *International Conference on Learning Representations (ICLR)*, 2024.
- [28] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 3
- [29] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024. 3
- [30] Ajay Sridhar, Jennifer Pan, Satvik Sharma, and Chelsea Finn. Memer: Scaling up memory for robot control via experience retrieval. *arXiv preprint arXiv:2510.20328*, 2025.
- [31] Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.
- [32] Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2505.11917*, 2025.
- [33] Haoquan Fang, Markus Grotz, Wilbert Pumacay, Yi Ru

- Wang, Dieter Fox, Ranjay Krishna, and Jiafei Duan. Sam2act: Integrating visual foundation model with a memory architecture for robotic manipulation. *arXiv preprint arXiv:2501.18564*, 2025.
- [34] Hao Li, Shuai Yang, Yilun Chen, Yang Tian, Xiaoda Yang, Xinyi Chen, Hanqing Wang, Tai Wang, Feng Zhao, Dahua Lin, et al. Cronusvla: Transferring latent motion across time for multi-frame prediction in manipulation. *arXiv preprint arXiv:2506.19816*, 2025.
- [35] Zongzheng Zhang, Haobo Xu, Zhuo Yang, Chenghao Yue, Zehao Lin, Huan-ang Gao, Ziwei Wang, and Hao Zhao. Ta-vla: Elucidating the design space of torque-aware vision-language-action models. *arXiv preprint arXiv:2509.07962*, 2025.
- [36] Huiwon Jang, Sihyun Yu, Heeseung Kwon, Hojin Jeon, Younggyo Seo, and Jinwoo Shin. Contextvla: Vision-language-action model with amortized multi-frame context. *arXiv preprint arXiv:2510.04246*, 2025.
- [37] Marcel Torne, Karl Pertsch, Homer Walke, Kyle Vedder, Suraj Nair, Brian Ichter, Allen Z Ren, Haohuan Wang, Jiaming Tang, Kyle Stachowicz, et al. Mem: Multi-scale embodied memory for vision language action models. *arXiv preprint arXiv:2603.03596*, 2026. 3, 4, 6, 10, 22
- [38] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, et al. Do as i can, not as i say: Grounding language in robotic affordances. *Conference on Robot Learning (CoRL)*, 2022. 3
- [39] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [40] Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, et al. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025. 3
- [41] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *Computer Vision and Pattern Recognition (CVPR)*, 2025. 3
- [42] Physical Intelligence Team. $\pi_{0.6}$ model card, 2025. 3, 8, 10
- [43] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024. 3
- [44] Xiaopeng Lin, Shijie Lian, Bin Yu, Ruoqi Yang, Changti Wu, Yuzhuo Miao, Yurun Jin, Yukun Shi, Cong Huang, Bojun Cheng, and Kai Chen. Physbrain: Human egocentric data as a bridge from vision language models to physical intelligence, 2025. URL <https://arxiv.org/abs/2512.16793>.
- [45] Simar Kareer, Karl Pertsch, James Darphinian, Judy Hoffman, Danfei Xu, Sergey Levine, Chelsea Finn, and Suraj Nair. Emergence of human to robot transfer in vision-language-action models. *arXiv preprint arXiv:2512.22414*, 2025. 3
- [46] Zuolei Li, Xingyu Gao, Xiaofan Wang, and Jianlong Fu. Latbot: Distilling universal latent actions for vision-language-action models, 2025. URL <https://arxiv.org/abs/2511.23034>.
- [47] Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, Hongxu Yin, Sifei Liu, Song Han, Yao Lu, and Xiaolong Wang. Egovla: Learning vision-language-action models from egocentric human videos, 2025. URL <https://arxiv.org/abs/2507.12440>.
- [48] Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, Yicheng Feng, and Zongqing Lu. Being-h0.5: Scaling human-centric robot learning for cross-embodiment generalization, 2026. URL <https://arxiv.org/abs/2601.12993>.
- [49] Chubin Zhang, Jianan Wang, Zifeng Gao, Yue Su, Tianru Dai, Cai Zhou, Jiwen Lu, and Yansong Tang. Clap: Contrastive latent action pretraining for learning vision-language-action models from human videos, 2026. URL <https://arxiv.org/abs/2601.04061>. 3
- [50] Physical Intelligence Team. $\pi_{0.6}^*$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025. 3, 8, 9
- [51] Charles Xu, Qiyang Li, Jianlan Luo, and Sergey Levine. Rldg: Robotic generalist policy distillation via reinforcement learning. *arXiv preprint arXiv:2412.09858*, 2024.
- [52] Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Yuqi Xie, Fengyuan Hu, Jimmy Wu, Zhengyi Luo, Linxi Fan, et al. Self-improving vision-language-action models with data generation via residual rl. *arXiv preprint arXiv:2511.00091*, 2025. 3
- [53] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pages 892–909. PMLR, 2023. 3
- [54] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*, 2022.
- [55] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.
- [56] Chethan Bhateja, Derek Guo, Dibya Ghosh, Anikait Singh, Manan Tomar, Quan Vuong, Yevgen Chebotar, Sergey Levine, and Aviral Kumar. Robotic offline rl from internet videos via value-function pre-training.

