VIMA: General Robot Manipulation with Multimodal Prompts

Yunfan Jiang [1] **Agrim Gupta** [1 †] **Zichen Zhang** [2 †] **Guanzhi Wang** [3 4 †] **Yongqiang Dou** [5] **Yanjun Chen** [1]
Li Fei-Fei [1] **Anima Anandkumar** [3 4] **Yuke Zhu** [3 6 ‡] **Linxi Fan** [3 ‡]

Abstract

Prompt-based learning has emerged as a successful paradigm in natural language processing, where a single general-purpose language model can be instructed to perform any task specified by input prompts. Yet task specification in robotics comes in various forms, such as imitating one-shot demonstrations, following language instructions, and reaching visual goals. They are often considered different tasks and tackled by specialized models. We show that a wide spectrum of robot manipulation tasks can be expressed with *multimodal prompts*, interleaving textual and visual tokens. Accordingly, we develop a new simulation benchmark that consists of thousands of procedurally-generated tabletop tasks with multimodal prompts, 600K+ expert trajectories for imitation learning, and a four-level evaluation protocol for systematic generalization. We design a transformer-based robot agent, VIMA, that processes these prompts and outputs motor actions autoregressively. VIMA features a recipe that achieves strong model scalability and data efficiency. It outperforms alternative designs in the hardest zero-shot generalization setting by up to $2.9\times$ task success rate given the same training data. With $10\times$ less training data, VIMA still performs $2.7\times$ better than the best competing variant. Code and video demos are available at `vimalabs.github.io`.

1. Introduction

Transformer models (Vaswani et al., 2017) have given rise to remarkable multi-task consolidation across many AI domains. For example, users can describe a task using natural language prompt to GPT-3 (Brown et al., 2020), allowing the same model to perform question answering, machine translation, text summarization, etc. Prompt-based learning provides an accessible and flexible interface to communicate a natural language understanding task to a general-purpose model.

We envision that a generalist robot should have a similarly intuitive and expressive interface for task specification. What does such an interface for robot learning look like? As a motivating example, consider a personal robot tasked with household activities. We can ask the robot to bring us a cup of water by a simple natural language instruction. If we require more specificity, we can instead instruct the robot to "bring me `<image of the cup>`". For tasks requiring new skills, the robot should be able to adapt, preferably from a few video demonstrations (Duan et al., 2017). Tasks that need interaction with unfamiliar objects can be easily explained via a few image examples for *novel concept grounding* (Hermann et al., 2017). Finally, to ensure safe deployment, we can further specify visual constraints like "do not enter `<image>` room".

To enable a single agent with all these capabilities, we make three key contributions in this work: 1) a novel **multimodal prompting formulation** that converts a wide spectrum of robot manipulation tasks into one sequence modeling problem; 2) a **large-scale benchmark** with diverse tasks to systematically evaluate an agent's scalability and generalization; and 3) a **multimodal-prompted robot agent** capable of multi-task and zero-shot generalization.

We start with the observation that many robot manipulation tasks can be formulated by **multimodal prompts that interleave language and images or video frames** (Fig. 1). For example, Rearrangement (Batra et al., 2020), a type of *Visual Goal*, can be formulated as "Please rearrange objects to match this $\{\texttt{scene\_image}\}$"; *Few-shot Imitation* can embed video snippet in the prompt "Follow this motion trajectory for the wooden cube: $\{\texttt{frame}_1\}, \{\texttt{frame}_2\}, \{\texttt{frame}_3\}, \{\texttt{frame}_4\}$". Multimodal prompts not only have more expressive power than individual modalities but also enable a **uniform sequence IO interface** for training generalist robots. Previously, different robot manipulation tasks required distinct policy architectures, objective functions, data pipelines, and training

[1]Stanford University; [2]Macalester College, now at Allen Institute for AI; [3]NVIDIA; [4]Caltech; [5]Tsinghua; [6]UT Austin. Work done during the first author's internship at NVIDIA. †: Equal contribution. ‡: Equal advising.

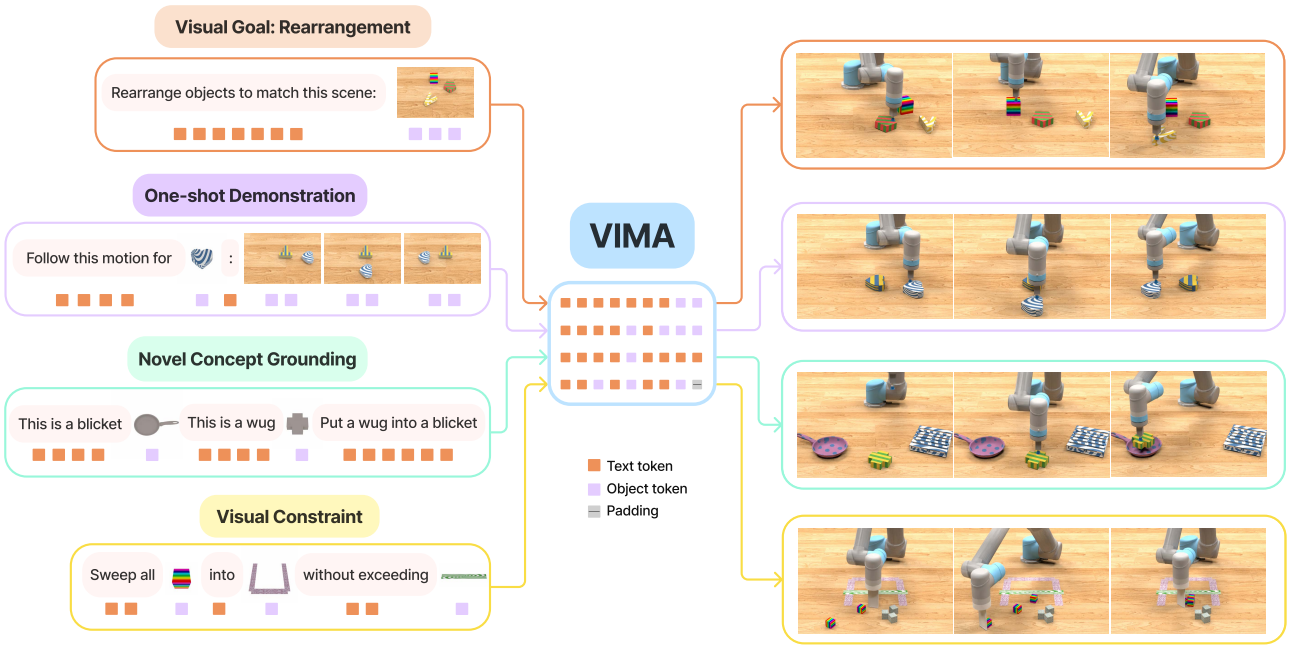

Figure 1: **Multimodal prompts for task specification.** We observe that many robot manipulation tasks can be expressed as *multimodal prompts* that interleave language and image/video frames. We introduce VIMA, an embodied agent capable of processing mulitimodal prompts (left) and controlling a robot arm to solve the task (right).

procedures (Aceituno et al., 2021; Stengel-Eskin et al., 2022; Lynch & Sermanet, 2021), leading to siloed robot systems that cannot be easily combined for a rich set of use cases. Instead, our multimodal prompt interface allows us to harness the latest advances in large transformer models (Lin et al., 2021; Tay et al., 2020; Khan et al., 2021) for developing scalable multi-task robot learners.

To systematically evaluate agents with multimodal prompts, we develop a new benchmark, named VIMA-BENCH, built on the Ravens simulator (Zeng et al., 2020; Shridhar et al., 2021). We provide 17 representative tasks with multimodal prompt templates. Each task can be procedurally instantiated into thousands of instances by various combinations of textures and tabletop objects. VIMA-BENCH establishes a four-level protocol to evaluate progressively stronger generalization capabilities, from randomized object placement to novel tasks (Fig. 2).

To this end, we introduce the **Vi**suo**M**otor **A**ttention agent (VIMA) to learn robot manipulation from multimodal prompts. The model architecture follows the encoder-decoder transformer design proven to be effective and scalable in NLP (Raffel et al., 2020). VIMA encodes an input sequence of interleaving textual and visual prompt tokens with a pre-trained language model (Tsimpoukelli et al., 2021) and decodes robot control actions autoregressively for each environment interaction step. The transformer decoder is conditioned on the prompt via cross-attention layers that alternate with the usual causal self-attention. Instead of operating on raw images, VIMA adopts an object-centric approach. We parse all images in the prompt or observation into objects by off-the-shelf then domain fine-tuned detectors (He et al., 2017) and flatten them into sequences of object tokens. To demonstrate the scalability of VIMA, we train a spectrum of 7 models ranging from 2M to 200M parameters. Our approach outperforms other design alternatives, such as image patch tokens (Reed et al., 2022), image Perceiver (Jaegle et al., 2021b; Alayrac et al., 2022), and decoder-only conditioning (Radford et al., 2018). VIMA obtains consistent performance gains across all four levels of zero-shot generalization and all model capacities, in some cases by a large margin (up to 2.9× task success rate given the same amount of training data, and 2.7× better even with 10× less data). We open-source the simulation environment, training dataset, algorithm code, and pre-trained model checkpoints to ensure reproducibility and facilitate future work from the community. These materials along with video demos are available at `vimalabs.github.io`.

2. Multimodal Prompts for Task Specification

A central and open problem in robot learning is task specification (Agrawal, 2022). In prior literature (Stepputtis et al., 2020; Dasari & Gupta, 2020; Brunke et al., 2021b), different tasks often require diverse and incompatible interfaces, resulting in siloed robot systems that do not generalize well

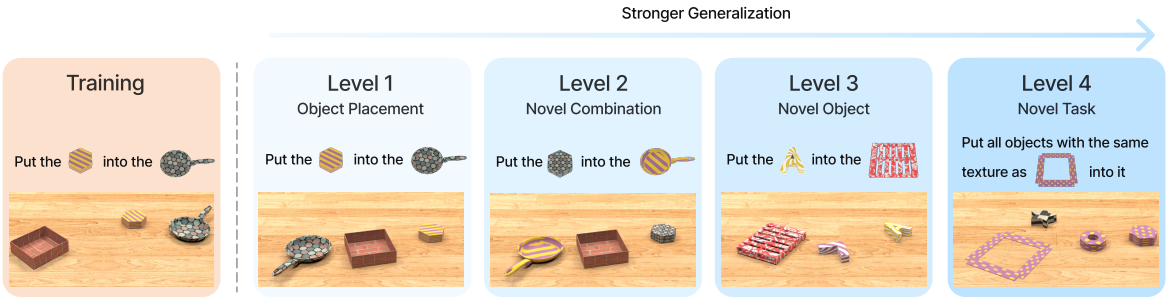

Figure 2: **Evaluation Protocol in VIMA-BENCH.** We design 4 levels of evaluation settings to systematically measure the zero-shot generalization capability of an agent. Each level deviates more from the training distribution, and thus is strictly more challenging than the previous level.

across tasks. Our key insight is that various task specification paradigms (such as goal conditioning, video demonstration, natural language instruction) can all be instantiated as multimodal prompts (Fig. 1). Concretely, a multimodal prompt $\mathcal{P}$ of length l is defined as an ordered sequence of arbitrarily interleaved texts and images $\mathcal{P} := [x_1, x_2, \ldots, x_l]$, where each element $x_i \in \{\text{text}, \text{image}\}$.

Task Suite. The flexibility afforded by multimodal prompts allows us to specify and build models for a variety of task specification formats. Here we consider the following six categories.

1. **Simple object manipulation.** Simple tasks like "put `<object>` into `<container>`", where each image in the prompt corresponds to a single object;
2. **Visual goal reaching.** Manipulating objects to reach a goal configuration, *e.g.*, *Rearrangement* (Batra et al., 2020);
3. **Novel concept grounding.** The prompt contains unfamiliar words like "dax" and "blicket", which are explained by in-prompt images and then immediately used in an instruction. This tests the agent's ability to rapidly internalize new concepts;
4. **One-shot video imitation.** Watching a video demonstration and learning to reproduce the same motion trajectory for a particular object;
5. **Visual constraint satisfaction.** The robot must manipulate the objects carefully and avoid violating the (safety) constraints;
6. **Visual reasoning.** Tasks that require reasoning skills, such as appearance matching "move all objects with same textures as `<object>` into `<container>`", and visual memory "put `<object>` in `<container>` and then restore to their original position".

Note that these six categories are not mutually exclusive. For example, a task may introduce a previously unseen verb (*Novel Concept*) by showing a video demonstration, or combine goal reaching with visual reasoning. More details about the task suite are discussed in Appendix, Sec. B.

3. VIMA-BENCH: Benchmark for Multimodal Robot Learning

Simulation Environment. Existing benchmarks are generally geared towards a particular task specification. To our knowledge, there is no benchmark that provides a rich suite of multimodal tasks and a comprehensive testbed for targeted probing of agent capabilities. To this end, we introduce a new benchmark suite for multimodal robot learning called VIMA-BENCH. We build our benchmark by extending the Ravens robot simulator (Zeng et al., 2020). VIMA-BENCH supports extensible collections of objects and textures to compose multimodal prompts and to procedurally generate a large number of tasks. Specifically, we provide 17 tasks with multimodal prompt templates, which can be instantiated into thousands of task instances. Each task belongs to one or more of the 6 task categories mentioned above. VIMA-BENCH can generate large quantities of imitation learning data via scripted oracle agents. More details are elaborated in Appendix, Sec. A.

Observation and Actions. The observation space of our simulator includes RGB images rendered from both frontal view and top-down view. Ground-truth object segmentation and bounding boxes are also provided for training object-centric models (Sec. 4). We inherit the high-level action space from Zeng et al. (2020), which consists of primitive motor skills like "pick and place" and "wipe". These are parameterized by poses of the end effector. Our simulator also features scripted oracle programs that can generate expert demonstrations by using privileged simulator state information, such as the precise location of all objects, and the ground-truth interpretation of the multimodal instruction.

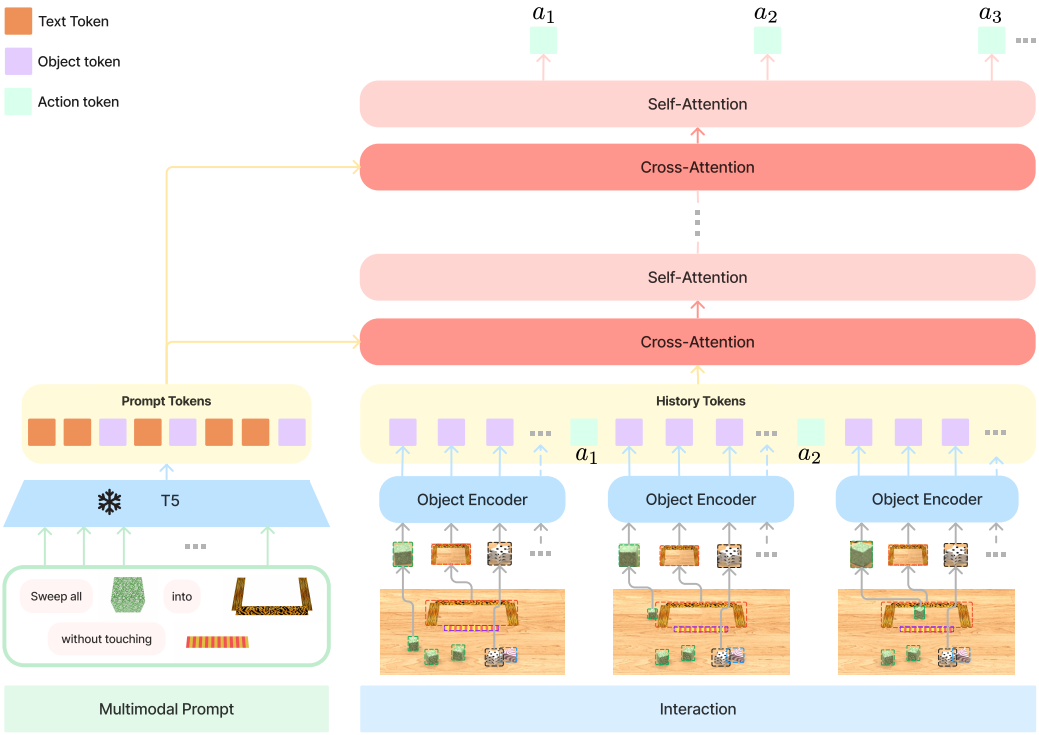

Figure 3: **VIMA Architecture.** We encode the multimodal prompts with a pre-trained T5 model, and condition the robot controller on the prompt through cross-attention layers. The controller is a causal transformer decoder consisting of alternating self and cross attention layers that predicts motor commands conditioned on prompts and interaction history.

Training Dataset. We leverage oracles to generate a large offline dataset of expert trajectories for imitation learning. Our dataset includes 50K trajectories per task, and 650K successful trajectories in total. We hold out a subset of objects and textures for evaluation and designate 4 out of 17 tasks as a testbed for zero-shot generalization.

Evaluating Zero-Shot Generalization. Each task in VIMA-BENCH has a binary success criterion and does not provide partial reward. During test time, we execute agent policies in the simulator for multiple episodes to compute a percentage success rate. The average success rate over all evaluated tasks will be the final reported metric.

We design a four-level evaluation protocol (Fig. 2) to systematically probe the generalization capabilities of learned agents. Each level deviates more from the training distribution, and is thus strictly harder than the previous one.

1. **Placement generalization.** All prompts are seen verbatim during training, but only the placement of objects on the tabletop is randomized at testing;

2. **Combinatorial generalization.** All textures and objects are seen during training, but new combinations of them appear in testing;

3. **Novel object generalization.** Test prompts and the simulated workspace include novel textures and objects;

4. **Novel task generalization.** New tasks with novel prompt templates at test time.

4. VIMA: Visuomotor Attention Agent

Our goal is to build a robot agent capable of performing any task specified by multimodal prompts. There is no prior method that works out of the box with multimodal prompts. To learn an effective multi-task robot policy, we propose VIMA, a robot agent with a multi-task encoder-decoder architecture and object-centric design (Fig. 3). Concretely, we learn a robot policy $\pi(a_t|\mathcal{P}, \mathcal{H})$, where $\mathcal{H} := [o_1, a_1, o_2, a_2, \ldots, o_t]$ denotes the past interaction history, and $o_t \in \mathcal{O}, a_t \in \mathcal{A}$ are observations and actions at each interaction steps. We encode multimodal prompts via a *frozen* pre-trained language model and decode robot waypoint commands conditioned on the encoded prompts via cross-attention layers. Unlike prior work (Florence et al., 2019; Sieb et al., 2019; Zhu et al., 2022), VIMA adopts an object-centric representation that computes tokens from bounding box coordinates and cropped RGB patches.

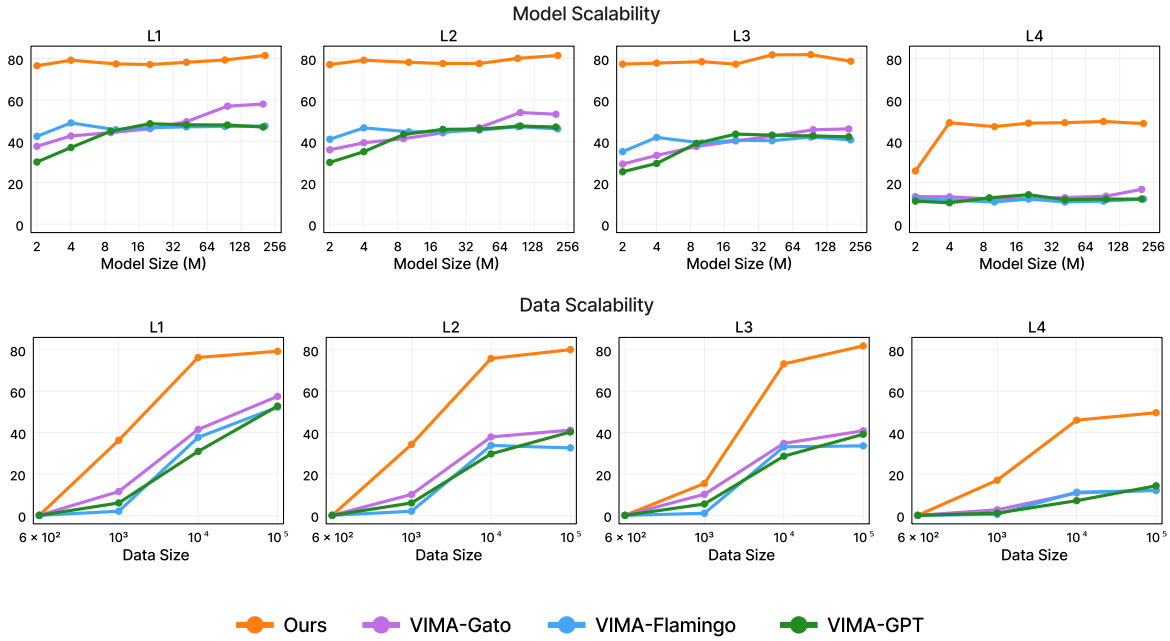

Figure 4: **Scaling model and data.** *Top*: We compare performance of different methods with model sizes ranging from 2M to 200M parameters. Across all model sizes and generalization levels, VIMA outperforms baseline variants. *Bottom*: For a fixed model size of 92M parameters we compare the effect of imitation learning dataset size with 0.1%, 1%, 10%, and full data. VIMA is extremely sample efficient and can achieve performance comparable to other methods with $10\times$ less data.