- arXiv preprint arXiv:2309.13041*, 2023.
- [57] Yuxiang Zhou, Yusuf Aytar, and Konstantinos Bousmalis. Manipulator-independent representations for visual imitation. *arXiv preprint arXiv:2103.09016*, 2021.
- [58] Homanga Bharadhwaj, Abhinav Gupta, and Shubham Tulsiani. Visual affordance prediction for guiding robot exploration. *arXiv preprint arXiv:2305.17783*, 2023. **3**
- [59] Hongyi Chen, Tony Dong, Tiancheng Wu, Liquan Wang, Yash Jangir, Yaru Niu, Yufei Ye, Homanga Bharadhwaj, Zackory Erickson, and Jeffrey Ichnowski. Dexterous manipulation policies from rgb human videos via 3d hand-object trajectory reconstruction. *arXiv preprint arXiv:2602.09013*, 2026. **3**
- [60] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In *Conference on Robot Learning*, pages 654–665. PMLR, 2023.
- [61] Homanga Bharadhwaj, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Zero-shot robot manipulation from passive human videos. *arXiv preprint arXiv:2302.02011*, 2023.
- [62] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022.
- [63] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *CVPR*, 2023.
- [64] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13226–13233. IEEE, 2025.
- [65] Liangzhi Shi, Yulin Liu, Lingqi Zeng, Bo Ai, Zhengdong Hong, and Hao Su. Learning adaptive dexterous grasping from single demonstrations. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9456–9463. IEEE, 2025. **3**
- [66] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pages 306–324. Springer, 2024. **3**
- [67] Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. Robotap: Tracking arbitrary points for few-shot visual imitation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5397–5403. IEEE, 2024.
- [68] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.
- [69] Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *International Conference on Learning Representations (ICLR)*, 2024. **3**
- [70] Ivan Kapelyukh, Vitalis Vosylius, and Edward Johns. Dall-e-bot: Introducing web-scale diffusion models to robotics. *IEEE Robotics and Automation Letters*, 8(7): 3956–3963, 2023. **3**
- [71] Zhao Mandi, Homanga Bharadhwaj, Vincent Moens, Shuran Song, Aravind Rajeswaran, and Vikash Kumar. Cacti: A framework for scalable multi-task multi-scene visual imitation learning. *arXiv preprint arXiv:2212.05711*, 2022.
- [72] Zoey Chen, Sho Miami, Abhishek Gupta, and Vikash Kumar. Genau: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.
- [73] Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023.
- [74] Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Brianna Zitkovich, Fei Xia, Chelsea Finn, et al. Openworld object manipulation using pre-trained vision-language models. *arXiv preprint arXiv:2303.00905*, 2023.
- [75] Danny Driess, Fei Xia, Mehdi S.M. Sajjadi, et al. Palme: An embodied multimodal language model. *International Conference on Machine Learning (ICML)*, 2023.
- [76] 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. *International Conference on Machine Learning (ICML)*, 2023. **3**
- [77] OX-Embodiment Collaboration, A Padalkar, A Pooley, A Jain, A Bewley, A Herzog, A Irpan, A Khazatsky, A Rai, A Singh, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 1(2), 2023. **3**
- [78] Jonathan Yang, Chelsea Finn, and Dorsa Sadigh. Data analogies enable efficient cross-embodiment transfer. *arXiv preprint arXiv:2603.06450*, 2026. **3**
- [79] Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *Conference on Robot Learning*, 2024.
- [80] Jonathan Yang, Catherine Glossop, Arjun Bhorkar, Dhruv Shah, Quan Vuong, Chelsea Finn, Dorsa Sadigh, and Sergey Levine. Pushing the limits of cross-embodiment learning for manipulation and navigation. *arXiv preprint arXiv:2402.19432*, 2024. **10**
- [81] Lihan Zha, Asher J Hancock, Mingtong Zhang, Tenny Yin, Yixuan Huang, Dhruv Shah, Allen Z Ren, and Anirudha Majumdar. Lap: Language-action pre-training enables zero-shot cross-embodiment transfer. *arXiv preprint arXiv:2602.10556*, 2026.