Tokenization. There are 3 formats of raw input in the prompt — text, image of a single object, and image of a full tabletop scene (*e.g.*, for *Rearrangement* or imitation from video frames). For **text inputs**, we use pre-trained T5 tokenizer and word embedding to obtain word tokens. For **images of full scenes**, we first extract individual objects using domain fine-tuned Mask R-CNN (He et al., 2017) (Appendix, Sec. C.4). Each object is represented as a bounding box and a cropped image. We then compute object tokens by encoding them with a bounding box encoder and a ViT (Dosovitskiy et al., 2020), respectively. Since Mask R-CNN is imperfect, the bounding boxes can be noisy and the cropped images may have irrelevant pixels. For **images of single objects**, we obtain tokens in the same way except with a dummy bounding box. Prompt tokenization produces a sequence of interleaved textual and visual tokens. We then follow the practice in Tsimpoukelli et al. (2021) and encode the prompt via a pre-trained T5 encoder (Raffel et al., 2020). Since T5 has been pre-trained on large text corpora, VIMA inherits the semantic understanding capability and robustness properties. To accommodate tokens from new modalities, we insert MLPs between non-textual tokens and T5.

Robot Controller. A challenging aspect of designing a multi-task policy is to select a suitable conditioning mechanism. In our schema (Fig. 3), the robot controller (decoder) is conditioned on the prompt sequence $\mathcal{P}$ by a series of cross-attention layers between $\mathcal{P}$ and the trajectory history sequence $\mathcal{H}$. We compute key $K_{\mathcal{P}}$ and value $V_{\mathcal{P}}$ sequences from the prompt and query $Q_{\mathcal{H}}$ from the trajectory history, following the encoder-decoder convention in Raffel et al. (2020). Each cross-attention layer then generates an output sequence $\mathcal{H}' = \text{softmax}\left(\frac{Q_{\mathcal{H}} K_{\mathcal{P}}^{\intercal}}{\sqrt{d}}\right) V_{\mathcal{P}}$, where d is the embedding dimension. Residual connections are added to connect higher layers with the input rollout trajectory sequence. The cross-attention design enjoys three advantages: 1) strengthened connection to prompt; 2) intact and deep flow of the original prompt tokens; and 3) better computational efficiency. VIMA decoder consists of L alternating cross-attention and self-attention layers. Finally, we follow common practice (Baker et al., 2022) to map predicted action tokens to discretized poses of the robot arm. See Appendix, Sec. C.2 for more details.

Training. We follow behavioral cloning to train our models by minimizing the negative log-likelihood of predicted actions. Concretely, for a trajectory with T steps, we optimize $\min_\theta \sum_{t=1}^{T} -\log \pi_\theta(a_t | \mathcal{P}, \mathcal{H})$. The entire training is conducted on an offline dataset with no simulator access. To make VIMA robust to detection inaccuracies and failures, we apply *object augmentation* by randomly injecting *false-positive* detection outputs. After training, we select model checkpoints for evaluation based on the aggregated accuracy on a held-out validation set. The evaluation involves interacting with the physics simulator. We follow the

best practices to train Transformer models. See Appendix, Sec. D for comprehensive training hyperparameters.

5. Experiments

In this section, we aim to answer three main questions:

1. What is the best recipe for building multi-task transformer-based robot agents with multimodal prompts?

2. What are the **scaling properties** of our approach in model capacity and data size?

3. How do different components, such as visual tokenizers, prompt conditioning, and prompt encoding, affect robot performance?

5.1. Baselines

Because there is no prior method that works out of the box with our multimodal prompting setup, we make our best effort to select a number of representative transformer-based agent architectures as baselines, and re-interpret them to be compatible with VIMA-BENCH:

Gato (Reed et al., 2022) introduces a decoder-only model that solves tasks from multiple domains where tasks are specified by prompting the model with the observation and action subsequence. For a fair comparison, we provide the same conditioning as VIMA, *i.e.*, our multimodal encoded prompts. Input images are divided into patches and encoded by a ViT model to produce observation tokens. This variant is referred to as "**VIMA-Gato**".

Flamingo (Alayrac et al., 2022) is a vision-language model that learns to generate textual completion in response to multimodal prompts. It embeds a variable number of prompt images into a fixed number of tokens via Perceiver (Jaegle et al., 2021b), and conditions the language decoder on the encoded prompt by cross-attention. Flamingo does not work with embodied agents out of the box. We adapt it to support decision-making by replacing the output layer with robot action heads. We denote the method as "**VIMA-Flamingo**".

VIMA-GPT is a decoder-only architecture conditioned on tokenized multimodal prompts. It autoregressively decodes the next actions given instructions and interaction histories. Similar to prior work (Chen et al., 2021; Janner et al., 2021), it encodes an image into a single *state* token by a ViT encoder and prepends the rollout trajectory with prompt tokens. This baseline does not use cross-attention.

A more detailed comparison between these variants can be found in Appendix, Sec. C.1.

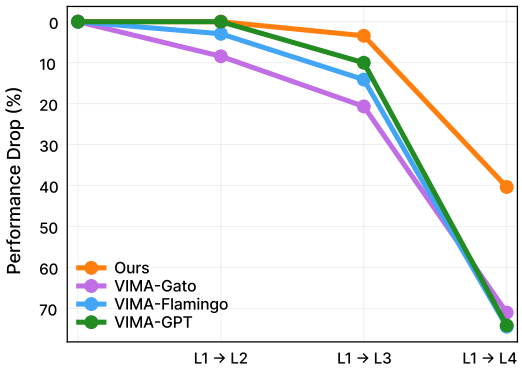

Figure 5: VIMA incurs much less performance drop than baselines as we evaluate on progressively harder settings.

5.2. Evaluation Results

We compare VIMA against the baseline variants on four levels of generalization provided in our benchmark for different model and training dataset sizes. Our empirical results demonstrate that VIMA's choice of object tokens combined with cross-attention conditioning is the most effective recipe among the model designs we consider.

Model Scaling. We train all methods for a spectrum of model capacities from 2M to 200M parameters, evenly spaced on the log scale (Fig. 4). The encoder size is kept constant (T5-Base, 111M) for all methods and excluded from the parameter count. Across *all* levels of zero-shot generalization, we find that VIMA strongly outperforms other alternatives. Although models like VIMA-Gato and VIMA-Flamingo show improved performance with bigger model sizes, VIMA consistently achieves superior performance over *all* model sizes. We note that this can only be achieved with *both* cross-attention and object token sequence representations — altering any component will significantly degrade the performance, especially in the low model capacity regime (ablations in Sec. 5.3).

Data Scaling. Next we investigate how different methods scale with varying dataset sizes. We compare model performance at 0.1%, 1%, 10% and full imitation learning dataset provided in VIMA-BENCH (Fig. 4). Note that to ensure all methods are fairly pre-trained on the same amount of data, we initialize baseline variants that directly learn from raw pixels with MVP pre-trained ViT (Xiao et al., 2022; Radosavovic et al., 2022). It is further MAE fine-tuned (He et al., 2021), using the *same* in-domain data as for the Mask R-CNN object detector. See Appendix, Sec. E.3 for detailed setup. VIMA is extremely sample efficient and, with just 1% of the data, can achieve performance similar to baseline methods trained with $10\times$ more data on L1 and L2 levels of generalization. In fact, for L4 we find that with

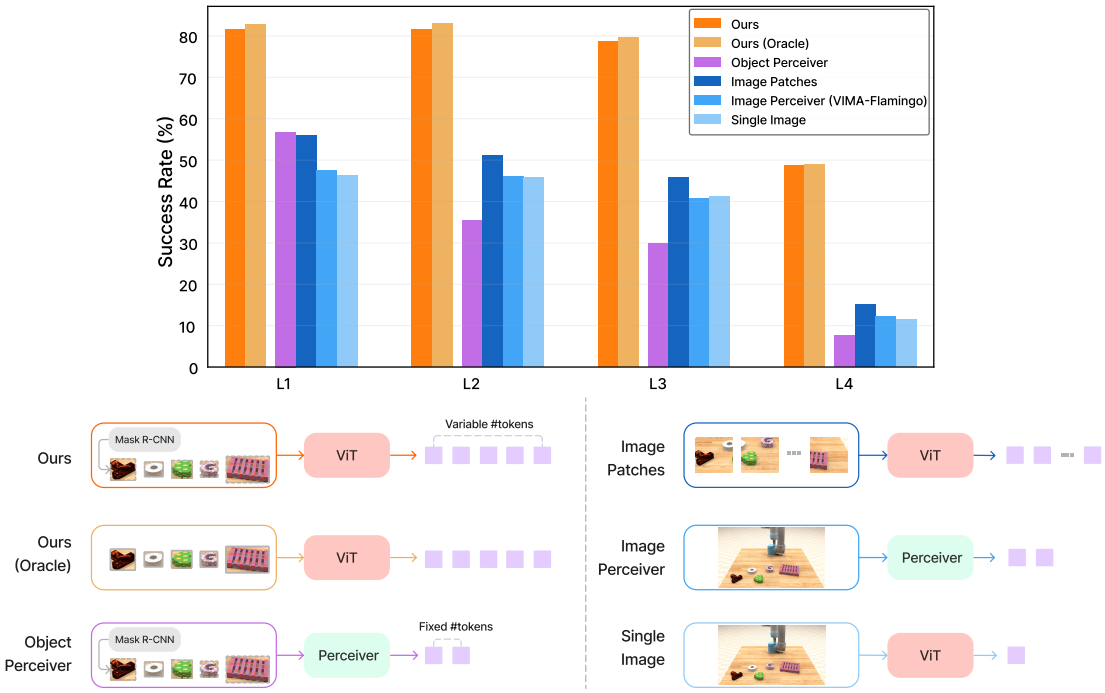

Figure 6: **Ablation on visual tokenizers.** We compare the performance of VIMA-200M model across different visual tokenizers. Our proposed object tokens outperform all methods that learn directly from raw pixels, and *Object Perceiver* that downsamples the object sequence to a fixed number of tokens.

just 1% of training data, VIMA already surpasses other variants trained with *entire* dataset. Finally, across all levels with just 10% of the data, VIMA can outperform other architectures trained with the full dataset by a significant margin. We hypothesize that the data efficiency can be attributed to the object-centric representation employed in the VIMA recipe, which is less prone to overfitting than learning directly from pixels in the low-data regime. This is consistent with findings from Sax et al. (2018), which demonstrates that embodied agents conditioned on mid-level visual representations tend to be significantly more sample-efficient than end-to-end control from raw pixels.

Progressive Generalization. Finally, we compare the relative performance degradation as we test the models on progressively challenging zero-shot evaluation levels without further fine-tuning (Fig. 5). Our method exhibits a minimal performance regression, especially between $L1 \rightarrow L2$ and $L1 \rightarrow L3$. In contrast, the baselines can degrade as much as 20%, particularly in more difficult generalization scenarios. Although all methods degrade significantly when evaluated on $L4$ (*Novel Tasks*), the performance drop for VIMA is only *half* as severe as all other baselines. These results suggest that VIMA has developed a more generalizable policy and robust representations than the alternative approaches.

5.3. Ablation Studies

Through extensive experiments, we ablate different design choices in VIMA and study their impact on robot decision making. We focus on four aspects: visual tokenization, prompt conditioning, prompt-encoding language models, and policy robustness against distractions and corruptions.

Visual Tokenization. As explained in Sec. 4, VIMA processes the prompt and observation images into a variable number of object tokens with a domain fine-tuned Mask R-CNN implementation. How important is this particular choice of visual tokenizer? We study 5 different variants and empirically evaluate their 4 levels of generalization performance on VIMA-BENCH. 1) **Ours (Oracle)**: instead of using Mask R-CNN, we directly read out the ground-truth bounding box from the simulator. In other words, we use a perfect object detector to estimate the upper bound on the performance of this study; 2) **Object Perceiver**: we apply a Perceiver module to convert the variable number of objects detected in each frame to a *fixed* number of tokens. Perceiver is more computationally efficient because it reduces the average sequence length; 3) **Image Perceiver**: the same architecture as the *Perceiver Resampler* in VIMA-Flamingo, which converts an image to a small, fixed number of tokens; 4) **Image patches**: following VIMA-Gato, we divide an RGB frame into square patches, and extract ViT embedding

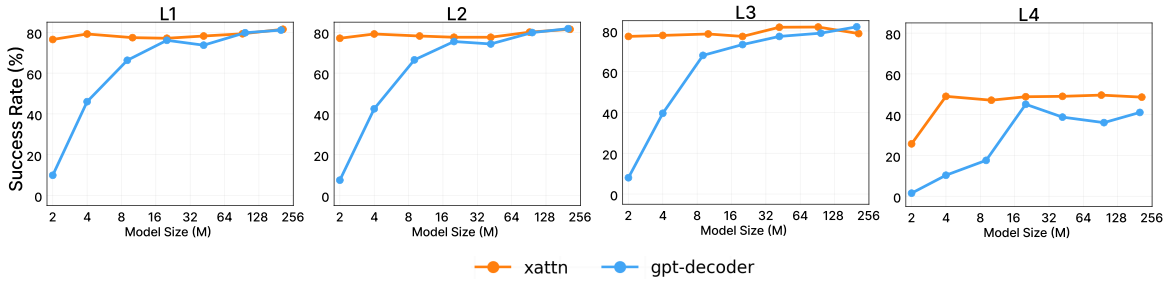

Figure 7: **Ablation on prompt conditioning.** We compare our method (*xattn*: cross-attention prompt conditioning) with a vanilla transformer decoder (*gpt-decoder*) across different model sizes. Cross-attention is especially helpful in low-parameter regime and for harder generalization tasks.

tokens. The number of patches is greater than the output of Image Perceiver; 5) **Single image**: VIMA-GPT's tokenizer, which encodes one image into a single token.

Fig. 6 shows the ablation results. We highlight a few findings. First, we note that our Mask R-CNN detection pipeline (Appendix, Sec. C.4) **incurs a minimal performance loss** compared to the oracle bounding boxes, thanks to the object augmentation (Sec. 4) that boosts robustness during training. Second, tokenizing from raw pixels (Image Perceiver, patches, or single embedding) consistently underperforms our object-centric format. We hypothesize that these tokenizers have to allocate extra internal capacity to parse the objects from low-level pixels, which likely impedes learning. Sax et al. (2018) echoes our finding that using mid-level vision can greatly improve agent generalization compared to an end-to-end pipeline. Third, even though *Ours* and *Object Perceiver* both use the same object bounding box inputs, the latter is significantly worse in decision making. We conclude that it is important to directly pass the **variable-length object sequence** to the robot controller rather than downsampling to a fixed number of tokens.

Prompt Conditioning. VIMA conditions the robot controller (decoder) on the encoded prompt by cross-attention. A simple alternative is to concatenate the prompt $\mathcal{P}$ and interaction history $\mathcal{H}$ into one big sequence, and then apply a decoder-only transformer like GPT (Radford et al., 2018) to predict actions. In this ablation, we keep the object tokenizer constant and only switch the conditioning mechanism to causal sequence modeling. Note that this variant is conceptually "VIMA-Gato with object tokens". Fig. 7 shows the comparison of VIMA (`xattn`) and the `gpt-decoder` variant across 4 generalization levels. While the variant achieves comparable performance in larger models, cross-attention still dominates in the small-capacity range and generalizes better in the most challenging L4 (*Novel Task*) setting. Our hypothesis is that cross-attention helps the controller stay better focused on the prompt instruction at each interaction step. This bears a resemblance to the empirical results in Sanh et al. (2021); Wang et al. (2022b), which show that well-tuned encoder-decoder architectures can outperform GPT-3 in zero-shot generalization.

Prompt Encoding. We vary the size of the pre-trained T5 encoder to study the effect of prompt encoding. We experiment with three T5 capacities: `small` (30M), `base` (111M), and `large` (368M). We further fix the parameter count of the decision-making part to be 200M. For all T5 variants, we fine-tune the last two layers and freeze all other layers. We find no significant difference among the variants (Appendix, Sec. E.4), thus we set `base` as default for all our models.

Policy Robustness. We study the policy robustness against increasing number of distractors and corrupted task specifications, including incomplete prompts (randomly masking out words with `<UNK>` token) and corrupted prompts (randomly swapping words, which could have changed the task meaning altogether). See Appendix, Sec. E.5 for exact setup and results. VIMA exhibits minimal performance degradation with increased distractors and minor decrease with corrupted prompts. We attribute this robustness to the high-quality pre-trained T5 backbone.

6. Related Work

Multi-Task Learning by Sequence Modeling. Transformers (Vaswani et al., 2017) have enabled task unification across many AI domains (Brown et al., 2020; Chen et al., 2022a;b; Lu et al., 2022; Wang et al., 2022c). For example, in **NLP**, the Natural Language Decathlon (McCann et al., 2018) adopts a consistent question-answering format for a suite of 10 NLP tasks. T5 (Raffel et al., 2020) unifies all language problems into the same text-to-text format. GPT-3 (Brown et al., 2020) and Megatron (Shoeybi et al., 2019) demonstrate emergent behaviours of intuitive task specifications by zero-shot prompting. In **computer vision**, Pix2Seq (Chen et al., 2022b) casts many vision problems into a unified sequence format. Florence (Yuan et al., 2021),

BiT (Kolesnikov et al., 2020), and MuST (Ghiasi et al., 2021) pre-train shared backbone models at scale for general visual representations and transfer them to downstream tasks. In **multimodal learning**, Perceiver (Jaegle et al., 2021b;a) proposes an efficient architecture to handle structured inputs and outputs. Flamingo (Alayrac et al., 2022) and Frozen (Tsimpoukelli et al., 2021) design a universal API that ingests interleaving sequences of images and text and generates free-form text. Gato (Reed et al., 2022) is a massively multi-task model across NLP, vision, and embodied agents. Our work is most similar in spirit to Gato, but we focus primarily on enabling an intuitive multimodal prompting interface for a generalist robot agent.

Foundation Models for Embodied Agents. Foundation models (Bommasani et al., 2021) have demonstrated strong emergent properties. There are many ongoing efforts to replicate this success for embodied agents (Yang et al., 2023), focusing on 3 aspects. 1) **Transformer agent architecture**: Decision Transformer and Trajectory Transformer (Chen et al., 2021; Janner et al., 2021; Zheng et al., 2022; Xu et al., 2022) leverage the powerful self-attention models for sequential decision making. CLIPort (Shridhar et al., 2021), Perceiver-Actor (Shridhar et al., 2022), and RT-1 (Brohan et al., 2022) apply large transformers to robot manipulation tasks. BeT (Shafiullah et al., 2022) and C-BeT (Cui et al., 2022) design novel techniques to learn from demonstrations with multiple modes with transformers. 2) **Pre-training for better representations**: MaskViT (Gupta et al., 2022b), R3M (Nair et al., 2022), VIP (Ma et al., 2022), and VC-1 (Majumdar et al., 2023) pre-train general visual representations for robotic perception. Li et al. (2022b) fine-tunes from LLM checkpoints to accelerate policy learning. MineDojo (Fan et al., 2022) and Ego4D (Grauman et al., 2021) provide large-scale multimodal databases to facilitate scalable policy training. 3) **LLMs for robot learning**: SayCan (Ahn et al., 2022) leverages PaLM (Chowdhery et al., 2022) for zero-shot concept grounding. Huang et al. (2022a), Inner Monologue (Huang et al., 2022b) and LM-Nav (Shah et al., 2022) apply LLMs to long-horizon robot planning. PaLM-E (Driess et al., 2023) is instead a multimodal language model that can be repurposed for sequential robotic manipulation planning. Ours differs from these works in our novel multimodal prompting formulation, which existing LLMs do not easily support.

Robot Manipulation and Benchmarks. A wide range of robot manipulation tasks require different skills and task specification formats, such as instruction following (Stepputtis et al., 2020), one-shot imitation (Finn et al., 2017; Duan et al., 2017), rearrangement (Batra et al., 2020), constraint satisfaction (Brunke et al., 2021a), and reasoning (Shridhar et al., 2020). Multiple physics simulation benchmarks are introduced to study the above tasks. For example, iGibson (Shen et al., 2020; Li et al., 2021; Srivastava et al., 2021; Li et al., 2022a) simulates interactive household scenarios. Ravens (Zeng et al., 2020) and Robosuite (Zhu et al., 2020; Fan et al., 2021) design various tabletop manipulation tasks with realistic robot arms. CALVIN (Mees et al., 2021) develops long-horizon language-conditioned tasks. Meta-World (Yu et al., 2019) is a widely used simulator benchmark studying robotics manipulation with tabletop settings. CausalWorld (Ahmed et al., 2021) is a benchmark for causal structure and transfer learning in manipulation, requiring long-horizon planning and precise low-level motor control. AI2-THOR (Ehsani et al., 2021; Deitke et al., 2022) is a framework that supports visual object manipulation and procedural generation of environments. Our VIMA-BENCH is the first robot learning benchmark to support multimodal-prompted tasks. We also standardize the evaluation protocol to systematically measure an agent's generalization capabilities.

An extended review can be found in Appendix, Sec. F.

7. Conclusion

In this work, we introduce a novel *multimodal* prompting formulation that converts diverse robot manipulation tasks into a uniform sequence modeling problem. We instantiate this formulation in VIMA-BENCH, a diverse benchmark with multimodal tasks and systematic evaluation protocols for generalization. We propose VIMA, a conceptually simple transformer-based agent capable of solving tasks such as visual goal reaching, one-shot video imitation, and novel concept grounding with a single model. Through comprehensive experiments, we show that VIMA exhibits strong model scalability and zero-shot generalization. Therefore, we recommend our agent design as a solid starting point for future work.

Acknowledgement

We are extremely grateful to Shyamal Buch, Jonathan Tremblay, Ajay Mandlekar, Chris Choy, De-An Huang, Silvio Savarese, Fei Xia, Josiah Wong, Abhishek Joshi, Soroush Nasiriany, and many other colleagues and friends for their helpful feedback and insightful discussions. We also thank the anonymous reviewers for offering us highly constructive advice and kind encouragement during the review period. NVIDIA provides the necessary computing resource and infrastructure for this project. This work is done during Yunfan Jiang and Guanzhi Wang's internships at NVIDIA. Guanzhi Wang is supported by the Kortschak fellowship in Computing and Mathematical Sciences at Caltech.

References

Abramson, J., Ahuja, A., Barr, I., Brussee, A., Carnevale, F., Cassin, M., Chhaparia, R., Clark, S., Damoc, B., Dudzik, A., Georgiev, P., Guy, A., Harley, T., Hill, F., Hung, A., Kenton, Z., Landon, J., Lillicrap, T., Mathewson, K., Mokrá, S., Muldal, A., Santoro, A., Savinov, N., Varma, V., Wayne, G., Williams, D., Wong, N., Yan, C., and Zhu, R. Imitating interactive intelligence. *arXiv preprint arXiv: Arxiv-2012.05672*, 2020.

Aceituno, B., Rodriguez, A., Tulsiani, S., Gupta, A., and Mukadam, M. A differentiable recipe for learning visual non-prehensile planar manipulation. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=f7KaqYLO3iE.

Agrawal, P. The task specification problem. In Faust, A., Hsu, D., and Neumann, G. (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1745–1751. PMLR, 08-11 Nov 2022. URL https://proceedings.mlr.press/v164/agrawal22a.html.

Ahmed, O., Träuble, F., Goyal, A., Neitz, A., Wuthrich, M., Bengio, Y., Schölkopf, B., and Bauer, S. Causalworld: A robotic manipulation benchmark for causal structure and transfer learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=SK7A5pdrgov.

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R. J., Jeffrey, K., Jesmonth, S., Joshi, N. J., Julian, R., Kalashnikov, D., Kuang, Y., Lee, K.-H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., and Yan, M. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv: Arxiv-2204.01691*, 2022.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv: Arxiv-2204.14198*, 2022.

Baker, B., Akkaya, I., Zhokhov, P., Huizinga, J., Tang, J., Ecoffet, A., Houghton, B., Sampedro, R., and Clune, J. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *arXiv preprint arXiv: Arxiv-2206.11795*, 2022.

Batra, D., Chang, A. X., Chernova, S., Davison, A. J., Deng, J., Koltun, V., Levine, S., Malik, J., Mordatch, I., Mottaghi, R., Savva, M., and Su, H. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv: Arxiv-2011.01975*, 2020.

Berscheid, L., Meißner, P., and Kröger, T. Self-supervised learning for precise pick-and-place without object model. *arXiv preprint arXiv: Arxiv-2006.08373*, 2020.

Bharadhwaj, H., Kumar, A., Rhinehart, N., Levine, S., Shkurti, F., and Garg, A. Conservative safety critics for exploration. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=iaO86DUuKi.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. On the opportunities and risks of foundation models. *arXiv preprint arXiv: Arxiv-2108.07258*, 2021.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jackson, T., Jesmonth, S., Joshi, N. J., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, K.-H., Levine, S., Lu, Y., Malla,

U., Manjunath, D., Mordatch, I., Nachum, O., Parada, C., Peralta, J., Perez, E., Pertsch, K., Quiambao, J., Rao, K., Ryoo, M., Salazar, G., Sanketi, P., Sayed, K., Singh, J., Sontakke, S., Stone, A., Tan, C., Tran, H., Vanhoucke, V., Vega, S., Vuong, Q., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., and Zitkovich, B. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv: Arxiv-2212.06817*, 2022.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, volume 33, pp. 1877–1901, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., and Schoellig, A. P. Safe learning in robotics: From learning-based control to safe reinforcement learning. *arXiv preprint arXiv: Arxiv-2108.06266*, 2021a.

Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., and Schoellig, A. P. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning, December 2021b. URL http://arxiv.org/abs/2108.06266. arXiv:2108.06266 [cs, eess].

Buch, S., Eyzaguirre, C., Gaidon, A., Wu, J., Fei-Fei, L., and Niebles, J. C. Revisiting the "video" in video-language understanding. *CVPR*, 2022.

Bucker, A., Figueredo, L., Haddadin, S., Kapoor, A., Ma, S., Vemprala, S., and Bonatti, R. Latte: Language trajectory transformer. *arXiv preprint arXiv: Arxiv-2208.02918*, 2022.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 15084–15097, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html.

Chen, T., Saxena, S., Li, L., Fleet, D. J., and Hinton, G. E. Pix2seq: A language modeling framework for object detection. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022a. URL https://openreview.net/forum?id=e42KbIw6Wb.

Chen, T., Saxena, S., Li, L., Lin, T.-Y., Fleet, D. J., and Hinton, G. A unified sequence interface for vision tasks. *arXiv preprint arXiv: Arxiv-2206.07669*, 2022b.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv: Arxiv-2204.02311*, 2022.

Collins, J., Chand, S., Vanderkop, A., and Howard, D. A review of physics simulators for robotic applications. *IEEE Access*, 9:51416–51431, 2021.

Coumans, E. and Bai, Y. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

Cui, Z. J., Wang, Y., Shafiullah, N. M. M., and Pinto, L. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv: Arxiv-2210.10047*, 2022.

Dasari, S. and Gupta, A. Transformers for one-shot visual imitation. In Kober, J., Ramos, F., and Tomlin, C. J. (eds.), *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, volume 155 of *Proceedings of Machine Learning Research*, pp. 2071–2084. PMLR, 2020. URL https://proceedings.mlr.press/v155/dasari21a.html.

Dasari, S., Ebert, F., Tian, S., Nair, S., Bucher, B., Schmeckpeper, K., Singh, S., Levine, S., and Finn, C. Robonet:

Large-scale multi-robot learning. In Kaelbling, L. P., Kragic, D., and Sugiura, K. (eds.), *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pp. 885–897. PMLR, 2019. URL http://proceedings.mlr.press/v100/dasari20a.html.

Deitke, M., VanderBilt, E., Herrasti, A., Weihs, L., Salvador, J., Ehsani, K., Han, W., Kolve, E., Farhadi, A., Kembhavi, A., and Mottaghi, R. Procthor: Large-scale embodied ai using procedural generation. *arXiv preprint arXiv: Arxiv-2206.06994*, 2022.

Devin, C., Rowghanian, P., Vigorito, C., Richards, W., and Rohanimanesh, K. Self-supervised goal-conditioned pick and place. *arXiv preprint arXiv: Arxiv-2008.11466*, 2020.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv: Arxiv-2010.11929*, 2020.

Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T. B., and Vanhoucke, V. Google scanned objects: A high-quality dataset of 3d scanned household items. *arXiv preprint arXiv:2204.11918*, 2022.

Driess, D., Xia, F., Sajjadi, M. S. M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., and Florence, P. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv: Arxiv-2303.03378*, 2023.

Duan, J., Yu, S., Tan, H. L., Zhu, H., and Tan, C. A survey of embodied AI: from simulators to research tasks. *IEEE Trans. Emerg. Top. Comput. Intell.*, 6(2):230–244, 2022. doi: 10.1109/TETCI.2022.3141105. URL https://doi.org/10.1109/TETCI.2022.3141105.

Duan, Y., Andrychowicz, M., Stadie, B. C., Ho, J., Schneider, J., Sutskever, I., Abbeel, P., and Zaremba, W. One-shot imitation learning. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1087–1098, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/ba3866600c3540f67c1e9575e213be0a-Abstract.html.

Ehsani, K., Han, W., Herrasti, A., VanderBilt, E., Weihs, L., Kolve, E., Kembhavi, A., and Mottaghi, R. Manipulathor: A framework for visual object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4497–4506, June 2021.

Fan, L., Zhu, Y., Zhu, J., Liu, Z., Zeng, O., Gupta, A., Creus-Costa, J., Savarese, S., and Fei-Fei, L. Surreal: Open-source reinforcement learning framework and robot manipulation benchmark. In Billard, A., Dragan, A., Peters, J., and Morimoto, J. (eds.), *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 767–782. PMLR, 29-31 Oct 2018. URL https://proceedings.mlr.press/v87/fan18a.html.

Fan, L., Zhu, Y., Zhu, J., Liu, Z., Zeng, O., Gupta, A., Creus-Costa, J., Savarese, S., and Fei-Fei, L. Surreal-system: Fully-integrated stack for distributed deep reinforcement learning. *arXiv preprint arXiv: Arxiv-1909.12989*, 2019.

Fan, L., Wang, G., Huang, D., Yu, Z., Fei-Fei, L., Zhu, Y., and Anandkumar, A. SECANT: self-expert cloning for zero-shot generalization of visual policies. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3088–3099. PMLR, 2021. URL http://proceedings.mlr.press/v139/fan21c.html.

Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.-A., Zhu, Y., and Anandkumar, A. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *arXiv preprint arXiv: Arxiv-2206.08853*, 2022.

Finn, C., Yu, T., Zhang, T., Abbeel, P., and Levine, S. One-shot visual imitation learning via meta-learning. *arXiv preprint arXiv: Arxiv-1709.04905*, 2017.

Florence, P., Manuelli, L., and Tedrake, R. Self-supervised correspondence in visuomotor policy learning. *arXiv preprint arXiv: Arxiv-1909.06933*, 2019.

Fu, T.-J., Li, L., Gan, Z., Lin, K., Wang, W. Y., Wang, L., and Liu, Z. Violet : End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv: Arxiv-2111.12681*, 2021.

Gan, C., Zhou, S., Schwartz, J., Alter, S., Bhandwaldar, A., Gutfreund, D., Yamins, D. L. K., DiCarlo, J. J., McDermott, J., Torralba, A., and Tenenbaum, J. B. The threedworld transport challenge: A visually guided task-and-motion planning benchmark for physically realistic embodied ai. *arXiv preprint arXiv: Arxiv-2103.14025*, 2021.

Ghiasi, G., Zoph, B., Cubuk, E. D., Le, Q. V., and Lin, T. Multi-task self-training for learning general representations. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 8836–8845. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00873. URL https://doi.org/10.1109/ICCV48922.2021.00873.

Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., Martin, M., Nagarajan, T., Radosavovic, I., Ramakrishnan, S. K., Ryan, F., Sharma, J., Wray, M., Xu, M., Xu, E. Z., Zhao, C., Bansal, S., Batra, D., Cartillier, V., Crane, S., Do, T., Doulaty, M., Erapalli, A., Feichtenhofer, C., Fragomeni, A., Fu, Q., Gebreselasie, A., Gonzalez, C., Hillis, J., Huang, X., Huang, Y., Jia, W., Khoo, W., Kolar, J., Kottur, S., Kumar, A., Landini, F., Li, C., Li, Y., Li, Z., Mangalam, K., Modhugu, R., Munro, J., Murrell, T., Nishiyasu, T., Price, W., Puentes, P. R., Ramazanova, M., Sari, L., Somasundaram, K., Southerland, A., Sugano, Y., Tao, R., Vo, M., Wang, Y., Wu, X., Yagi, T., Zhao, Z., Zhu, Y., Arbelaez, P., Crandall, D., Damen, D., Farinella, G. M., Fuegen, C., Ghanem, B., Ithapu, V. K., Jawahar, C. V., Joo, H., Kitani, K., Li, H., Newcombe, R., Oliva, A., Park, H. S., Rehg, J. M., Sato, Y., Shi, J., Shou, M. Z., Torralba, A., Torresani, L., Yan, M., and Malik, J. Ego4d: Around the world in 3,000 hours of egocentric video. *arXiv preprint arXiv: Arxiv-2110.07058*, 2021.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In Kaelbling, L. P., Kragic, D., and Sugiura, K. (eds.), *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1025–1037. PMLR, 2019. URL http://proceedings.mlr.press/v100/gupta20a.html.

Gupta, A., Fan, L., Ganguli, S., and Fei-Fei, L. Metamorph: Learning universal controllers with transformers. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=Opmqtk_GvYL.

Gupta, A., Tian, S., Zhang, Y., Wu, J., Martín-Martín, R., and Fei-Fei, L. Maskvit: Masked visual pre-training for video prediction. *arXiv preprint arXiv: Arxiv-2206.11894*, 2022b.

Hansen, N., Yuan, Z., Ze, Y., Mu, T., Rajeswaran, A., Su, H., Xu, H., and Wang, X. On pre-training for visuomotor control: Revisiting a learning-from-scratch baseline. *arXiv preprint arXiv: Arxiv-2212.05749*, 2022.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. *arXiv preprint arXiv: Arxiv-1703.06870*, 2017.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv: Arxiv-2111.06377*, 2021.

Heibeck, T. H. and Markman, E. M. Word learning in children: An examination of fast mapping. *Child development*, pp. 1021–1034, 1987.

Hermann, K. M., Hill, F., Green, S., Wang, F., Faulkner, R., Soyer, H., Szepesvari, D., Czarnecki, W. M., Jaderberg, M., Teplyashin, D., Wainwright, M., Apps, C., Hassabis, D., and Blunsom, P. Grounded language learning in a simulated 3d world. *arXiv preprint arXiv: Arxiv-1706.06551*, 2017.

Hill, F., Tieleman, O., von Glehn, T., Wong, N., Merzic, H., and Clark, S. Grounded language learning fast and slow. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=wpSWuz_hyqA.

Huang, D., Xu, D., Zhu, Y., Garg, A., Savarese, S., Fei-Fei, L., and Niebles, J. C. Continuous relaxation of symbolic planner for one-shot imitation learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, SAR, China, November 3-8, 2019*, pp. 2635–2642. IEEE, 2019. doi: 10.1109/IROS40897.2019.8967761. URL https://doi.org/10.1109/IROS40897.2019.8967761.

Huang, W., Abbeel, P., Pathak, D., and Mordatch, I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9118–9147. PMLR, 2022a. URL https://proceedings.mlr.press/v162/huang22a.html.

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., Sermanet, P., Brown, N., Jackson, T., Luu, L., Levine, S., Hausman, K., and Ichter, B. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv: Arxiv-2207.05608*, 2022b.

Jaegle, A., Borgeaud, S., Alayrac, J.-B., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., Hénaff, O., Botvinick, M. M., Zisserman, A., Vinyals, O., and Carreira, J. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv: Arxiv-2107.14795*, 2021a.

Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., and Carreira, J. Perceiver: General perception with iterative attention. *arXiv preprint arXiv: Arxiv-2103.03206*, 2021b.

James, S., Ma, Z., Arrojo, D. R., and Davison, A. J. Rlbench: The robot learning benchmark & learning environment. *arXiv preprint arXiv: Arxiv-1909.12271*, 2019.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 1273–1286, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/099fe6b0b444c23836c4a5d07346082b-Abstract.html.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., and Shah, M. Transformers in vision: A survey. *arXiv preprint arXiv: Arxiv-2101.01169*, 2021.

Khandelwal, A., Weihs, L., Mottaghi, R., and Kembhavi, A. Simple but effective: Clip embeddings for embodied ai. *arXiv preprint arXiv: Arxiv-2111.09888*, 2021.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. Segment anything. *arXiv preprint arXiv: Arxiv-2304.02643*, 2023.

Kokkinos, I. Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. *arXiv preprint arXiv: Arxiv-1609.02132*, 2016.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (bit): General visual representation learning. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J. (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12350 of *Lecture Notes in Computer Science*, pp. 491–507. Springer, 2020. doi: 10.1007/978-3-030-58558-7_29. URL https://doi.org/10.1007/978-3-030-58558-7_29.

Kolesnikov, A., Pinto, A. S., Beyer, L., Zhai, X., Harmsen, J., and Houlsby, N. Uvim: A unified modeling approach for vision with learned guiding codes. *arXiv preprint arXiv: Arxiv-2205.10337*, 2022.

Li, C., Xia, F., Martín-Martín, R., Lingelbach, M., Srivastava, S., Shen, B., Vainio, K. E., Gokmen, C., Dharan, G., Jain, T., Kurenkov, A., Liu, C. K., Gweon, H., Wu, J., Fei-Fei, L., and Savarese, S. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In Faust, A., Hsu, D., and Neumann, G. (eds.), *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pp. 455–465. PMLR, 2021. URL https://proceedings.mlr.press/v164/li22b.html.

Li, C., Zhang, R., Wong, J., Gokmen, C., Srivastava, S., Martín-Martín, R., Wang, C., Levine, G., Lingelbach, M., Sun, J., Anvari, M., Hwang, M., Sharma, M., Aydin, A., Bansal, D., Hunter, S., Kim, K.-Y., Lou, A., Matthews, C. R., Villa-Renteria, I., Tang, J. H., Tang, C., Xia, F., Savarese, S., Gweon, H., Liu, K., Wu, J., and Fei-Fei, L. BEHAVIOR-1k: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In *6th Annual Conference on Robot Learning*, 2022a. URL https://openreview.net/forum?id=_8DoIe8G3t.

Li, S., Puig, X., Paxton, C., Du, Y., Wang, C., Fan, L., Chen, T., Huang, D.-A., Akyürek, E., Anandkumar, A., Andreas, J., Mordatch, I., Torralba, A., and Zhu, Y. Pre-trained language models for interactive decision-making. *arXiv preprint arXiv: Arxiv-2202.01771*, 2022b.

Lim, M. H., Zeng, A., Ichter, B., Bandari, M., Coumans, E., Tomlin, C., Schaal, S., and Faust, A. Multi-task learning with sequence-conditioned transporter networks. *arXiv preprint arXiv: Arxiv-2109.07578*, 2021.

Lin, T., Wang, Y., Liu, X., and Qiu, X. A survey of transformers. *arXiv preprint arXiv: Arxiv-2106.04554*, 2021.

Liu, F., Liu, H., Grover, A., and Abbeel, P. Masked autoencoding for scalable and generalizable decision making. *arXiv preprint arXiv: Arxiv-2211.12740*, 2022a.

Liu, H., Lee, L., Lee, K., and Abbeel, P. Instructrl: Instruction-following agents with jointly pre-trained vision-language models. *arXiv preprint arXiv: Arxiv-2210.13431*, 2022b.

Liu, Z., Liu, W., Qin, Y., Xiang, F., Gou, M., Xin, S., Roa, M. A., Calli, B., Su, H., Sun, Y., et al. Ocrtoc: A cloud-based competition and benchmark for robotic grasping and manipulation. *IEEE Robotics and Automation Letters*, 7(1):486–493, 2021.

Loshchilov, I. and Hutter, F. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Lu, J., Goswami, V., Rohrbach, M., Parikh, D., and Lee, S. 12-in-1: Multi-task vision and language representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 10434–10443. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01045. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Lu_12-in-1_Multi-Task_Vision_and_Language_Representation_Learning_CVPR_2020_paper.html.

Lu, J., Clark, C., Zellers, R., Mottaghi, R., and Kembhavi, A. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv: Arxiv-2206.08916*, 2022.

Lynch, C. and Sermanet, P. Language conditioned imitation learning over unstructured data. In Shell, D. A., Toussaint, M., and Hsieh, M. A. (eds.), *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021. doi: 10.15607/RSS.2021.XVII.047. URL https://doi.org/10.15607/RSS.2021.XVII.047.

Ma, Y. J., Sodhani, S., Jayaraman, D., Bastani, O., Kumar, V., and Zhang, A. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv: Arxiv-2210.00030*, 2022.

Majumdar, A., Yadav, K., Arnaud, S., Ma, Y. J., Chen, C., Silwal, S., Jain, A., Berges, V.-P., Abbeel, P., Malik, J., Batra, D., Lin, Y., Maksymets, O., Rajeswaran, A., and Meier, F. Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv: Arxiv-2303.18240*, 2023.