- [82] Shresth Grover, Akshay Gopalkrishnan, Bo Ai, Henrik I Christensen, Hao Su, and Xuanlin Li. Enhancing generalization in vision-language-action models by preserving pretrained representations. *arXiv preprint arXiv:2509.11417*, 2025.
- [83] Bo Ai, Liu Dai, Nico Bohlinger, Dichen Li, Tongzhou Mu, Zhanxin Wu, K Fay, Henrik I Christensen, Jan Peters, and Hao Su. Towards embodiment scaling laws in robot locomotion. *Conference on Robot Learning (CoRL)*, 2025.
- [84] Zihao He, Bo Ai, Tongzhou Mu, Yulin Liu, Weikang Wan, Jiawei Fu, Yilun Du, Henrik I Christensen, and Hao Su. Scaling cross-embodiment world models for dexterous manipulation. *arXiv preprint arXiv:2511.01177*, 2025. 3
- [85] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024. 3
- [86] Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy. In *Conference on Robot Learning*, pages 1992–2005. PMLR, 2021. 3
- [87] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. Zero-shot visual imitation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2050–2053, 2018. 3
- [88] Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jacob Varley, Alex Irpan, Benjamin Eysenbach, Ryan C Julian, Chelsea Finn, et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. In *International Conference on Machine Learning*, pages 1518–1528. PMLR, 2021.
- [89] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X Lee, Maria Bauza, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving foundation agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023.
- [90] Vivek Myers, Andre He, Kuan Fang, Homer Walke, Philippe Hansen-Estruch, Ching-An Cheng, Mihai Jalobeanu, Andrey Kolobov, Anca Dragan, and Sergey Levine. Goal representations for instruction following: A semi-supervised language interface to control. *Conference on Robot Learning (CoRL)*, 2023. 3
- [91] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. *Advances in neural information processing systems*, 31, 2018. 3
- [92] Suraj Nair and Chelsea Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. In *International Conference on Learning Representations*, 2020.
- [93] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023. 5, 22
- [94] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. In *The Twelfth International Conference on Learning Representations*, 2024.
- [95] Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. Uniskill: Imitating human videos via cross-embodiment skill representations. In *Conference on Robot Learning*, pages 4269–4294. PMLR, 2025.
- [96] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. In *Conference on Robot Learning*, pages 3943–3960. PMLR, 2025.
- [97] Zhuoyang Zhang, Shang Yang, Qinghao Hu, Luke J Huang, James Hou, Yufei Sun, Yao Lu, and Song Han. Foreact: Steering your vla with efficient visual foresight planning. *arXiv preprint arXiv:2602.12322*, 2026.
- [98] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. *International Conference on Learning Representations (ICLR)*, 2024. 3
- [99] Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025. 3
- [100] Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.
- [101] Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3
- [102] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 3, 5
- [103] Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, 2025. 3
- [104] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny

- Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *Robotics: Science and Systems*, 2025. 3
- [105] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. 4, 5, 22
- [106] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesh Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Pöder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. 3
- [107] Kevin Black, Manuel Y Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *arXiv preprint arXiv:2506.07339*, 2025. 7, 22
- [108] Kevin Black, Allen Z Ren, Michael Equi, and Sergey Levine. Training-time action conditioning for efficient real-time chunking. *arXiv preprint arXiv:2512.05964*, 2025. 7, 22
- [109] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 7
- [110] Joey Hejna, Suvir Mirchandani, Ashwin Balakrishna, Annie Xie, Ayzaan Wahid, Jonathan Tompson, Pannag Sanketi, Dhruv Shah, Coline Devin, and Dorsa Sadigh. Robot data curation with mutual information estimators. *arXiv preprint arXiv:2502.08623*, 2025. 14
- [111] Yunfei Li, Xiao Ma, Jiafeng Xu, Yu Cui, Zhongren Cui, Zhigang Han, Liqun Huang, Tao Kong, Yuxiao Liu, Hao Niu, et al. Gr-rl: Going dexterous and precise for long-horizon robotic manipulation. *arXiv preprint arXiv:2512.01801*, 2025. 14
- [112] Jintao Zhang, Jia Wei, Haofeng Huang, Penge Zhang, Jun Zhu, and Jianfei Chen. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. *arXiv preprint arXiv:2410.02367*, 2024. 23