Mandlekar, A., Xu, D., Wong, J., Nasiriany, S., Wang, C., Kulkarni, R., Fei-Fei, L., Savarese, S., Zhu, Y., and Martín-Martín, R. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv: Arxiv-2108.03298*, 2021.

McCann, B., Keskar, N. S., Xiong, C., and Socher, R. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv: Arxiv-1806.08730*, 2018.

Mees, O., Hermann, L., Rosete-Beas, E., and Burgard, W. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *arXiv preprint arXiv: Arxiv-2112.03227*, 2021.

Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., Wang, X., Zhai, X., Kipf, T., and Houlsby, N. Simple open-vocabulary object detection with vision transformers. *arXiv preprint arXiv: Arxiv-2205.06230*, 2022.

Mittal, M., Yu, C., Yu, Q., Liu, J., Rudin, N., Hoeller, D., Yuan, J. L., Tehrani, P. P., Singh, R., Guo, Y., Mazhar, H., Mandlekar, A., Babich, B., State, G., Hutter, M., and Garg, A. Orbit: A unified simulation framework for interactive robot learning environments. *arXiv preprint arXiv: Arxiv-2301.04195*, 2023.

Morrical, N., Tremblay, J., Birchfield, S., and Wald, I. NVISII: Nvidia scene imaging interface, 2020. https://github.com/owl-project/NVISII/.

Nair, S., Rajeswaran, A., Kumar, V., Finn, C., and Gupta, A. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv: Arxiv-2203.12601*, 2022.

OpenAI, Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., d. O. Pinto, H. P., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F., and Zhang, S. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv: Arxiv-1912.06680*, 2019.

Paine, T. L., Colmenarejo, S. G., Wang, Z., Reed, S., Aytar, Y., Pfaff, T., Hoffman, M. W., Barth-Maron, G., Cabi, S., Budden, D., and de Freitas, N. One-shot high-fidelity imitation: Training large-scale deep nets with rl. *arXiv preprint arXiv: Arxiv-1810.05017*, 2018.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., and Torralba, A. Virtualhome: Simulating household activities via programs. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 8494–8502. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00886. URL http://openaccess.thecvf.com/content_

cvpr_2018/html/Puig_VirtualHome_Simulating_Household_CVPR_2018_paper.html.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. *OpenAI*, 2018.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Radosavovic, I., Xiao, T., James, S., Abbeel, P., Malik, J., and Darrell, T. Real-world robot learning with masked visual pre-training. *CoRL*, 2022.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Ravichandar, H., Polydoros, A. S., Chernova, S., and Billard, A. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3:297–330, 2020.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., and de Freitas, N. A generalist agent. *arXiv preprint arXiv: Arxiv-2205.06175*, 2022.

Reid, M., Yamada, Y., and Gu, S. S. Can wikipedia help offline reinforcement learning? *arXiv preprint arXiv: Arxiv-2201.12122*, 2022.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv: Arxiv-1506.01497*, 2015.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L. A., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N. V., Datta, D., Chang, J., Jiang, M. T.-J., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Févry, T., Fries, J. A., Teehan, R., Biderman, S. R., Gao, L., Bers, T., Wolf, T., and Rush, A. M. Multitask prompted training enables zero-shot task generalization. *Iclr*, 2021.

Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., and Batra, D. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Sax, A., Emi, B., Zamir, A. R., Guibas, L., Savarese, S., and Malik, J. Mid-level visual representations improve generalization and sample efficiency for learning visuomotor policies. *arXiv preprint arXiv: Arxiv-1812.11971*, 2018.

Shafiullah, N. M. M., Cui, Z. J., Altanzaya, A., and Pinto, L. Behavior transformers: Cloning k modes with one stone. *arXiv preprint arXiv: Arxiv-2206.11251*, 2022.

Shah, D., Osinski, B., Ichter, B., and Levine, S. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. *arXiv preprint arXiv: Arxiv-2207.04429*, 2022.

Shazeer, N. Glu variants improve transformer. *arXiv preprint arXiv: Arxiv-2002.05202*, 2020.

Shen, B., Xia, F., Li, C., Martín-Martín, R., Fan, L., Wang, G., Pérez-D'Arpino, C., Buch, S., Srivastava, S., Tchapmi, L. P., Tchapmi, M. E., Vainio, K., Wong, J., Fei-Fei, L., and Savarese, S. igibson 1.0: a simulation environment for interactive tasks in large realistic scenes. *arXiv preprint arXiv: Arxiv-2012.02924*, 2020.

Shi, T. T., Karpathy, A., Fan, L. J., Hernandez, J., and Liang, P. World of bits: an open-domain platform for web-based agents. *ICML*, 2017. URL https://dl.acm.org/doi/10.5555/3305890.3306005.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., and Fox, D. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 10737–10746. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01075. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Shridhar_ALFRED_A_Benchmark_for_Interpreting_Grounded_Instructions_for_Everyday_Tasks_CVPR_2020_paper.html.

Shridhar, M., Manuelli, L., and Fox, D. Cliport: What and where pathways for robotic manipulation. *arXiv preprint arXiv: Arxiv-2109.12098*, 2021.

Shridhar, M., Manuelli, L., and Fox, D. Perceiver-actor: A multi-task transformer for robotic manipulation. *arXiv preprint arXiv: Arxiv-2209.05451*, 2022.

Sieb, M., Xian, Z., Huang, A., Kroemer, O., and Fragkiadaki, K. Graph-structured visual imitation. *arXiv preprint arXiv: Arxiv-1907.05518*, 2019.

Song, S., Zeng, A., Lee, J., and Funkhouser, T. Grasping in the wild:learning 6dof closed-loop grasping from low-cost demonstrations. *arXiv preprint arXiv: Arxiv-1912.04344*, 2019.

Srinivasan, K., Eysenbach, B., Ha, S., Tan, J., and Finn, C. Learning to be safe: Deep rl with a safety critic. *arXiv preprint arXiv: Arxiv-2010.14603*, 2020.

Srivastava, S., Li, C., Lingelbach, M., Martín-Martín, R., Xia, F., Vainio, K. E., Lian, Z., Gokmen, C., Buch, S., Liu, C. K., Savarese, S., Gweon, H., Wu, J., and Fei-Fei, L. BEHAVIOR: benchmark for everyday household activities in virtual, interactive, and ecological environments. In Faust, A., Hsu, D., and Neumann, G. (eds.), *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pp. 477–490. PMLR, 2021. URL https://proceedings.mlr.press/v164/srivastava22a.html.

Stengel-Eskin, E., Hundt, A., He, Z., Murali, A., Gopalan, N., Gombolay, M., and Hager, G. Guiding multi-step rearrangement tasks with natural language instructions. In Faust, A., Hsu, D., and Neumann, G. (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1486–1501. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/stengel-eskin22a.html.

Stepputtis, S., Campbell, J., Phielipp, M., Lee, S., Baral, C., and Amor, H. B. Language-Conditioned Imitation Learning for Robot Manipulation Tasks, October 2020. URL http://arxiv.org/abs/2010.12083. arXiv:2010.12083 [cs].

Stone, A., Xiao, T., Lu, Y., Gopalakrishnan, K., Lee, K.-H., Vuong, Q., Wohlhart, P., Zitkovich, B., Xia, F., Finn, C., and Hausman, K. Open-world object manipulation using pre-trained vision-language models. *arXiv preprint arXiv: Arxiv-2303.00905*, 2023.

Szot, A., Clegg, A., Undersander, E., Wijmans, E., Zhao, Y., Turner, J., Maestre, N., Mukadam, M., Chaplot, D. S., Maksymets, O., Gokaslan, A., Vondrus, V., Dharur, S., Meier, F., Galuba, W., Chang, A. X., Kira, Z., Koltun, V., Malik, J., Savva, M., and Batra, D. Habitat 2.0: Training home assistants to rearrange their habitat. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 251–266, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/021bbc7ee20b71134d53e20206bd6feb-Abstract.html.

Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. Efficient transformers: A survey. *arXiv preprint arXiv: Arxiv-2009.06732*, 2020.

Team, O. E. L., Stooke, A., Mahajan, A., Barros, C., Deck, C., Bauer, J., Sygnowski, J., Trebacz, M., Jaderberg, M., Mathieu, M., McAleese, N., Bradley-Schmieg, N., Wong, N., Porcel, N., Raileanu, R., Hughes-Fitt, S., Dalibard, V., and Czarnecki, W. M. Open-ended learning leads to generally capable agents. *arXiv preprint arXiv: Arxiv-2107.12808*, 2021.

Thananjeyan, B., Balakrishna, A., Nair, S., Luo, M., Srinivasan, K., Hwang, M., Gonzalez, J. E., Ibarz, J., Finn, C., and Goldberg, K. Recovery RL: safe reinforcement learning with learned recovery zones. *IEEE Robotics Autom. Lett.*, 6(3):4915–4922, 2021. doi: 10.1109/LRA.2021.3070252. URL https://doi.org/10.1109/LRA.2021.3070252.

Toyama, D., Hamel, P., Gergely, A., Comanici, G., Glaese, A., Ahmed, Z., Jackson, T., Mourad, S., and Precup, D. Androidenv: A reinforcement learning platform for android. *arXiv preprint arXiv: Arxiv-2105.13231*, 2021.

Toyer, S., Shah, R., Critch, A., and Russell, S. The MAGICAL benchmark for robust imitation. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/d464b5ac99e74462f321c06ccacc4bff-Abstract.html.

Tsimpoukelli, M., Menick, J., Cabi, S., Eslami, S. M. A., Vinyals, O., and Hill, F. Multimodal few-shot learning with frozen language models. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 200–212, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/

01b7575c38dac42f3cfb7d500438b875-Abstract.html.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *arXiv preprint arXiv: Arxiv-1706.03762*, 2017.

Vinyals, O., Babuschkin, I., Chung, J., Mathieu, M., Jaderberg, M., Czarnecki, W. M., Dudzik, A., Huang, A., Georgiev, P., Powell, R., et al. Alphastar: Mastering the real-time strategy game starcraft ii. *DeepMind blog*, 2, 2019.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv: Arxiv-2305.16291*, 2023.

Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., and Yang, H. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv preprint arXiv: Arxiv-2202.03052*, 2022a.

Wang, T., Roberts, A., Hesslow, D., Scao, T. L., Chung, H. W., Beltagy, I., Launay, J., and Raffel, C. What language model architecture and pretraining objective work best for zero-shot generalization? *Icml*, 2022b. doi: 10.48550/arXiv.2204.05832.

Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., Aggarwal, K., Mohammed, O. K., Singhal, S., Som, S., and Wei, F. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv: Arxiv-2208.10442*, 2022c.

Weihs, L., Deitke, M., Kembhavi, A., and Mottaghi, R. Visual room rearrangement. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 5922–5931. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00586. URL https://openaccess.thecvf.com/content/CVPR2021/html/Weihs_Visual_Room_Rearrangement_CVPR_2021_paper.html.

Weikert, A., Goralczyk, A., Salmela, B., Dansie, B., Barton, C., Valenza, E., Alexandrov, G., Hubert, I., Tysdal, K., Sokolowski, L., Järvinen, M., Pulieso, M., Ebb, M., Vazquez, P., Tuytel, R., Hess, R., Feldlaufer, S., König, S., Platen, S., and Mäter., S. Blender online libraries for textures, 2022. URL https://cloud.blender.org/p/textures/.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv: Arxiv-1910.03771*, 2019.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

Xiao, T., Radosavovic, I., Darrell, T., and Malik, J. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

Xu, M., Shen, Y., Zhang, S., Lu, Y., Zhao, D., Tenenbaum, J. B., and Gan, C. Prompting decision transformer for few-shot policy generalization. *arXiv preprint arXiv: Arxiv-2206.13499*, 2022.

Yang, S., Nachum, O., Du, Y., Wei, J., Abbeel, P., and Schuurmans, D. Foundation models for decision making: Problems, methods, and opportunities. *arXiv preprint arXiv: Arxiv-2303.04129*, 2023.

Yang, Z., Fang, Y., Zhu, C., Pryzant, R., Chen, D., Shi, Y., Xu, Y., Qian, Y., Gao, M., Chen, Y.-L., Lu, L., Xie, Y., Gmyr, R., Codella, N., Kanda, N., Xiao, B., Yuan, L., Yoshioka, T., Zeng, M., and Huang, X. i-code: An integrative and composable multimodal learning framework. *arXiv preprint arXiv: Arxiv-2205.01818*, 2022.

Yoon, Y., DeSouza, G., and Kak, A. Real-time tracking and pose estimation for industrial objects using geometric features. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 3, pp. 3473–3478 vol.3, 2003. doi: 10.1109/ROBOT.2003.1242127.

Yu, T., Quillen, D., He, Z., Julian, R., Narayan, A., Shively, H., Bellathur, A., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. *arXiv preprint arXiv: Arxiv-1910.10897*, 2019.

Yu, T., Xiao, T., Stone, A., Tompson, J., Brohan, A., Wang, S., Singh, J., Tan, C., M, D., Peralta, J., Ichter, B., Hausman, K., and Xia, F. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv: Arxiv-2302.11550*, 2023.

Yuan, L., Chen, D., Chen, Y.-L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., Liu, C., Liu, M., Liu, Z., Lu, Y., Shi, Y., Wang, L., Wang, J., Xiao, B., Xiao, Z., Yang, J., Zeng, M., Zhou, L., and Zhang, P. Florence: A new foundation model for computer vision. *arXiv preprint arXiv: Arxiv-2111.11432*, 2021.

Zellers, R., Lu, X., Hessel, J., Yu, Y., Park, J. S., Cao, J., Farhadi, A., and Choi, Y. Merlot: Multimodal neural script knowledge models. *arXiv preprint arXiv: Arxiv-2106.02636*, 2021.

Zellers, R., Lu, J., Lu, X., Yu, Y., Zhao, Y., Salehi, M., Kusupati, A., Hessel, J., Farhadi, A., and Choi, Y. Merlot reserve: Neural script knowledge through vision and language and sound. *CVPR*, 2022.

Zeng, A., Florence, P., Tompson, J., Welker, S., Chien, J., Attarian, M., Armstrong, T., Krasin, I., Duong, D., Wahid, A., Sindhwani, V., and Lee, J. Transporter networks: Rearranging the visual world for robotic manipulation. *arXiv preprint arXiv: Arxiv-2010.14406*, 2020.

Zeng, A., Wong, A., Welker, S., Choromanski, K., Tombari, F., Purohit, A., Ryoo, M., Sindhwani, V., Lee, J., Vanhoucke, V., and Florence, P. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv: Arxiv-2204.00598*, 2022.

Zhang, Z. and Weihs, L. When learning is out of reach, reset: Generalization in autonomous visuomotor reinforcement learning. *arXiv preprint arXiv: Arxiv-2303.17600*, 2023.

Zhao, M., Liu, F., Lee, K., and Abbeel, P. Towards more generalizable one-shot visual imitation learning. In *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pp. 2434–2444. IEEE, 2022. doi: 10.1109/ICRA46639.2022.9812450. URL https://doi.org/10.1109/ICRA46639.2022.9812450.

Zheng, Q., Zhang, A., and Grover, A. Online decision transformer. *arXiv preprint arXiv: Arxiv-2202.05607*, 2022.

Zhu, Y., Wong, J., Mandlekar, A., and Martín-Martín, R. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv: Arxiv-2009.12293*, 2020.

Zhu, Y., Joshi, A., Stone, P., and Zhu, Y. Viola: Imitation learning for vision-based manipulation with object proposal priors. *arXiv preprint arXiv: Arxiv-2210.11339*, 2022.

A. Simulator Details

We build our VIMA-BENCH simulation suite upon the Ravens physics simulator (Zeng et al., 2020; Shridhar et al., 2021). Specifically, it is supported by PyBullet (Coumans & Bai, 2016–2021) with a Universal Robot UR5 arm. The size of the tabletop workspace is 0.5×1m. Our benchmark contains extensible sets of 3D objects and textures. Instantiated from an object-texture combination, all object instances can be rendered as RGB images appeared in multimodal prompts. Figure A.1 displays all 3D objects. Figure A.2 displays all textures.

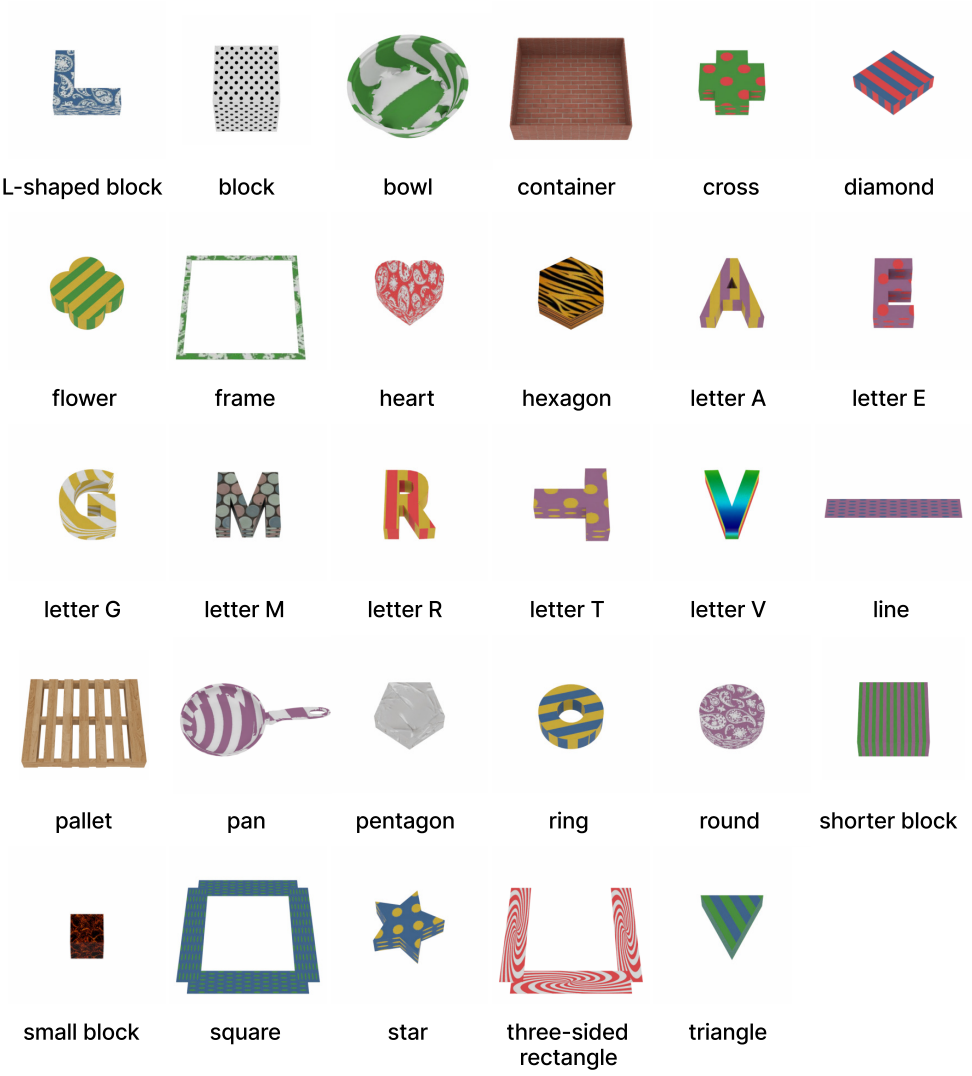

Figure A.1: **Object Gallery in VIMA-BENCH** textured with random textures. Bowl and pan are from Google Scanned Objects (Downs et al., 2022), while others are from Ravens (Zeng et al., 2020).

The observation space of VIMA-BENCH includes RGB images from both frontal and top-down views. It also includes a one-hot vector $\in \{0, 1\}^2$ to indicate type of the end-effector $\in \{\text{suction cup, spatula}\}$. While a suction cup is equipped in most manipulation tasks, a spatula is used in particular for visual constraint tasks, where an agent is asked to "wipe" objects. VIMA-BENCH inherits the same action space from Zeng et al. (2020) and Shridhar et al. (2021), which consists of primitive actions of "pick and place" for tasks with a suction cup as the end effector, or "push" for tasks with a spatula. Both primitive actions contain two poses $\in \mathbf{SE}(2)$ specifying target poses of the end effector. For the "pick and place" primitive, they represent the pick pose and the place pose. For the "push" primitive, they represent the push starting pose and push

ending pose.

Similar to prior work (Zeng et al., 2020; Shridhar et al., 2021), VIMA-BENCH provides scripted oracles to generate successful demonstrations for all tasks. We leverage them to construct an offline imitation dataset for behavioral cloning. Given a prompt, these programmed bots can access privileged information, such as the correct object to pick and target location to place.

Figure A.2: **Texture Gallery in VIMA-BENCH**. The first row of image-based textures is from Blender Cloud Libraries (Weikert et al., 2022), while others are hard-coded.