APPENDIX

A. Contributions

Data collection and operations. Ashwin Balakrishna, George Bokinsky, Thomas Charbonnier, Grace Connors, Michael Equi, Chelsea Finn, Lachlan Groom, Hunter Hancock, Karol Hausman, Connor Jacobsen, Rowan Jen, Marinda Lamb, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Tyler Patterson, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Will Stoeckle, Anna Walling, Jason Wang, Samuel Whitmore, Blake Williams.

Annotation and supplemental data. Ashwin Balakrishna, Karan Dhabalia, Danny Driess, Chelsea Finn, Haroun Habeeb, Rowan Jen, Chandra Kuchi, Karl Pertsch, Lucy Xiaoyang Shi, Will Stoeckle, Quan Vuong.

Policy training and research. Bo Ai, Ashwin Balakrishna, Kevin Black, Danny Driess, Michael Equi, Yunhao Fang, Chelsea Finn, Catherine Glossop, Haroun Habeeb, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Liyiming Ke, Sergey Levine, Xinyu Li, Yao Lu, Suraj Nair, Karl Pertsch, Allen Z. Ren, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Jiaming Tang, Marcel Torne, Kyle Vedder, Quan Vuong, XuDong Wang, Charles Xu, Lili Yu, Wuming Zhang, Zhuoyang Zhang.

Policy infrastructure. Kevin Black, Karan Dhabalia, Danny Driess, Mairbek Khadikov, Chandra Kuchi, Adrian Li-Bell, Vladislav Lialin, Wallace Lim, Yao Lu, Allen Z. Ren, Lucy Xiaoyang Shi, Kyle Stachowicz, Jiaming Tang, Quan Vuong, Haohuan Wang, Ury Zhilinsky.

Robot hardware. Ali Amin, Raichelle Aniceto, Greg Balke, Vedant Choudhary, Foster Collins, Grace Connors, Maitrayee Dhaka, Adnan Esmail, Thomas Godden, Ivan Goryachev, Tim Jones, Gregg Kammerer, Ben Katz, Devin LeBlanc, Brendon LeCount, Zhonglin Liang, Enyu Luo, Liam Murphy, Gavin Schelske, Shalom Tekeste, Chris Whalen, Sukwon Yoo.

Robot infrastructure. Greg Balke, Kevin Black, Shihao Cao, Ken Conley, James Darpinian, Jared DiCarlo, Hunter Hancock, Karol Hausman, Szymon Jakubczak, Jimmy Tanner.

Writing and illustration. Bo Ai, Ashwin Balakrishna, Kevin Black, Chelsea Finn, Sergey Levine, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Kyle Stachowicz.

B. Attention pattern

We describe the attention pattern used in training $\pi_{0.7}$ and the lightweight world model (for subgoal image generation), as well as running inference with them, in Fig. 19.

C. Training of the world model

Our world model is initialized from BAGEL [105], and largely uses the same training recipe. We use a subset of our robot data and egocentric human video data with high-quality segmented language labels, as we found that label quality (especially temporal segmentation quality) has a large impact on subgoal quality. We additionally mix in several open-source image editing datasets and open-source video datasets to better preserve the semantic knowledge of the model. Each training example consists of a subtask instruction, $\hat{\ell}$, 3 camera inputs,

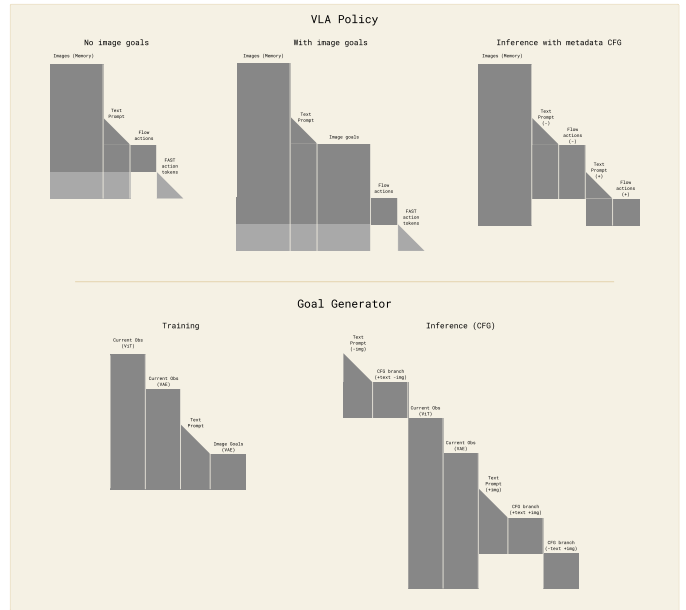


Fig. 19: The $\pi_{0.7}$ model and its world model (for generating subgoal images) use several different nontrivial attention patterns during training and inference. From top left: in absence of image goals we use the same attention patterns as in $\pi_{0.5}$, with global bidirectional attention between embeddings for all memory-aware image views. Note that the FAST tokens (only available at training time) and the flow actions do not attend to each other. When image goals are present, we include them as an additional block-causal bidirectional block, after the text prompt. When we do classifier-free guidance (at inference time), we pack both the positive and negative example into the same sequence for efficient inference by constructing an “attention tree” with two branches (positive and negative), which do not attend to one another. Following BAGEL, at training time the world model receives three copies of the image, each of which is block-bidirectional within the multiview group: current observation, encoded with ViT; current observation encoded with VAE; noisy image goal encoded with VAE. The world model uses a similar inference-time CFG trick, albeit with a more complex mask because it has three CFG groups rather than two.

\mathbf{o}_t , and 3 target images, $\mathbf{o}_{t_{\text{end}}}$, where t_{end} is the last timestep of the segment spanning t . Following BAGEL, the camera inputs are processed using both a ViT (for semantic understanding) and VAE (for fine-grained image details). The ViT tokens are further processed by a 7B LLM backbone, and the VAE tokens are processed by a 7B generation backbone. ViT inputs are resized to a resolution of 448x336, and VAE inputs (including target images) are resized to a resolution of 512x384. The discrepancy is due to the differing patch size of the ViT and VAE (14 and 16, respectively). At test time, we set $\Delta = 4$ seconds as the time interval for regenerating subgoals to match SuSIE [93].

D. Inference speed and optimization

The $\pi_{0.7}$ model and the high-level policy, which are both based on Gemma3 4B, use a single NVIDIA H100 GPU for inference. Various optimizations implemented after RTC [107] have brought the inference time of the minimal variant of $\pi_{0.7}$ down to 38ms with 3 camera inputs, 5 denoising steps, and training-time RTC [108] (which, unlike test-time RTC, incurs no additional inference-time overhead). Enabling the MEM vision encoder [37] and adding subgoal images to the

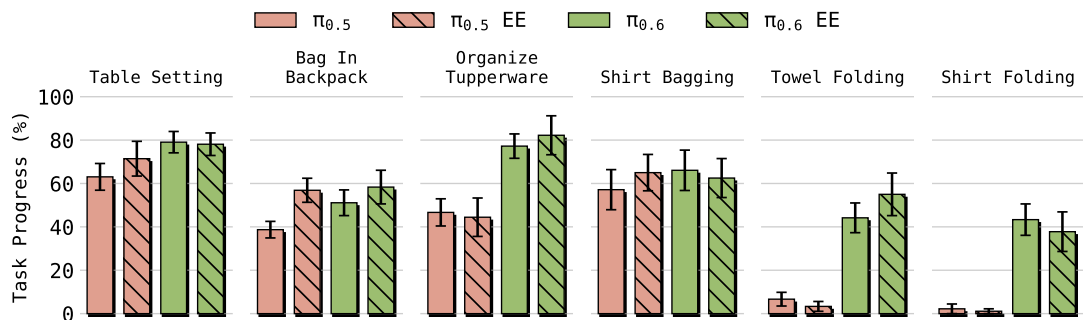


Fig. 20: **Joint vs. end-effector control for prior models on cross-embodiment tasks.** We compare joint-space and end-effector (EE) control for baseline policies across a range of tasks, observing no substantial difference in performance between the two control modes.