B. Task Suite

We develop 17 task templates that belong to 6 diverse categories. Thousands of individual task instances and their corresponding multimodal prompts can be procedurally generated from these task templates. We use PyBullet (Coumans & Bai, 2016–2021) as our backend and the default renderer to produce the RGB frames for training data and interactive test environments. For demonstration purpose, we apply the NVISII (Morrical et al., 2020) ray tracing to enhance the visual quality. We elaborate on each task in the following subsections.

B.1. Simple Object Manipulation

This task category asks agents to follow basic instructions specified by multimodal prompts.

Task 01: Pick the specified object(s) and place it (them) into the specified container.

- **Prompt:** `Put the` $\{\texttt{object}\}_1$ `into the` $\{\texttt{object}\}_2$`.`
- **Description:** The image placeholder $\{\texttt{object}\}_1$ is the object to be picked and the $\{\texttt{object}\}_2$ is the container object. The agent requires to recognize the objects with the correct color-shape combinations. To extend the difficulties, it supports more than one object to be picked or placed. For example, the prompt "`Put the` $\{\texttt{object}\}_1$ `and` $\{\texttt{object}\}_2$ `into the` $\{\texttt{object}\}_3$" asks to pick two different objects and place into a target container. We uniformly sample different color-shape combos for objects to be picked and containers.
- **Success Criteria:** All specified object(s) to pick are within the bounds of the container object(s), with specified shapes and textures provided in the prompt.
- **Oracle Trajectory:** Shown in Fig. A.3 with its multimodal prompt.

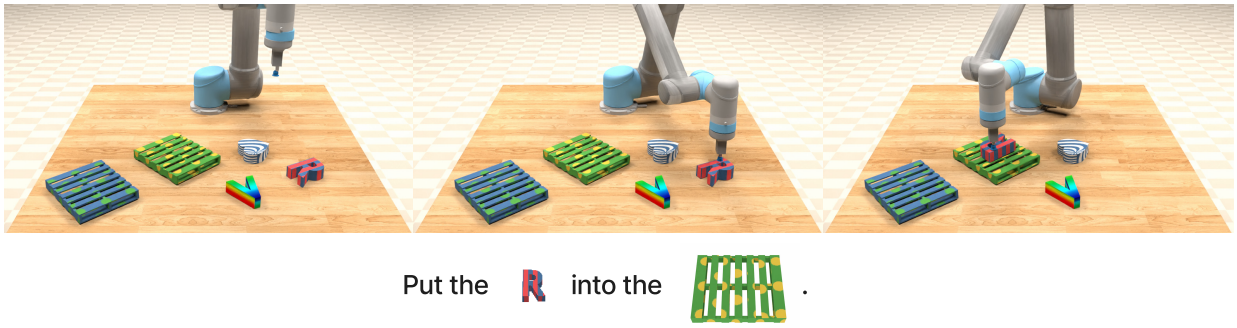

Figure A.3: Simple Object Manipulation: Task 01

Task 02: In the workspace, put the objects with a specified texture shown in the scene image in the prompt into container object(s) with a specified color. This task requires the agent to find the correct object to manipulate by grounding the textural attributes from both natural language descriptions and the visual scene images.

- **Prompt:** `Put the` $\{\texttt{texture}\}_1$ `object in` $\{\texttt{scene}\}$ `into the` $\{\texttt{texture}\}_2$ `object.`
- **Description:** The text placeholder $\{\texttt{texture}\}_1$ and $\{\texttt{texture}\}_2$ are sampled textures for objects to be picked and the container objects, respectively. The number of dragged objects with the same texture can be varied. $\{\texttt{scene}\}$ is the workspace-like image placeholder. There is a designated number of distractors with different textures (and potentially different shapes) in the scene. For each distractor in the workspace, it has 50% chance to be either dragged or container distractor object with different textures from those specified in the prompt.
- **Success Criteria:** All objects in the workspace with $\{\texttt{texture}\}_1$ are within the bounds of the container object with $\{\texttt{texture}\}_2$.

- **Oracle Trajectory:** Shown in Fig. A.4 with its multimodal prompt.

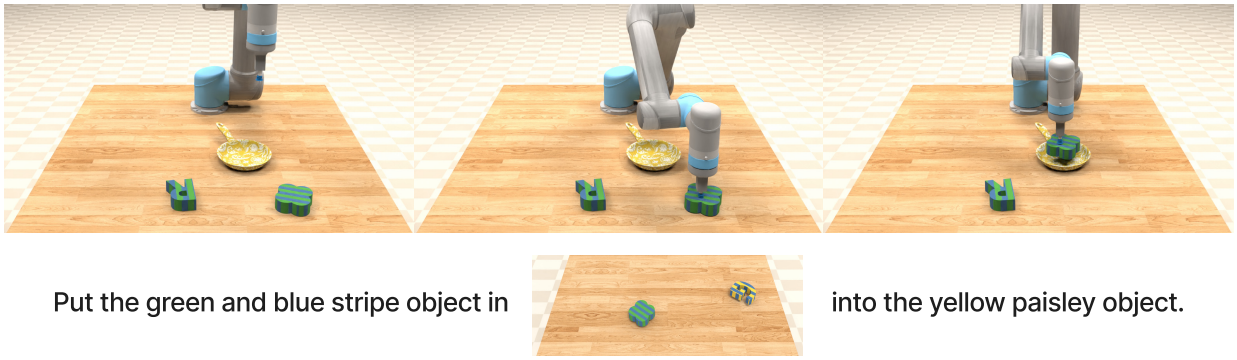

Figure A.4: Simple Object Manipulation: Task 02

Task 03: Rotate objects clockwise by certain degrees along z-axis. Only rotationally asymmetric objects are considered in this task.

- **Prompt:** `Rotate the` $\{$`object`$\}_1$ $\{$`angles`$\}$ `degrees.`
- **Description:** The agent is required to rotate all objects in the workspace specified by the image placeholder $\{$`object`$\}_1$. There are also objects with different color-shape combinations in the workspace as distractors. $\{$`angles`$\}$ is the sampled degree that needs to be rotated. A target angle is sampled from $30°$, $60°$, $90°$, $120°$, and $150°$.
- **Success Criteria:** The position of the specified object matches its original position, and the orientation matches the orientation after rotating specific angles.
- **Oracle Trajectory:** Shown in Fig. A.5 with its multimodal prompt.

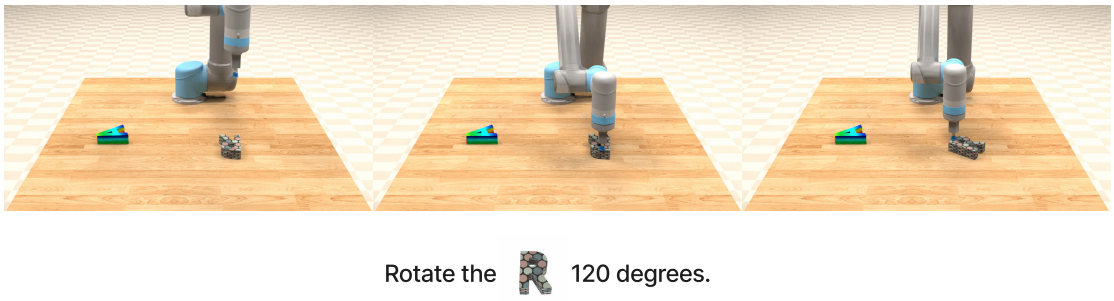

Figure A.5: Simple Object Manipulation: Task 03

B.2. Visual Goal Reaching

This task category requires agents to manipulate objects in the workspace to reach goal states represented as images shown in prompts.

Task 04: Rearrange target objects in the workspace to match goal configuration shown in prompts. Note that to achieve the goal configuration, distractors may need to be moved away first.

- **Prompt:** `Rearrange to this {scene}.`
- **Description:** Objects in the scene placeholder `{scene}` are target objects to be manipulated and rearranged. In the workspace, the same target objects are spawned randomly, potentially with distractors randomly spawned as well. With a pre-defined distractor conflict rate, the position of each distractor has this probability to occupy the position of any target object such that the rearrangement can only succeed if moving away that distractor first.
- **Success Criteria:** The configuration of target objects in the workspace matches that specified in the prompt.
- **Oracle Trajectory:** Shown in Fig. A.6 with its multimodal prompt.

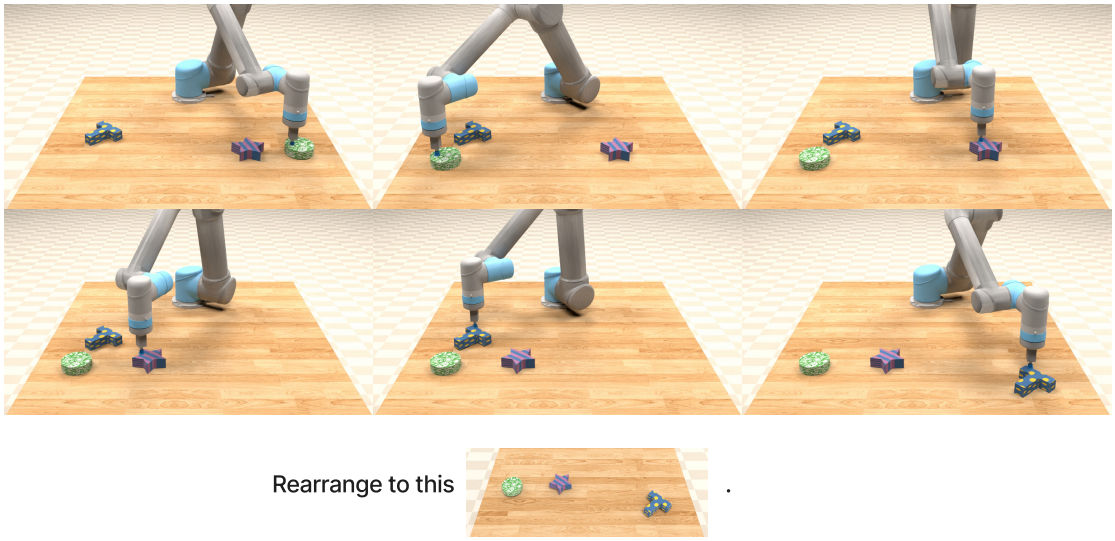

Figure A.6: Visual Goal Reaching: Task 04

Task 05: Extend the task *04* by requiring the agent to restore rearranged objects to the initial setup after the "rearranging" phase.

- **Prompt:** `Rearrange objects to this setup {scene} and then restore.`
- **Description:** Same as the task *04*, except introducing the instruction "restore".
- **Success Criteria:** Meet the success criteria of the task *04*, and then within the allowed max steps restore all target objects to their initial configurations.
- **Oracle Trajectory:** Shown in Fig. A.7 with its multimodal prompt.

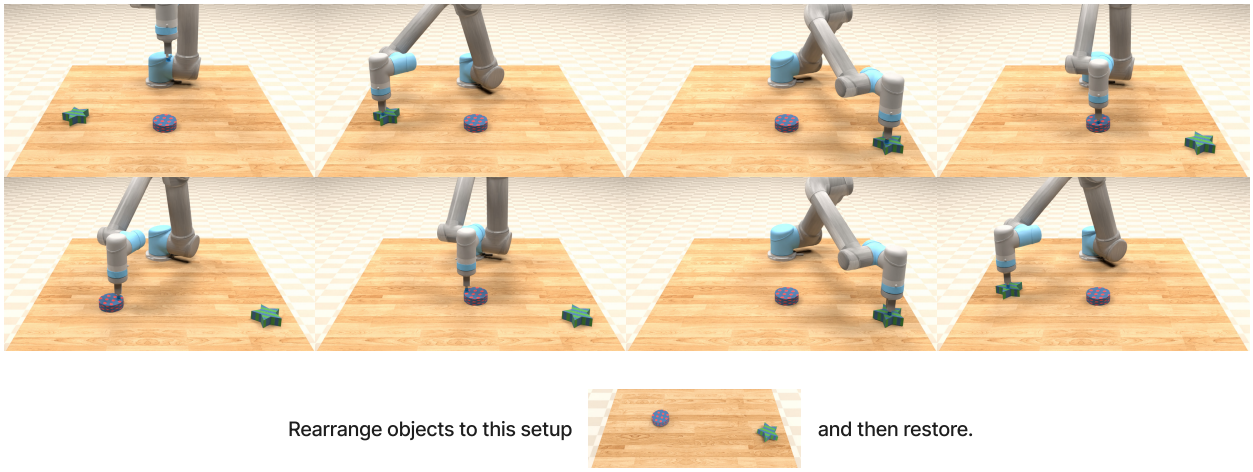

Figure A.7: Visual Goal Reaching: Task 05

B.3. Novel Concept Grounding

This task category requires agents to ground new concepts of adjectives, nouns, or verbs via visual perception and language understanding. Similar task design can be found in prior work (Hill et al., 2021). Completing these tasks are challenging, because the model should a) first understand prompts with interleaved texts, images, and even video frames; b) quickly internalize new concepts that are different across task instances, which even tests the ability to meta-learn; and c) do complicated reasoning such as comparing between "taller" vs "less taller" vs "shorter" and then ground this reasoning into the robot action space.

Prompts consist of two parts: a definition part followed by an instruction part. In the definition part, novel concepts are defined by multimodal illustrations with multiple support examples. In the instruction part, agents are asked to achieve the goal by properly applying concepts from the definition part. The assignment of dummy object names is varied and independent for each task instance such that tasks can only be solved if the agent applies the reasoning correctly. This ability is also referred to as *fast-mapping* (Heibeck & Markman, 1987).

Task 06: Ground comparative adjectives by comparing the size or the textural saturation of objects and manipulating the correct object(s) instructed in the prompt.

- **Prompt:** $\{\texttt{demo\_object}\}_1$ `is` $\{\texttt{novel\_adj}\}$ `than` $\{\texttt{demo\_object}\}_2$`.` `Put the` $\{\texttt{adv}\}$ $\{\texttt{novel\_adj}\}$ $\{\texttt{object}\}_1$ `into the` $\{\texttt{object}\}_2$`.`
- **Description:** The sampled adjective $\{\texttt{novel\_adj}\}$ is a dummy adjective placeholder for agent to ground. By default, the novel adjective set is $\{\texttt{daxer, blicker, modier, kobar}\}$. The real meaning can be related to size (smaller/larger) or textural saturation (lighter/darker texture). The image placeholders $\{\texttt{demo\_object}\}_1$ and $\{\texttt{demo\_object}\}_2$ illustrate how the novel adjective is defined. For example, if the real comparison is "taller", then the sampled object in $\{\texttt{demo\_object}\}_1$ is taller than $\{\texttt{demo\_object}\}_2$. The choices of the novel adjective and the real meaning are independently sampled for different task instances. For the instruction part, this task is similar to task *01*, where the agent is required to pick the specified object(s) with the novel adjective attribute and then place it into the specified container object. To avoid revealing the correct object to manipulate, we use a neutral texture for objects appeared in the instruction part.
- **Success Criteria:** All target objects with the specified adjective attribute are within the bounds of the specified container object.
- **Oracle Trajectory:** Shown in Fig. A.8 with its multimodal prompt.

Figure A.8: Novel Concept Grounding: Task 06

Task 07: Orthogonal to task *06* by requiring to learn mappings of novel nouns.

- **Prompt:** `This is a {novel_name}_1 {object}_1. This is a {novel_name}_2 {object}_2. Put {novel_name}_1 into a {novel_name}_2.`
- **Description:** Novel noun words are defined with the text placeholders $\{\texttt{novel\_name}\}_1$ and $\{\texttt{novel\_name}\}_2$, following their image placeholders $\{\texttt{object}\}_1$ and $\{\texttt{object}\}_2$, for the target object and container object, respectively. Novel nouns are sampled from {dax, blicket, wug, zup}. In the instruction part, objects are expressed as novel nouns defined in the previous definition part. Distractors are defined the same as task *01*.
- **Success Criteria:** All target object(s) are within the bounds of the container object(s).
- **Oracle Trajectory:** Shown in Fig. A.9 with its multimodal prompt.

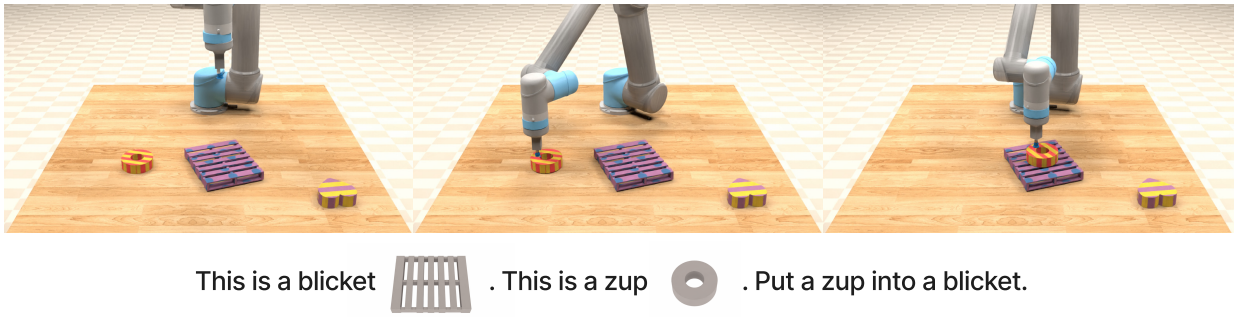

Figure A.9: Novel Concept Grounding: Task 07

Task 08: Combination of tasks *06* and *07*.

- **Prompt:** `This is a {novel_name}_1 {object}_1. This is a {novel_name}_2 {object}_2. {demo_object}_1 is {adj} than {demo_object}_2. Put the {adv} {novel_adj} {novel_name}_1 into the {novel_name}_2.`
- **Description:** See task description for task *06* and task *07*.
- **Success Criteria:** Similar as tasks *06* and *07*.
- **Oracle Trajectory:** Shown in Fig. A.10 with its multimodal prompt.

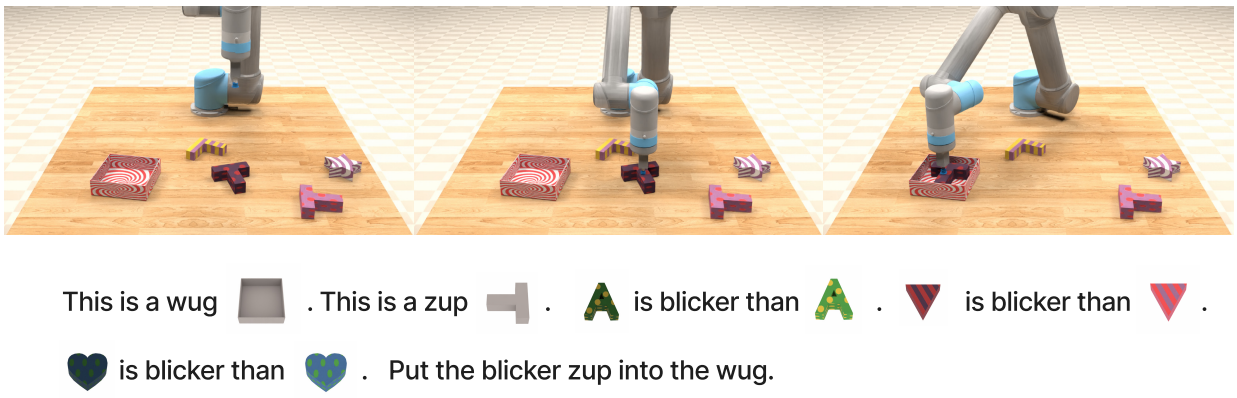

Figure A.10: Novel Concept Grounding: Task 08

Task 09: A novel verb "twist" is defined as rotating a specific angle illustrated by several examples. This task is similar to task *03*, but it requires the agent to infer what is the exact angle to rotate from the prompt and to ground novel verbs that are semantically similar but different in exact definitions.

- **Prompt:** `"Twist" is defined as rotating object a specific angle. For examples: From {before_twist}$_i$ to {after_twist}$_i$. Now twist all {texture} objects.`
- **Description:** Both `{before_twist}`$_i$ and `{after_twist}`$_i$ are scene placeholders where `{before_twist}`$_i$ shows a randomly sampled object before "twisting" and `{after_twist}`$_i$ shows the same object pose after "twisting". All examples illustrate the same sampled angle to rotate. In the workspace, the target objects have the texture specified by `{texture}` and randomly sampled shapes.
- **Success Criteria:** Same as the task *03*.
- **Oracle Trajectory:** Shown in Fig. A.11 with its multimodal prompt.

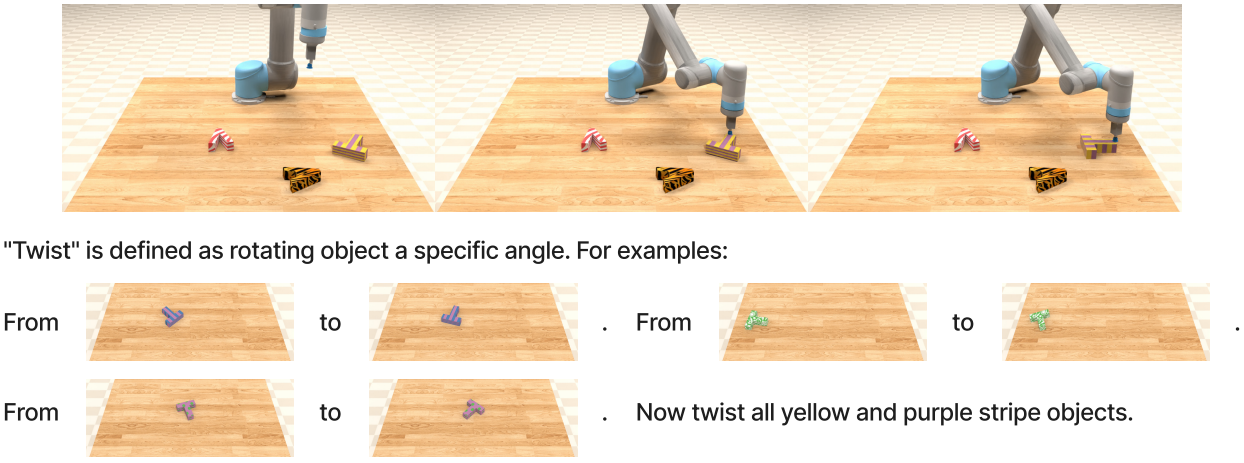

Figure A.11: Novel Concept Grounding: Task 09

B.4. One-Shot Video Imitation

This task category requires agents to imitate motions demonstrated through videos shown in prompts. We follow prior works (Finn et al., 2017; Dasari & Gupta, 2020; Duan et al., 2017) to formulate the problem by giving one video demonstration (represented as key frames in prompts), then test the learned imitator's ability to produce target trajectories. This setup is challenging because a) only one demonstration is available to the agent; b) the model needs to understand video frames interleaved with textual instructions; and c) missing correspondences between demonstrations and target trajectories since demonstrations only show partial key frames.

Task 10: Follow motions for specific objects.

- **Prompt:** `Follow this motion for {object}:` $\{\texttt{frame}\}_1 \ldots \{\texttt{frame}\}_i \ldots \{\texttt{frame}\}_n$.
- **Description:** Image placeholder $\{\texttt{object}\}$ is the target object to be manipulated and $\{\{\texttt{frame}\}_i\}$ is set of workspace-like scene placeholders to represent a video trajectory, where n is the trajectory length. There is an object spawned at the center in both the workspace and the prompt video but with different textures as a distractor. The initial position of the target object matches that in $\{\texttt{frame}\}_1$.
- **Success Criteria:** In each step, the pose of the target object matches the pose in the corresponding video frame. Incorrect manipulation sequences are considered as failures.
- **Oracle Trajectory:** Shown in Fig. A.12 with its multimodal prompt.

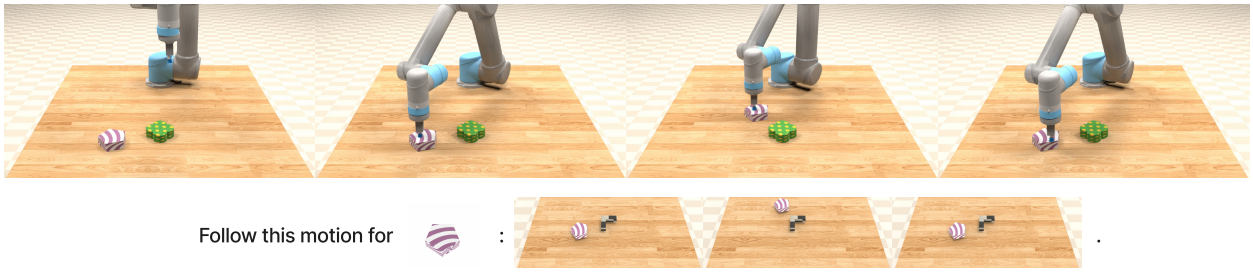

Figure A.12: One-shot video imitation: Task 10

Task 11: Stack objects with the order illustrated in the prompt video.

- **Prompt:** `Stack objects in this order` $\{\texttt{frame}\}_1 \ldots \{\texttt{frame}\}_i \ldots \{\texttt{frame}\}_n$.
- **Description:** There are multiple objects with the same shape but different textures spawned in the workspace without any stacking initially. Distractor objects with different shapes are spawned in the workspace but not in the prompt video. At each step of the prompt video, one object is stacked over another or put at an empty position.
- **Success Criteria:** Similar as task 10.
- **Oracle Trajectory:** Shown in Fig. A.13 with its multimodal prompt.

B.5. Visual Constraint Satisfaction

This task category requires agents to wipe a specific number of objects in the workspace to a goal region while also satisfy the given visual constraint.

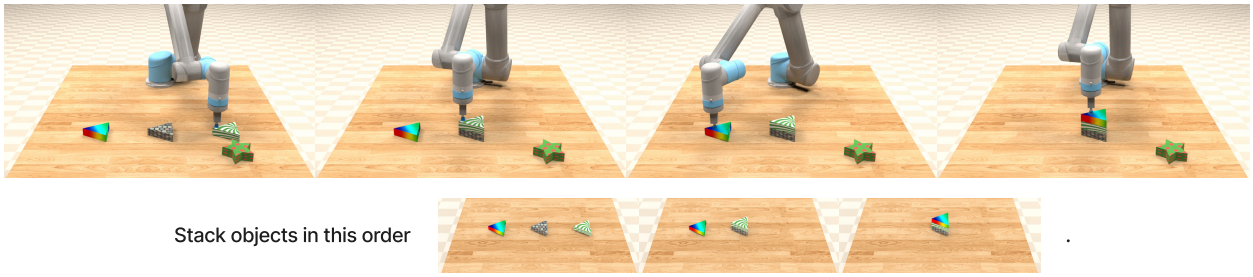

Figure A.13: One-shot video imitation: Task 11

Task 12: Sweep the designated number of objects into a specified region without exceeding the boundary.

- **Prompt:** `Sweep {quantifier} {object} into {bounds} without exceeding {constraint}.`
- **Description:** `{object}` is the image placeholder of the target object to be swept spawned with a random amount in the workspace. Distractors have the same amount, same shape, but different color from target objects. `{quantifier}` is the text placeholder to determine the target quantity of objects to be wiped, sampled from `any`, `one`, `two`, `three`, and `all`. `{bounds}` is the image placeholder for a three-sided rectangle as the goal region. `{constraint}` is the constraint line.
- **Success Criteria:** The exact number of target objects to be swept are all inside the specified region. Potential failure cases include 1) any distractor being wiped into the region, 2) target object exceeding the constraint, or 3) incorrect number of target objects being swept into the goal region.
- **Oracle Trajectory:** Shown in Fig. A.14 with its multimodal prompt.

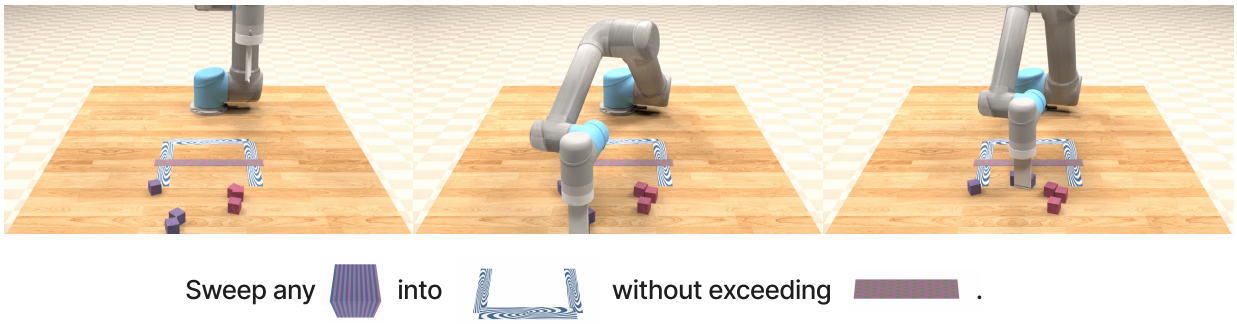

Figure A.14: Visual Constraint Satisfaction: Task 12

Task 13: Sweep the designated number of objects into a specified region without touching the constraint.

- **Prompt:** `Sweep {quantifier} {object} into {bounds} without touching {constraint}.`
- **Description:** Similar as task *12* but requiring a different way to satisfy the constraint. The agent has to learn to avoid contacting the constraint line in this case.
- **Success Criteria:** Similar as task *12* except that the constraint is to not touch the red line.
- **Oracle Trajectory:** Shown in Fig. A.15 with its multimodal prompt.

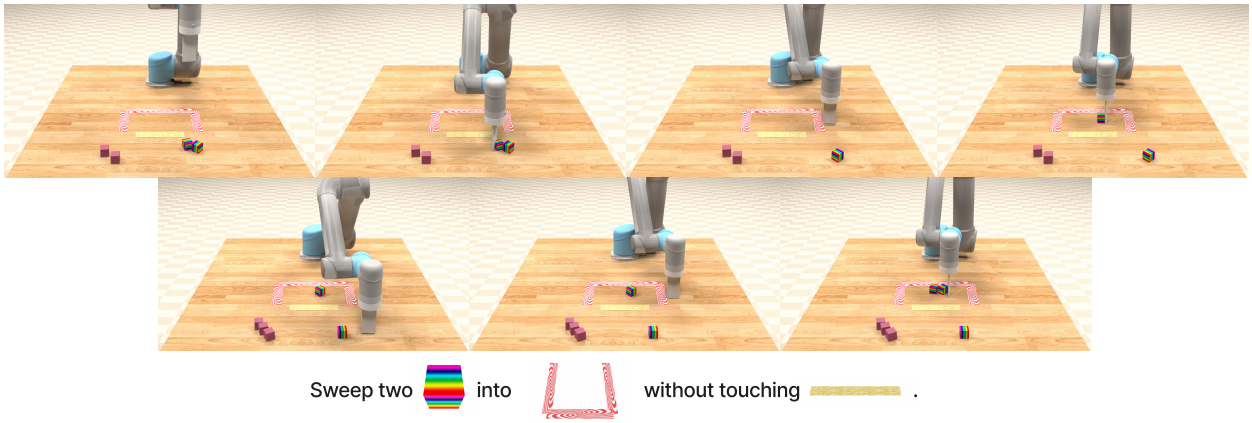

Figure A.15: Visual Constraint Satisfaction: Task 13

B.6. Visual Reasoning

This task category requires agents to make decisions by reasoning over or memorizing information conveyed through multimodal prompts.

Task 14: By reasoning the "same texture", the agent is required to pick all objects in the workspace with the same texture as the container objects specified in the prompt and place them into it.

- **Prompt:** `Put all objects with the same texture as {object} into it.`
- **Description:** `{object}` is the sampled goal container object. In the workspace, there are objects with the same texture as the container but potentially different shapes. Distractors with different textures are spawned.
- **Success Criteria:** All objects with the same texture as the goal container are within the bounds of the container.
- **Oracle Trajectory:** Shown in Fig. A.16 with its multimodal prompt.

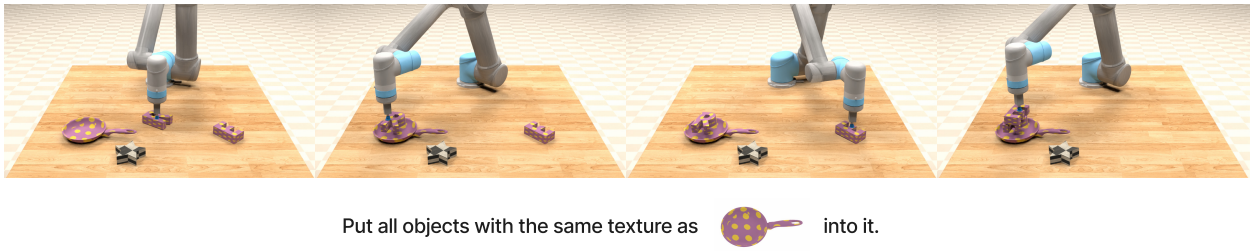

Figure A.16: Visual Reasoning: Task 14

Task 15: By reasoning the "same shape", the agent is required to pick all objects in the workspace with the same top-down profile as the goal container specified in the prompt and place them into it. For example, blocks and boxes have the same rectangular profile.

- **Prompt:** `Put all objects with the same profile as {object} into it.`
- **Description:** Similar to the task *14* except the objects to be picked and placed have the same shape. There are three different shapes: *rectangular-like* (e.g. block and pallet), *circle-like* (e.g. ring and bowl), and *undetermined* for the rest.

- **Success Criteria:** All objects with the same shape as the container are within the container.
- **Oracle Trajectory:** Shown in Fig. A.17 with its multimodal prompt.

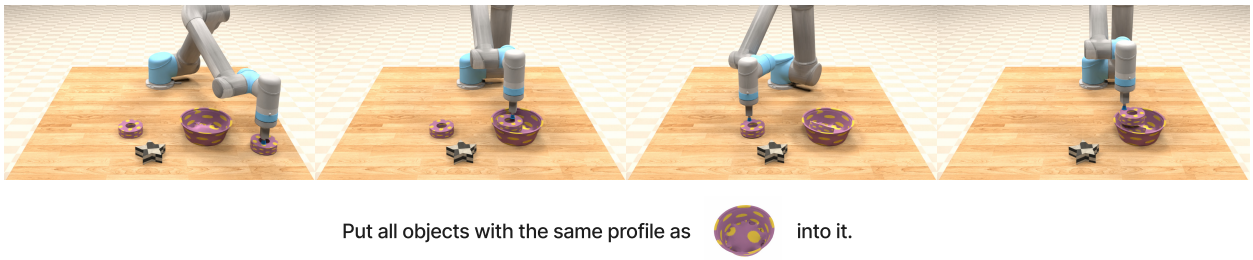

Figure A.17: Visual Reasoning: Task 15

Task 16: Put the target object into the container, and then put one of its old neighbors into the same container.

- **Prompt:** `First put` $\{\texttt{object}\}_1$ `into` $\{\texttt{object}\}_2$ `then put the object that was previously at its` $\{\texttt{direction}\}$ `into the same` $\{\texttt{object}\}_2$.
- **Description:** Objects in image placeholders $\{\texttt{object}\}_1$ and $\{\texttt{object}\}_2$ are the target object to be picked and the container, respectively. We then ask the agent to put one of old neighbors of the previous target object into the same container. The old neighboring object is specified through cardinal directions $\{\texttt{north, south, west, east}\}$.
- **Success Criteria:** The target object and the correct neighboring object are inside the container.
- **Oracle Trajectory:** Shown in Fig. A.18 with its multimodal prompt.

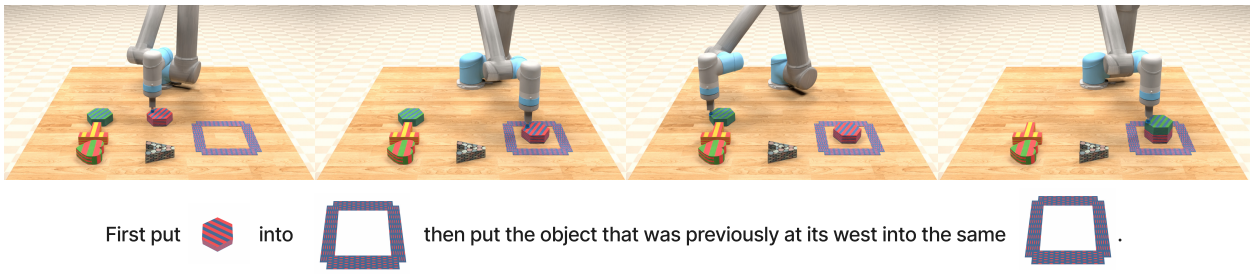

Figure A.18: Visual Reasoning: Task 16

Task 17: Pick and place the target object specified in the prompt into different containers in order then restore to the initial container.

- **Prompt:** `Put` $\{\texttt{object}\}_1$ `into` $\{\texttt{object}\}_2$ `. Finally restore it into its original container.`
- **Description:** The object in the image placeholder $\{\texttt{object}\}_1$ is the target object to be manipulated across the task. There are more than one target containers (e.g. "`Put` $\{\texttt{object}\}_1$ `into` $\{\texttt{object}\}_2$ `then` $\{\texttt{object}\}_3$`. Finally restore it into its original container`" for two target containers to be placed in order). The rest of spawned containers naturally becomes distractors.

- **Success Criteria:** The target object is first put into multiple containers following the specific order. Finally it should be restored into its original container.
- **Oracle Trajectory:** Shown in Fig.A.19 with its multimodal prompt.

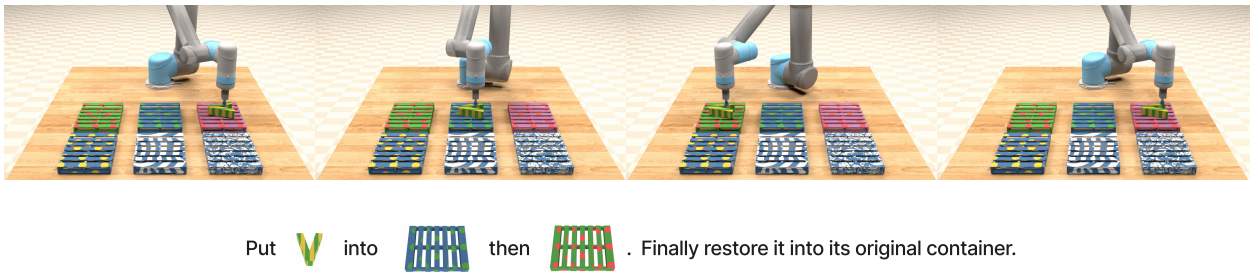

Figure A.19: Visual Reasoning: Task 17

C. Model Architecture

In this section, we provide comprehensive details about VIMA model architecture as well as other adapted baseline methods. We implement all models in PyTorch (Paszke et al., 2019) and adapt Transformer-related implementation from Wolf et al. (2019).

C.1. Summary of Different Methods

We summarize differences between VIMA and other baseline variants in Table 1. In the column "Prompt Conditioning", an alternative to cross-attention is to first concatenate prompt and interaction into a big sequence, then repetitively apply transformer decoders to predict actions. It is referred to as "Direct modeling". The relative computation cost is quadratically proportional to the number of observation tokens.

Table 1: Comparison of different methods.

	Visual Tokenizer	**Prompt Conditioning**	**Number of Observation Tokens per Step**
Ours	Object tokens consisting of cropped images and bounding boxes	Cross-attention	Equal to number of objects, typically 3 to 8
VIMA-Gato (Reed et al., 2022)	Image patch tokens encoded by a ViT	Direct modeling	Equal to number of image patches, 16
VIMA-Flamingo (Alayrac et al., 2022)	Image patch tokens encoded by a ViT, further downsampled by a Perceiver module	Cross-attention	Equal to number of learned query vectors, 4
VIMA-GPT (Brown et al., 2020)	Single image token encoded by a ViT	Direct modeling	Single visual feature, 1

C.2. VIMA Architecture

C.2.1. Multimodal Prompt Tokenization

As introduced in Section 4, there are 3 types of input formats in multimodal prompts, namely (1) **text inputs**, (2) **images of full scenes**, and (3) **images of single objects**.

For **text inputs**, we follow the standard pipeline in NLP to first tokenize raw language to discrete indices through pre-trained `t5-base` tokenizer. We then obtain corresponding word tokens from the embedding look-up of the pre-trained `t5-base` model. For **images of full scenes**, we first parse the scene through a fine-tuned Mask R-CNN detection model (He et al., 2017; Wu et al., 2019) to extract individual objects. Each object representation contains a bounding box and a cropped image. The bounding box is in the format of $\big[x_{\text{center}}, y_{\text{center}}, \text{height}, \text{width}\big]$. We normalize it to be within $[0, 1]$ by dividing each dimension with corresponding upper-bound value. We then pass it through a bounding box encoder MLP and obtain a feature vector. To process the cropped image, we first pad non-square image to a square by padding along the shorter dimension. We then resize it to a pre-configured size and pass it through a ViT (trained from scratch) to obtain the image feature. Finally, an object token is obtained by concatenating the bounding box feature and the image feature and mapping to the embedding dimension. For **images of single objects**, we obtain tokens in the same way except with a dummy bounding box. Detailed model hyperparameters about tokenization are listed in Table 2.

After obtaining a sequence of prompt tokens, we follow Tsimpoukelli et al. (2021) to pass it through a pre-trained `t5-base` encoder to obtain encoded prompt. Note that we add adapter MLP between object tokens and the T5 encoder. To prevent catastrophic forgetting, VIMA only fine-tunes the last two layers of the language encoder with layer-wise learning rate decay (He et al., 2021) but freezes all other layers. We adopt learned absolute positional embedding. Model hyperparameters are listed in Table 2 as well.