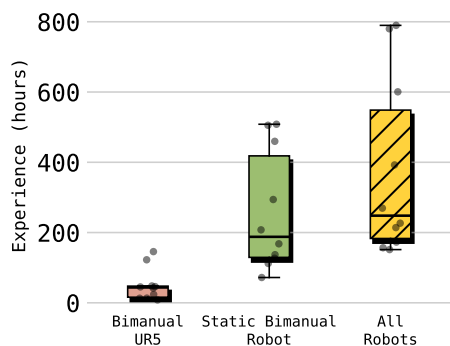


Fig. 21: **Operator experience in the human subject study.** Box plots show teleoperation experience (in hours) of the ten recruited operators across three categories: UR5e (target robot), the static bimanual robot (source robot), and all robots combined. The selected operators rank within the top 2% of our operator fleet in terms of teleoperation experience.

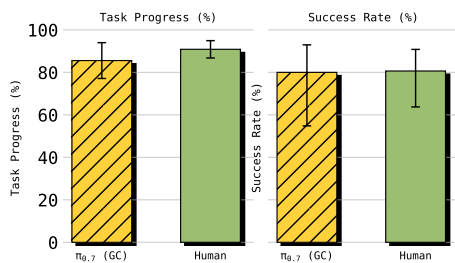


Fig. 22: **Comparison of $\pi_{0.7}$ (GC) and human.** We find that $\pi_{0.7}$ (GC) achieves competitive performance compared to the human operators, in the shirt folding task with the UR5e bimanual platform.

context both incur additional overhead, bringing the inference time to 127ms in the worst case.

Generating subgoal images at a reasonable latency is challenging, due to the computational expense of iterative denoising with a 14B model and total sequence length of nearly 10,000 tokens. In addition to the aforementioned optimizations, we also use 4-way tensor parallelism on 4xH100 GPUs, quantize all large matrix multiplications to 8-bit precision, and use a modified version of SageAttention [112] for the backbone attention operations. This allows us to generate subgoal images with 25 denoising steps (each including both text and image CFG) in 1.25 seconds. At inference time, we

execute using a naive asynchronous strategy, meaning that $\pi_{0.7}$ continues executing while the world model generates the next subgoal.

E. Comparison of action spaces in cross-embodiment transfer

Figure 20 compares joint-space and end-effector (EE) control on cross-embodiment tasks. Across tasks, EE control does not show clear advantage. Therefore, we focus on joint-space control in the main cross-embodiment experiments (Sec. IX-C) for clarity.

F. Human subject study for cross-embodiment shirt folding

The human subject study serves two purposes. First, it provides the strongest possible baseline for evaluating $\pi_{0.7}$ by measuring how well expert operators can teleoperate the UR5e bimanual robot to perform the same task. Second, the result potentially motivates for cross-embodiment transfer: UR5e, an industrial manipulator with high joint inertia, is difficult to teleoperate precisely for dexterous tasks like shirt folding, making demonstration collection on this platform difficult. A model that transfers to new embodiment opens up opportunities to collect autonomous data without human teleoperation.

Participant selection. We recruited ten operators with the top 2 percentile of experience from the entire group of operators. They all have extensive prior experience teleoperating the source static bimanual robot, with an average experience of ~ 375 hours across all robot platforms (Fig. 21). Crucially, none had prior experience performing the shirt folding task on the UR5e, mirroring the “zero-shot” setting of our learned policy.

Protocol. Each operator performed three trials, yielding 30 total trials. To match the zero-shot transfer setting of $\pi_{0.7}$, operators received no practice or warm-up period before their first attempt. The initial shirt configuration (flattened on the table), the time limit, and the evaluation criteria were all identical to those used in the $\pi_{0.7}$ policy evaluation. Operators were instructed to maximize task success to align with the evaluation metric. We report task progress and success rate using the same metrics as in our robot experiments, ensuring a direct and fair comparison.

Results. Fig. 22 compares $\pi_{0.7}$ (GC) with human teleoperators under the same zero-shot setting. Human operators achieve an average task progress of 90.9% and a success rate of 80.6%. $\pi_{0.7}$ achieves 85.6% task progress and an 80% success rate, demonstrating performance comparable to these expert operators despite not trained with any folding data from the UR5e platforms. These results provide strong evidence of zero-shot cross-embodiment transfer in $\pi_{0.7}$.

G. Detailed task descriptions and scoring rubric

Laundry (T-shirts and Shorts). Fold a single pair of shorts or a T-shirt taken from a laundry basket into a compact, neatly aligned fold and place/stack it appropriately. Scoring: success if successfully folding the item and properly stacking it on top of other laundry.

Laundry (Diverse — Hardest Item). Fold a single buttonup shirt taken from a laundry basket into a compact, neatly aligned fold and place/stack it appropriately. Scoring: success if successfully folding the item and properly stacking it on top of other laundry.

Make Espresso. Execute an espresso workflow to prepare and extract a doppio (double espresso) with correct sequencing and placements. Scoring: success if completing the required steps in order: grounds dispensed and tamped, portafilter locked into the correct grouphead, extraction performed, cup placed on saucer and moved right.

Box Building. Assemble a flat box into a 3D box. Scoring: success if the box is folded correctly without major damage.

Make Peanut Butter sandwich. Make a peanut-butter sandwich, cut it diagonally all the way through, and present it on the plate; close the peanut butter jar; push the plate away. Maximum score: 9. Points awarded for: removing the jar lid; progressively spreading peanut butter coverage; placing the plain slice on top; cutting diagonally fully through; placing the knife back on the plate; pushing the plate away; and overall neatness.

Turn a T-shirt Inside Out. Retrieve an inside-out t-shirt, turn it fully right-side-out (torso + both sleeves), fold it into a compact form. Maximum score: 7. +1 retrieve shirt; +1 torso right-side-out; +1 left sleeve right-side-out; +1 right sleeve right-side-out; +1 compact fold; +1 placed on pile; +1 positioned in upper-right corner.

Drive Through Door. Open a self-closing closet door, drive the robot fully inside, and finish with the door closed and the robot inside. Maximum score: 3. +1 for opening the door; +1 for driving far enough inside for the door to close; +1 for smooth entry without bumping into the doorway.

Cut Zucchini. Slice a zucchini into thin slices using a knife attached to a lanyard while stabilizing the zucchini with the opposite gripper. Maximum score: 3. +1 for successfully picking up the knife correctly; +1 for fully cutting the zucchini into thin, even slices; +1 for safely returning the knife to the right side of the cutting board.

Peel Fruits and Vegetables. Peel a fruit/vegetable completely by holding the item against cutting board, and peeling it with the other gripper. Maximum score: 9. +1 for picking up the

peeler; +1 for peeling up to 25%; +1 for peeling 25–50%; +1 for peeling 50–75%; +1 for peeling >75%; +1 for putting the fruit/veg into the bowl; +1 for scraping 25–50% of food scraps into the trash can; +1 for scraping 50–75% of food scraps into the trash can; +1 for scraping >75% of food scraps into the trash can.

Take Out Trash. Remove an old trash bag, place it away from the bin, line the trash can with a new bag, and return the bin to its home location (and close the cabinet / replace lid as appropriate). Maximum score: 12. Taking out trash can (3 points): +2 if the robot correctly accesses the trash can (opens the under-sink cabinet door); +1 if the robot relocates the trash can from under the sink to an accessible position on the kitchen floor. Remove from bin (3 points): +1 if the robot pulls the trash bag away from the corners of the bin; +1 if the robot gathers and securely lifts the trash bag out of the bin; +1 if the robot removes the trash bag from the bin and places it on the floor near the trash can. Replace bag (3 points): +1 if the robot picks up the replacement trash bag; +1 if the robot opens the replacement bag; +1 if the robot places the replacement trash bag fully inside the trash can and stretches the edges securely around the rim. Return trash can back (3 points): +2 if the robot lifts and returns the trash can to the under-sink cabinet; +1 if the robot closes the cabinet door fully at the end of the episode.

Swap 3 Mugs. Sequentially place three mugs onto a coffee machine drip tray (one at a time), ensuring each mug occupies the tray once, with mugs returned to the table between placements as required. Maximum score: 4. +1 for removing mug 1 from the coffee maker and placing it on the table; +1 for placing mug 2 from the table onto the coffee maker; +1 for removing that same mug 2 from the coffee maker and placing it on the table; +1 for placing the remaining mug 3 from the table onto the coffee maker.