C.2.2. Observation Encoding

Since all RGB observations are images of full scenes, we follow the same procedure discussed above to obtain flattened object tokens. Because we provide RGBs from two views (frontal and top-down), we order object tokens by following the

Table 2: Model hyperparameters for multimodal prompt tokenization.

Hyperparameter	Value
Text Tokenization	
Tokenizer	`t5-base` tokenizer
Embedding Dimension	768
Image Tokenization	
ViT Input Image Size	32×32
ViT Patch Size	16
ViT Width	768
ViT Layer	4
ViT Number of Heads	24
Bounding Box MLP	
Hidden Dimension	768
Hidden Depth	2
Prompt Encoding	
Pre-Trained LM	`t5-base`
Unfrozen Last N Layers	2
Positional Embedding	Absolute
Token Adapter MLP Depth	2

order of $\big[$frontal, top-down$\big]$. We one-hot encode the state of the end effector. We then concatenate object tokens with the end-effector state and transform to observation tokens. We adopt learned absolute positional embedding. Detailed model hyperparameters about observation encoding is provided in Table 3.

Table 3: Model hyperparameters for observation encoding.

Hyperparameter	Value
Observation Token Dimension	768
End Effector Embedding Dimension	2
Positional Embedding	Absolute

C.2.3. Action Encoding

Since our model is conditioned on observation-action interleaved history, we also tokenize past actions. We follow common practice in Chen et al. (2021); Zheng et al. (2022) to encode past actions with a two-layer MLP. It has a hidden dimension of 256. We then map outputs to token dimension and obtain action tokens.

C.2.4. Sequence Modeling

The robot controller in VIMA is a causal decoder that autoregressively predicts actions. To condition the decoder on prompt tokens, we perform cross-attention between history tokens and prompt tokens (Figure 3). Concretely, we pass history tokens as the query sequence and prompt tokens as the key-value sequence into cross-attention blocks. The output prompt-aware trajectory tokens then go through causal self-attention blocks. We alternate cross-attention and self-attention L times. This procedure is technically described in Pseudocode 1.

```
def xattn_sequence_modeling(
    prompt_tokens,      # the [L, d] prompt tokens (L=prompt length)
    obs_tokens,         # the [T, d] obs tokens (T=time step)
    act_tokens,         # the [T-1, d] action tokens
    traj_pos_embd,      # learned positional embedding for trajectory
    prompt_pos_embd,    # learned positional embedding for prompt
):
    # interleave obs and action tokens
    traj_tokens = interleave(obs_tokens, act_tokens)  # [2T-1, d]
    # add positional embedding to trajectory tokens
    x = traj_tokens + traj_pos_embd
    # add positional embedding to prompt tokens
    prompt_tokens = prompt_tokens + prompt_pos_embd

    # apply xattn and causal self-attn
    for i in range(num_layers):
        # cross-attention
        x = x + attn_i(q=x, kv=prompt_tokens)
        # feed forward
        x = x + ffw_xattn_i(x)
        # self-attention
        x = x + causal_attn_i(q=x, kv=x)
        # feed forward
        x = x + ffw_i(x)

    # the last token is the predicted action token
    predicted_act_token = x[-1]
    return predicted_act_token
```

Pseudocode 1: Cross-attention operation that conditions the trajectory history on prompt. We repetitively alternate cross-attention and self-attention to model the trajectory given a specific task.

C.2.5. Action Decoding

After obtaining the predicted action token, we map it to the action space $\mathcal{A}$ and obtain the predicted action. This is achieved though a group of action heads. Since the action space consists of two $\mathbf{SE}(2)$ poses, for each pose we use six independent heads to decode discrete actions (two for xy coordinate and four for rotation represented in quaternion). These discrete actions are then integrated and mapped to continuous actions through affine transformation. The two poses are modeled independently. Early ablations show that this independent modeling is equally good as alternative techniques, such as autoregressive decoding (Vinyals et al., 2019; OpenAI et al., 2019). Detailed model hyperparameters are listed in Table 4.

Table 4: Model hyperparameters for action decoders.

Hyperparameter	Value
Hidden Dimension	512
Hidden Depth	2
Activation	ReLU
X-Axis Discrete Bins	50
Y-Axis Discrete Bins	100
Rotation Discrete Bins	50

C.3. Baselines Architectures

In this section, we elaborate model architectures for adapted baseline methods. Some components such as the action decoder are same across all models. Therefore, we only discuss unique model components.

C.3.1. VIMA-GATO

Gato (Reed et al., 2022) introduces a decoder-only model that solves tasks from multiple domains including robotics, video game, image captioning, language modeling, etc. Different tasks are specified by supplying the model with an initial sequence of corresponding tokens. For example, in tasks involving decision making, these tokens include observation and action tokens. For fair comparison, we provide the same conditioning as VIMA, i.e., our multimodal tokenized prompts. This adapted baseline variant is referred to as "**VIMA-Gato**". Similar to our method, VIMA-Gato also predicts actions in an autoregressive manner. VIMA-Gato and our method share the same training philosophy to only optimize the causal behavior cloning objective. However, unlike our method that adopts an object-centric representation to treat individual objects as observation tokens, VIMA-Gato divides input images into patches and encodes them by a ViT (Dosovitskiy et al., 2020) to produce observation tokens. Furthermore, VIMA-Gato relies on causal self-attention to model entire trajectory sequences starting with prompt tokens. Hyperparameters of VIMA-Gato's ViT is listed in Table 5. The transformer-decoder style sequence modeling is technically illustrated in Pseudocode 2.

Table 5: Model hyperparameters for ViT used in baseline methods.

Hyperparameter	Value
Image Size	64×128
Patch Size	32
ViT Width	768
ViT Layers	4
ViT Heads	24

```
def causal_sequence_modeling(
    prompt_tokens,  # the [L, d] prompt tokens (L=prompt length)
    sep_token,      # the [1, d] learned token to separate prompt and trajectory history
    obs_tokens,     # the [T, d] obs tokens (T=time step)
    act_tokens,     # the [T-1, d] action tokens
    pos_embd,       # learned positional embedding
):
    # interleave obs and action tokens
    traj_tokens = interleave(obs_tokens, act_tokens)  # [2T-1, d]
    # assemble input tokens
    x = concat([prompt_tokens, sep_token, traj_tokens])
    x = x + pos_embd

    # apply GPT layers with causal mask
    for i in range(num_layers):
        # self-attention
        x = x + causal_attn_i(q=x, kv=x)
        # feed forward
        x = x + ffw_i(x)

    # the last token is the predicted action token
    predicted_act_token = x[-1]
    return predicted_act_token
```

Pseudocode 2: Plain sequence modeling that directly concatenates prompt and trajectory history and repetitively perform causal self-attention operation.

C.3.2. VIMA-FLAMINGO

Flamingo (Alayrac et al., 2022) is a vision-language model that learns to generate textual completion in response to multimodal prompts. It embeds a variable number of prompt images into a fixed number of tokens via the Perceiver Resampler module (Jaegle et al., 2021b), and conditions the language decoder on encoded prompts by cross-attention. Flamingo does not work with embodied agents out of the box. We adapt it by replacing the output layer with robot action heads (hyperparameters listed in Table 4) and using tokenized rollout histories as inputs. We thus call it "**VIMA-Flamingo**".

We train it end-to-end with causal behavior cloning loss. VIMA-Flamingo differs from ours since it processes image observations into a fixed number of visual tokens through a learned Perceiver Resampler. Model hyperparameters for our reimplementation of the Perceiver Resampler is listed in Table 6.

Table 6: Model hyperparameters for Perceiver Resampler used in VIMA-Flamingo method.

Hyperparameter	Value
Number of Latent Queries	4
Number of Blocks	4
Self-Attn per Block	4
Self-Attn Heads	24
Cross-Attn Heads	24

C.3.3. VIMA-GPT

VIMA-GPT is a GPT-based behavior cloning agent conditioned on tokenized multimodal prompts with the GPT architecture. It autoregressively decodes next actions given multimodal prompts and interaction histories. We optimize this method end-to-end with causal behavior cloning loss. Similar to prior works of casting RL problems as sequence modeling (Chen et al., 2021; Janner et al., 2021; Zheng et al., 2022), it encodes an image into a single "state" token through a learned ViT encoder. It also directly models entire trajectory sequences prepended with prompt tokens. Therefore, it differs from our method in the representation of observation tokens and prompt conditioning. For visual tokenizer, we employ a learned ViT with hyperparameters listed in Table 5.

C.4. Mask R-CNN Detection Model

Finally, we elaborate on the mask R-CNN model (He et al., 2017) for scene parsing and object extraction. We fine-tune a pre-trained lightweight mask R-CNN (`mask_rcnn_R_50_FPN_3x`) from Wu et al. (2019) to adapt to scenes and images in our tabletop environment. We fine-tune it on a subset of agent training dataset. It contains 100 trajectories for each task, resulting in 22,741 images and 61,822 annotations in total. We use learning rate 5×10^{-4} and train for 10 epochs. During model selection, we particularly favor models with high recall to reduce the number of missed objects. To compensate for resulting false-positives, we adopt object augmentation during agent training (Appendix, Sec. D).

A visualization of its output is provided in Figure A.20. We do not use the predicted object names in our models.

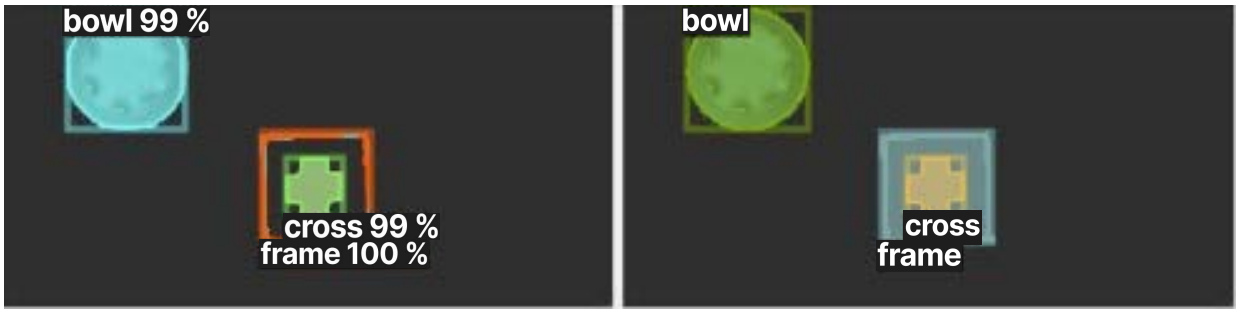

Figure A.20: Visualization of fine-tuned mask R-CNN. *Left*: Prediction from the detection model. *Right*: Ground-truth scene parsing. The detection model agrees well with ground-truth objects.

D. VIMA Training Details

We follow the best practice to train Transformer models using the AdamW optimizer (Loshchilov & Hutter, 2019), learning rate warm-up, cosine annealing (Loshchilov & Hutter, 2017), etc. Training hyperparameters are provided in Table 7. We use GEGLU activation (Shazeer, 2020) inside Transformer models across all methods.

Table 7: Hyperparameters used during training.

Hyperparameter	Value
Learning Rate	0.0001
Warmup Steps	7K
LR Cosine Annealing Steps	17K
Weight Decay	0
Dropout	0.1
Gradient Clip Threshold	1.0

To make trained models robust to detection inaccuracies and failures, we apply *object augmentation* by randomly injecting *false-positive* detection outputs. Concretely, for observation at each time step, we sample number of augmented objects i.i.d. $n_{\text{augmented objects}} \sim \text{Cat}(K, \mathbf{p})$, where $\text{Cat}(\cdot)$ denotes a categorical distribution with K supports parameterized by $\mathbf{p}$. For each augmented object, we then randomly sample a bounding box and corresponding cropped image to add to object tokens. In our experiments, we set $\mathbf{p} = \{0 : 0.95, 1 : 0.05\}$ with $K = 2$.

D.1. Vary Model Capacity

We train a spectrum of 7 models ranging from 2M to 200M parameters. To vary the model capacity, we follow prior work (Chowdhery et al., 2022) to change embedding dimension and number of layers. We list configurations for methods with cross-attention prompt conditioning (i.e., ours and VIMA-Flamingo) in Table 8, and configurations for methods only with causal self-attention (i.e., VIMA-Gato and VIMA-GPT) in Table 9.

Table 8: Configurations for differently sized models with cross-attention prompt conditioning.

Model Size (M)	Embedding Dimension	Num Blocks	X-Attn Heads	Self-Attn Heads
2	256	1	8	8
4	256	2	8	8
9	320	3	10	10
20	384	4	12	12
43	512	5	16	16
92	640	7	20	20
200	768	11	24	24

Table 9: Configurations for differently sized models with causal self-attention prompt conditioning.

Model Size (M)	Embedding Dimension	Num Blocks	Self-Attn Heads
2	64	1	2
4	96	2	3
9	192	3	6
20	320	4	10
43	512	5	16
92	768	7	24
200	768	18	24

E. Extended Experiment Results

E.1. Training Time and Compute

All experiments are conducted on cluster nodes, each with 8 NVIDIA V100 GPUs. The largest experiment takes approximately one day. We utilize DDP (distributed data parallel) to accelerate the training.

E.2. Model Scaling

E.2.1. Numerical Results

We present numerical results that constitute Fig. 4 in Table 10. The claim of "up to $2.9\times$ improvement" made in Abstract and Sec. 1 is calculated as follows. The best competing variant is VIMA-Gato. On the hardest L4, our method shows the most significant relative improvement with a model size of 20M. We compute the performance gap, divide by VIMA-Gato's performance, and only keep the first digit after decimal to obtain the result.

Table 10: Model scaling numerical results that constitute Fig. 4. Numbers in the first row indicate robot controller parameter count.

Level	Method	2M	4M	9M	20M	43M	92M	200M
L1	Ours	**76.5**	**79.2**	**77.4**	**77.1**	**78.2**	**79.3**	**81.5**
	VIMA-Gato	37.6	42.6	44.2	46.1	49.5	57.0	58.0
	VIMA-Flamingo	42.4	48.9	45.6	46.6	47.0	47.2	47.4
	VIMA-GPT	30.0	37.0	44.9	48.5	48.0	47.9	46.9
L2	Ours	**77.1**	**79.2**	**78.2**	**77.6**	**77.6**	**80.1**	**81.5**
	VIMA-Gato	35.9	39.3	41.3	44.1	46.6	53.9	53.1
	VIMA-Flamingo	41.0	46.5	44.6	44.6	45.4	47.1	46.0
	VIMA-GPT	29.8	35.0	43.3	45.8	45.9	47.4	46.9
L3	Ours	**77.3**	**77.8**	**78.5**	**77.3**	**81.8**	**81.9**	**78.7**
	VIMA-Gato	29.0	33.2	37.5	40.2	42.5	45.6	46.0
	VIMA-Flamingo	35.0	41.9	39.2	40.5	40.3	42.1	40.7
	VIMA-GPT	25.3	29.3	39.0	43.5	43.0	42.6	42.2
L4	Ours	**25.7**	**49.0**	**47.1**	**48.8**	**49.0**	**49.6**	**48.6**
	VIMA-Gato	13.3	13.2	12.2	12.3	12.8	13.5	16.8
	VIMA-Flamingo	12.3	11.6	10.7	12.1	10.7	11.1	12.1
	VIMA-GPT	11.1	10.3	12.7	14.2	11.8	12.1	12.1

E.3. Data Scaling

E.3.1. Detailed Setup

To ensure all methods are fairly pre-trained on the same amount of data (i.e., they have roughly the same amount of built-in information, thus the x-axis in Fig. 4 faithfully corresponds to the extra bits of information seen during further training), we initialize variants that directly learn from raw pixels with MVP pre-trained ViT (Xiao et al., 2022; Radosavovic et al., 2022). It is further MAE fine-tuned (He et al., 2021), using the *same* in-domain data as for the Mask R-CNN object detector. Note that the MVP pre-trained then domain fine-tuned ViT also updates weights jointly with robot controllers later on. We use the `ViT-B` backbone from MVP. The in-domain data for fine-tuning include 100 trajectories for each task.

E.3.2. Numerical Results

We present numerical results that constitute Fig. 4 in Table 11. The claim of "$2.7\times$ improvement" made in Abstract and Sec. 1 is calculated as follows. The best competing variant is VIMA-Gato that achieves 12.2% average success rate trained with full data on L4. Our method trained with 10% data achieves 46% average success rate on the same level. We compute the performance gap, divide by VIMA-Gato's performance, and only keep the first digit after decimal to obtain the result.

Table 11: Data scaling numerical results that constitute Fig. 4. Numbers in the first row indicate the size of training dataset.

Level	Method	0.1%	1%	10%	Full (100%)
L1	Ours	**0.0**	**36.3**	**76.3**	**79.3**
	VIMA-Gato	**0.0**	11.5	41.5	57.5
	VIMA-Flamingo	**0.0**	2.0	37.7	52.3
	VIMA-GPT	**0.0**	6.0	30.9	52.8
L2	Ours	**0.0**	**34.3**	**75.8**	**80.1**
	VIMA-Gato	**0.0**	10.1	37.9	41.2
	VIMA-Flamingo	**0.0**	2.0	33.8	32.6
	VIMA-GPT	**0.0**	6.0	29.7	40.3
L3	Ours	**0.0**	**15.4**	**73.2**	**81.9**
	VIMA-Gato	**0.0**	10.2	34.8	40.9
	VIMA-Flamingo	**0.0**	1.0	33.1	33.6
	VIMA-GPT	**0.0**	5.5	28.6	39.2
L4	Ours	**0.0**	**17.0**	**46.0**	**49.6**
	VIMA-Gato	**0.0**	2.7	10.8	12.2
	VIMA-Flamingo	**0.0**	0.5	11.2	12.0
	VIMA-GPT	**0.0**	1.1	7.1	14.3

E.3.3. What If Baseline Variants' ViT Is Trained from Scratch?

We further investigate what if baseline variants' ViT is trained from scratch and end-to-end with the robot controllers. We visualize the results in Fig. A.21 and numerically present them in Table 12. We annotate with arrows to indicate performance increase (↑) and decrease (↓). We highlight two findings.

First, MVP pre-trained ViT is most beneficial in the setting with sufficient in-domain training data (i.e., the 10% data scenario). It boosts the performance for the most competing baseline variant VIMA-Gato. However, in other settings with abundant in-domain data (i.e., the full data scenario) or insufficient in-domain data (i.e., 1% and 0.1% scenarios), the advantage of MVP pre-trained ViT diminishes and it even becomes detrimental. This aligns with the finding in previous empirical studies (Hansen et al., 2022). Second, in settings with reasonable amounts of in-domain data (i.e., the 1%, 10%, and 100% scenarios), our recommended recipe always outperforms other variants. We notice that such a data demand generally can be satisfied by both simulated robotics data (Mandlekar et al., 2021) and real robotics data (Dasari et al., 2019; Brohan et al., 2022). Therefore, it demonstrates that our recommended recipe is highly sample-efficient compared to alternative designs, especially under practical settings.

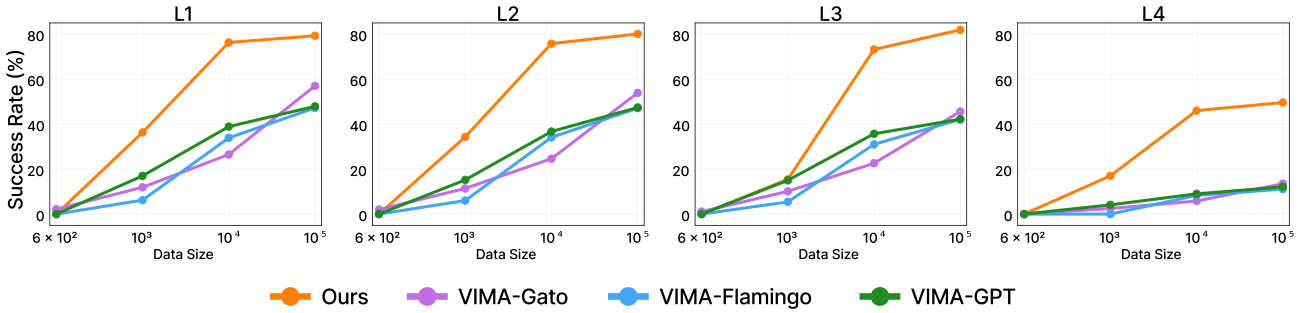

Figure A.21: Data scaling when baseline variants' ViT is trained from scratch. In settings with reasonable amounts of in-domain data (i.e., the 1%, 10%, and 100% scenarios), our recommended recipe always outperforms other variants.

E.4. Vary T5 Encoder Sizes

We vary the size of the pre-trained T5 encoder (Raffel et al., 2020) to study the effect of prompt encoding. We experiment with three T5 model capacities: `t5-small` (30M), `t5-base` (111M), and `t5-large` (368M). For all T5 variants, we

Table 12: Data scaling when baseline variants' ViT is trained from scratch, indicated inside parentheses. ↑ and ↓ denote performance increase and decrease. Numbers in the first row represent the size of training dataset.