Find Object. Retrieve an object hidden in a drawer and restore a clean final scene (object on table, drawers closed). Maximum score: 4. +1 for fully opening the target drawer with the item on the first try; +1 for picking up the sponge stick; +1 for placing the sponge stick on the table (anywhere); +1 for closing the opened target drawer.

Scoop Beans. Scoop exactly two scoops of coffee beans into a grinder using a measuring cup, then close the grinder lid. Maximum score: 5. +1 for opening the coffee grinder; +1 for grasping the scoop; +1 for successfully collecting and pouring the first full scoop of beans into the coffee grinder; +1 for successfully collecting and pouring the second full scoop of beans into the coffee grinder; +1 for closing the coffee grinder lid.

Window Cleaning. Spray the phone booth window with windex, rip a paper towel, wipe the glass fully dry, and discard the towel. Maximum score: 5. +1 for spraying the booth; +1 for getting the paper towel for cleaning; +1 for cleaning overall the whole door and leaving it dry; +1 for throwing the paper towel in the trash can; +1 for not leaving any drops left on the door.

Reverse Bussing. Sort 12 objects with reversed mapping:

place trash into the bussing bin and dishes/utensils into the trash can. Maximum score: 12. +1 for successfully putting a plate, cup, bowl, or utensil in the trash (7 total); +1 for successfully putting a plastic bottle, foil, plastic lid, take out container, or chip bag in the bussing bin (5 total).

Reverse Fridge to Microwave. Move a plate of (real) frozen microwavable food from the microwave into the refrigerator (reverse sequence version), and complete the episode with the plate stored in the fridge. Maximum score: 6. +1 for microwave door opening; +1 for microwave door closing; +1 for plate removed from microwave; +1 for plate placed in refrigerator; +1 for refrigerator door opening; +1 for refrigerator door closing.

Table Setting. Set a table by placing a placemat, cup, plates, napkin, and utensils from a bin into a reasonable table setting. Maximum score: 7. +1 per item successfully placed; critically incorrect placements can be scored as -1.

Bag in Backpack. Place a small pouch/bag into a backpack (no need to zip closed). Maximum score: 3. +1 pick up pouch; +1 grasp backpack; +1 place pouch inside backpack.

Organize Tupperware. Nest three tupperware containers (descending sizes) and stack their corresponding lids. Maximum score: 6. +1 for each container correctly nested; +1 for each lid correctly stacked/aligned.

Shirt Bagging. Place two shirts fully inside a brown grocery bag (per-scene), optionally using language commands with timing-sensitive instruction handoff. Maximum score: 4. +1 for correct pick for each shirt; +1 for each shirt fully placed inside the bag.

Shirt Folding. Fold a single t-shirt from a flattened start state into a folded end state. Maximum score: 6. +1 for completing the first fold (grasp cloth with both arms; folded edge aligned with intended fold within 5 inches); +1 for completing the second fold (same criterion); +1 for completing the final fold (completed with right arm; aligned within 5 inches); plus a fold quality score from 0–3 based on the final folded state chart. Full score of 6 is considered success.

Press French Press Plunger. Press the french press plunger fully down to the bottom. Maximum score: 1. +1 if the robot manages to press the plunger down to the bottom (through coffee).

Scoop Rice into Rice Cooker. Pick up a rice scooper from a rice container, scoop rice, and pour it into an open rice cooker. Maximum score: 3. +1 pick up scooper; +1 scoop rice; +1 pour into cooker.

Loading an Air Fryer. Open an air fryer, place a sweet potato into the basket, and close the air fryer. Maximum score: 4. +1 open air fryer; +1 pick up sweet potato; +1 place into air fryer; +1 close air fryer.

Unloading an Air Fryer. Pull out an air fryer basket and dump 8 fake fries onto a plate. Maximum score: 2. +1 if the robot successfully picks up all food items from the air fryer; +1 if the robot places all food items onto the plate.

Toast a Bagel. Place a sliced half-bagel into a toaster oven, initiate toasting by turning the knob, then retrieve and serve it on a plate taken from an overhead cabinet. Maximum score: 7

(open oven; insert bagel; close oven; turn knob; retrieve plate; retrieve toast; place on plate).