Level	Method	0.1%	1%	10%	Full (100%)
L1	Ours	0.0	**36.3**	**76.3**	**79.3**
	VIMA-Gato	0.0 (**2.2** ↑)	11.5 (11.9 ↑)	41.5 (26.5 ↓)	57.5 (57.0 ↓)
	VIMA-Flamingo	0.0 (0.0)	2.0 (6.2 ↑)	37.7 (33.9 ↓)	52.3 (47.2 ↓)
	VIMA-GPT	0.0 (0.0)	6.0 (17.0 ↑)	30.9 (38.9 ↑)	52.8 (47.9 ↓)
L2	Ours	0.0	**34.3**	**75.8**	**80.1**
	VIMA-Gato	0.0 (**2.0** ↑)	10.1 (11.4 ↑)	37.9 (24.6 ↓)	41.2 (53.9 ↑)
	VIMA-Flamingo	0.0 (0.0)	2.0 (6.0 ↑)	33.8 (34.1 ↑)	32.6 (47.1 ↑)
	VIMA-GPT	0.0 (0.0)	6.0 (15.2 ↑)	29.7 (36.7 ↑)	40.3 (47.4 ↑)
L3	Ours	0.0	**15.4**	**73.2**	**81.9**
	VIMA-Gato	0.0 (**1.1** ↑)	10.2 (10.1 ↓)	34.8 (22.6 ↓)	40.9 (45.6 ↑)
	VIMA-Flamingo	0.0 (0.0)	1.0 (5.4 ↑)	33.1 (31.0 ↓)	33.6 (42.1 ↑)
	VIMA-GPT	0.0 (0.0)	5.5 (15.0 ↑)	28.6 (35.7 ↑)	39.2 (42.2 ↑)
L4	Ours	0.0	**17.0**	**46.0**	**49.6**
	VIMA-Gato	0.0 (0.0)	2.7 (2.5 ↓)	10.8 (5.8 ↓)	12.2 (13.5 ↑)
	VIMA-Flamingo	0.0 (0.0)	0.5 (0.0 ↓)	11.2 (8.3 ↓)	12.0 (11.1 ↓)
	VIMA-GPT	0.0 (0.0)	1.1 (4.1 ↑)	7.1 (9.0 ↑)	14.3 (12.1 ↓)

fine-tune the last two layers and freeze all other layers. We fix the parameter count of the decision-making part to be 200M. As shown in Table 13, we find no significant difference among the variants. Thus we set the standard `t5-base` as default for all our models.

Table 13: Performances of our method with differently sized pre-trained T5 prompt encoder. We fix the parameter count of the decision-making part to be 200M.

	`t5-small` (30M)	`t5-base` (111M)	`t5-large` (368M)
L1	78.8	81.5	80.8
L2	79.0	81.5	81.0
L3	80.3	78.7	81.0
L4	49.1	48.6	49.3

E.5. Policy Robustness

Increasing Amounts of Distractors. We study the policy robustness against increasing amounts of distractors in scenes. For all tasks being evaluated, we add one more distractor object. We run our largest VIMA model with 200M parameters. The result is presented in Table 14.

It turns out that the performance of VIMA degrades minimally with more distractors than the training distribution. This indicates that our agent has learned a reasonably robust policy against objects that are irrelevant to the task.

Table 14: Evaluation results on tasks with increased amounts of distractors. We fix the parameter count of the decision-making part to be 200M.

	L1	L2	L3	L4
Original	81.5	81.5	78.7	48.6
More Distractors	78.5	78.6	72.9	47.8
Relevant Performance Decrease (%)	3.6	3.5	7.3	1.6

Imperfect Prompts. We then study the policy robustness against imperfect prompts, including incomplete prompts (randomly masking out words with <UNK> token) and corrupted prompts (randomly swapping words, which could have changed the task meaning altogether). We run our largest VIMA model with 200M parameters, results are shown in Table 15.

Our well-trained model exhibits minimal performance decrease when evaluated on masked prompts and minor decrease on corrupted prompts. We attribute this robustness to the high-quality pre-trained T5 language backbone.

Table 15: Evaluation results with incomplete and corrupted prompts. We fix the parameter count of the decision-making part to be 200M.

	L1	**L2**	**L3**	**L4**
Original	81.5	81.5	78.7	48.6
Incomplete Prompts	80.8	81.1	77.0	48.0
Corrupted Prompts	78.2	78.1	73.8	45.3
Relevant Performance Decrease w/ Incomplete Prompts (%)	0.8	0.4	2.1	1.2
Relevant Performance Decrease w/ Corrupted Prompts (%)	4.2	4.3	6.6	7.2

F. Extended Related Work

In this section, we provide an extended review of related work as complementary to Section 6.

Multi-Task Learning by Sequence Modeling. In computer vision, Mask R-CNN (He et al., 2017), UberNet (Kokkinos, 2016), and 12-in-1 (Lu et al., 2020) leverage a single backbone model with multiple independent heads for different tasks. UVim (Kolesnikov et al., 2022) is another unified approach for vision that uses a language model to generate the guiding code for a second model to predict raw vision outputs. In multimodal learning, numerous works (Lu et al., 2022; Wang et al., 2022a; Zellers et al., 2021; 2022; Buch et al., 2022; Fu et al., 2021; Yang et al., 2022) investigate the unification of image, video, audio, and/or language modalities to deliver multi-purpose foundation models, although most of which are not equipped with decision-making capabilities. BEiT-3 (Wang et al., 2022c) performs masked data modeling on images, texts and image-text pairs to pre-train a backbone for various downstream tasks. MetaMorph (Gupta et al., 2022a) learns a universal controller over a modular robot design space.

Foundation Models for Embodied Agents. Embodied agent research (Duan et al., 2022; Batra et al., 2020; Ravichandar et al., 2020; Collins et al., 2021) is adopting the large-scale pre-training paradigm (Yang et al., 2023), powered by a collection of learning environments (Abramson et al., 2020; Shridhar et al., 2020; Savva et al., 2019; Puig et al., 2018; Team et al., 2021; Toyama et al., 2021; Shi et al., 2017). From the aspect of **pre-training for better representations**, Reid et al. (2022) fine-tunes from LLM checkpoints to accelerate policy learning. LaTTe (Bucker et al., 2022) and Embodied-CLIP (Khandelwal et al., 2021) leverage the frozen visual and textual representations of CLIP (Radford et al., 2021) for robotic manipulation. MaskDP (Liu et al., 2022a) pre-trains bidirectional transformers for various downstream embodied tasks. From the perspective of leveraging **transformer as agent architecture**, methods such as Dasari & Gupta (2020) and MOSAIC (Zhao et al., 2022) achieve superior performance in one-shot video imitation tasks. They both use the self-attention mechanism with auxiliary losses such as inverse dynamics loss (Dasari & Gupta, 2020) and contrastive loss (Zhao et al., 2022) to learn robot controllers. *Instruct*RL (Liu et al., 2022b) leverages jointly pre-trained vision-language models as robot agents to perform manipulation tasks. From the perspective of **large language models for robot learning**, Socratic Models (Zeng et al., 2022) composes multiple vision and language foundation models for multimodal reasoning in videos. ROSIE (Yu et al., 2023) leverages text-to-image diffusion models to augment existing robotic dataset (Brohan et al., 2022) via inpainting. MOO (Minderer et al., 2022) adopts a similar object-centric representation as ours for open-world object manipulation. Furthermore, Voyager (Wang et al., 2023) develops a LLM-powered agent operating in an open-ended virtual world (Fan et al., 2022).

Robot Manipulation and Benchmarks. There are many prior works that are not mentioned in the main paper that study different robotic manipulation tasks, such as instruction following (Shridhar et al., 2021; Lynch & Sermanet, 2021), constraint satisfaction (Bharadhwaj et al., 2021; Srinivasan et al., 2020; Thananjeyan et al., 2021), one-shot imitation (Paine et al., 2018; Huang et al., 2019; Dasari & Gupta, 2020; Aceituno et al., 2021; Zhao et al., 2022), rearrangement (Weihs et al., 2021; Szot et al., 2021; Liu et al., 2021; Ehsani et al., 2021; Gan et al., 2021; Stengel-Eskin et al., 2022), and reasoning (Gupta et al., 2019; Ahmed et al., 2021; Toyer et al., 2020; Lim et al., 2021). Multiple simulation benchmarks are introduced to study the above tasks: 1) **Indoor simulation environments**: Habitat (Savva et al., 2019; Szot et al., 2021) is equipped with a high-performance 3D simulator for fast rendering and proposes a suite of common tasks for assistive robots. 2) **Tabletop environments**: RLBench (James et al., 2019) and SURREAL (Fan et al., 2018; 2019) are other widely used simulator benchmarks studying robotics manipulation with tabletop settings. STRETCH-P&P (Zhang & Weihs, 2023) studies generalization across goals for reset-free reinforcement learning. All these aforementioned simulators and benchmarks do not natively support task specification and prompting with multiple modalities.

G. Limitations and Further Discussions

Reliance on a separate object detector. VIMA inherits the errors from the standalone object detector, which may struggle in the cases of occlusion or out-of-distribution object forms. However, using object detectors is not entirely without merits. First, it allows us to seamlessly switch to stronger detection models when they become available. For example, we can switch to object detectors that are more robust and open-vocabulary, such as OWL-ViT (Minderer et al., 2022). This would enable VIMA to transfer to real-world scenarios with minimal modifications. Second, by leveraging pre-trained vision pipelines, several concurrent works have demonstrated the superiority of object-centric representation in robot manipulation. For example, VIOLA (Zhu et al., 2022) achieves better performance with a pre-trained Region Proposal Network (Ren et al., 2015). MOO (Stone et al., 2023) also shows that a robot agent with OWL-ViT (Minderer et al., 2022) as the object detector significantly outperforms RT-1 (Brohan et al., 2022), which directly learns from raw pixels, on various real-world manipulation tasks. In fact, MOO (Stone et al., 2023) includes a baseline called "VIMA-like" that already demonstrates strong performance on real robots under real-world scenarios. As we witness image segmentation is becoming more robust and general-purpose (Kirillov et al., 2023), we envision such design choice will become more effective and further gain more popularity.

Limited simulator realism and task complexity. Our goal with VIMA-BENCH is to explore the multi-task ability, generalization, and understanding of multi-modality. Therefore, these aspects are not the primary focus of this work. However, we envision future works can combine this formulation with more physically realistic simulators such as Zhu et al. (2020), Srivastava et al. (2021), and Mittal et al. (2023).

Limited action primitives. We inherit the same high-level action space from well-established prior works, such as Transporter (Zeng et al., 2020). While "pick-and-place" and "wipe" seem simple, they do cover a wide range of tabletop manipulation tasks and are crucial to industrial use cases like warehouse robots (Yoon et al., 2003; Berscheid et al., 2020; Devin et al., 2020; Song et al., 2019). While VIMA is currently using these two actions, the algorithm design is general-purpose and does not make assumptions about the particular action choices. For example, VIMA would require only minimal modifications to support more low-level action spaces like joint-torque control.

H. Full Tables

This section contains more detailed tables that correspond to the results in Figure 4. Specifically, we show breakdown results on each task that constitute the model scaling results in Tables 16, 17, 18, and 19.

Table 16: L1 level generalization results. Model indicates robot controller parameter count. Integers in the first row refer to indices of tasks described in Appendix, Sec. B.

Model	Method	01	02	03	04	05	06	07	09	11	12	15	16	17
2M	Ours	**100.0**	**100.0**	**100.0**	**96.0**	37.0	**100.0**	**100.0**	**9.5**	**87.0**	64.0	**93.5**	**45.0**	**63.0**
	VIMA-Gato	62.0	61.0	22.5	13.5	7.0	44.5	54.0	4.0	48.0	**85.0**	44.5	43.0	0.0
	VIMA-Flamingo	56.0	56.0	53.5	36.5	**37.5**	45.0	55.5	3.5	54.0	83.5	40.5	28.5	2.0
	VIMA-GPT	59.5	50.5	7.5	7.0	0.5	43.5	49.5	2.0	61.5	76.5	27.5	5.0	0.0
4M	Ours	**100.0**	**100.0**	**100.0**	**99.5**	**45.5**	**100.0**	**100.0**	**10.5**	**90.5**	**90.0**	**96.5**	**46.5**	**51.0**
	VIMA-Gato	61.0	61.5	8.0	46.0	32.5	45.5	57.0	1.0	64.5	86.0	46.5	42.5	2.0
	VIMA-Flamingo	61.0	62.0	57.5	47.5	45.0	49.5	59.5	5.5	80.0	83.5	40.5	43.0	2.0
	VIMA-GPT	58.0	55.0	17.5	25.0	12.0	47.5	54.5	3.0	59.5	80.5	27.0	41.5	0.5
9M	Ours	**100.0**	**100.0**	**100.0**	**99.5**	**51.5**	**100.0**	**100.0**	**13.0**	**82.5**	58.5	**96.0**	**42.0**	**63.5**
	VIMA-Gato	59.0	61.0	41.0	50.5	38.5	47.5	59.5	9.5	58.0	80.5	44.0	24.0	2.5
	VIMA-Flamingo	58.5	60.0	46.0	49.0	42.5	45.5	60.5	4.0	66.5	81.5	36.5	41.5	1.0
	VIMA-GPT	58.5	54.5	40.5	47.5	37.5	47.5	58.5	9.0	72.0	**85.0**	38.5	34.0	1.0
20M	Ours	**100.0**	**100.0**	**100.0**	**100.0**	**59.5**	**100.0**	**100.0**	**13.5**	74.0	72.5	**96.5**	39.5	**47.5**
	VIMA-Gato	61.5	62.0	32.5	49.0	38.0	46.0	60.0	5.0	68.0	83.0	47.0	**46.5**	2.0
	VIMA-Flamingo	63.0	61.5	55.0	50.0	42.5	41.5	58.0	6.0	62.0	83.0	44.0	38.5	1.0
	VIMA-GPT	60.5	64.0	50.5	44.0	41.0	48.0	61.5	7.0	**85.0**	**84.0**	44.5	39.0	2.5
43M	Ours	**100.0**	**100.0**	**100.0**	**100.0**	**57.0**	**99.5**	**100.0**	**15.0**	**86.0**	69.5	**99.0**	40.0	**51.5**
	VIMA-Gato	57.0	65.5	59.0	57.5	43.5	50.0	56.0	5.0	67.0	**83.5**	63.0	37.0	0.0
	VIMA-Flamingo	54.5	57.0	54.5	54.0	45.0	43.5	55.5	6.0	67.5	82.5	49.0	**40.5**	1.5
	VIMA-GPT	58.0	60.5	69.5	53.5	41.5	47.0	55.5	4.0	66.5	81.5	45.0	**40.5**	1.5
92M	Ours	**100.0**	**100.0**	**99.5**	**100.0**	**58.0**	**100.0**	**100.0**	**14.0**	**80.5**	**92.0**	**98.5**	40.5	**48.5**
	VIMA-Gato	76.5	59.5	90.0	56.5	44.5	48.5	68.5	**14.0**	64.5	89.5	85.0	**43.0**	1.5
	VIMA-Flamingo	56.0	56.0	65.5	50.5	41.0	48.0	56.0	3.0	70.0	87.0	41.5	38.0	2.0
	VIMA-GPT	57.0	57.5	58.5	53.0	45.0	51.0	61.0	8.0	65.5	87.0	46.0	33.0	1.0
200M	Ours	**100.0**	**100.0**	**99.5**	**100.0**	**56.5**	**100.0**	**100.0**	**18.0**	**77.0**	**93.0**	**97.0**	**76.5**	**43.0**
	VIMA-Gato	79.0	68.0	91.5	57.0	44.5	54.0	74.0	**18.0**	61.0	88.5	83.5	33.5	2.5
	VIMA-Flamingo	56.0	58.5	63.0	48.5	38.0	48.5	62.5	3.5	66.5	86.0	40.0	43.5	2.5
	VIMA-GPT	62.0	57.5	41.0	55.5	45.5	47.5	54.5	8.5	**77.0**	81.5	41.0	38.0	0.5

Table 17: L2 level generalization results. Model indicates robot controller parameter count. Integers in the first row refer to indices of tasks described in Appendix, Sec. B.

Model	Method	01	02	03	04	05	06	07	09	11	12	15	16	17
2M	Ours	**100.0**	**100.0**	**100.0**	**95.5**	**37.5**	**100.0**	**100.0**	**17.5**	**87.5**	67.0	**97.5**	**46.0**	**54.5**
	VIMA-Gato	49.5	49.0	23.0	17.5	5.0	47.5	46.5	5.5	50.0	**82.5**	49.0	42.0	0.5
	VIMA-Flamingo	45.5	46.0	56.0	39.5	35.5	49.0	47.0	9.0	53.0	80.0	43.0	29.5	1.0
	VIMA-GPT	51.0	45.5	9.5	7.0	0.5	45.5	45.0	0.0	65.0	81.5	32.0	5.0	0.0
4M	Ours	**100.0**	**100.0**	**100.0**	**99.5**	**44.5**	**99.5**	**100.0**	**14.5**	**89.5**	**91.5**	**95.5**	**43.0**	**52.5**
	VIMA-Gato	44.5	52.0	9.0	39.0	28.0	49.5	48.5	2.0	64.0	86.5	44.5	42.5	2.0
	VIMA-Flamingo	49.5	50.5	51.0	48.0	43.0	50.5	53.5	5.5	81.5	82.5	48.5	39.5	1.0
	VIMA-GPT	50.5	49.5	16.5	25.5	12.0	41.0	47.0	4.0	63.0	79.0	28.5	39.0	0.0
9M	Ours	**100.0**	**100.0**	**100.0**	**100.0**	**49.5**	**100.0**	**100.0**	**19.0**	**80.5**	65.0	**95.5**	**42.0**	**66.0**
	VIMA-Gato	47.0	44.5	39.5	46.5	37.5	48.5	51.0	5.5	59.0	83.0	51.5	23.5	1.0
	VIMA-Flamingo	48.0	47.5	49.0	52.5	42.0	47.5	48.5	8.5	66.0	81.5	45.5	**42.0**	2.0
	VIMA-GPT	48.5	47.0	43.5	47.0	37.0	47.5	45.5	10.5	74.5	**85.0**	43.5	33.0	1.0
20M	Ours	**100.0**	**100.0**	**100.0**	**100.0**	**61.0**	**100.0**	**100.0**	**16.5**	75.5	75.0	**96.0**	37.5	**47.5**
	VIMA-Gato	44.0	51.5	39.0	51.0	38.5	47.5	52.5	6.0	65.5	**84.0**	52.5	40.5	1.0
	VIMA-Flamingo	48.5	49.0	55.5	48.0	42.5	46.5	52.0	6.0	66.0	82.0	47.5	37.0	0.5
	VIMA-GPT	50.5	49.5	53.0	44.5	43.5	47.0	46.0	8.0	**83.5**	80.0	46.5	**41.0**	2.5
43M	Ours	**100.0**	**100.0**	**100.0**	**100.0**	**54.5**	**100.0**	**100.0**	**14.5**	**83.5**	69.0	**98.0**	38.5	**51.5**
	VIMA-Gato	50.0	51.5	53.0	57.5	42.5	47.0	51.0	8.5	67.0	**83.0**	63.5	32.0	0.5
	VIMA-Flamingo	48.0	46.5	52.0	51.5	43.5	45.0	51.5	5.0	68.0	81.5	52.5	**44.0**	1.5
	VIMA-GPT	45.0	49.0	64.5	53.5	40.0	46.5	48.5	8.5	68.0	82.0	50.0	40.0	1.5
92M	Ours	**100.0**	**100.0**	**99.0**	**100.0**	**57.5**	**99.5**	**100.0**	**19.5**	**81.5**	92.0	**97.5**	**42.0**	**53.5**
	VIMA-Gato	64.5	50.0	83.0	56.5	46.0	55.5	54.5	10.5	64.5	**92.5**	81.0	**42.0**	1.0
	VIMA-Flamingo	53.0	48.5	67.5	53.0	43.0	49.0	53.0	4.5	67.0	84.0	50.0	40.0	1.0
	VIMA-GPT	50.5	55.0	55.5	54.5	43.0	51.5	54.5	10.5	68.5	87.0	49.5	34.0	3.0
200M	Ours	**100.0**	**100.0**	**99.5**	**100.0**	**54.5**	**100.0**	**100.0**	**17.5**	**77.0**	**93.0**	**98.5**	**75.0**	**45.0**
	VIMA-Gato	56.5	53.5	88.0	55.5	43.5	55.5	53.0	14.0	63.0	90.5	81.5	33.0	4.0
	VIMA-Flamingo	51.0	52.5	61.5	49.5	38.5	47.5	55.5	5.5	70.5	82.0	42.0	39.0	3.0
	VIMA-GPT	52.0	52.0	49.5	54.5	45.5	52.5	51.0	11.0	76.5	84.0	43.0	38.0	0.5

Table 18: L3 level generalization results. Model indicates robot controller parameter count. Integers in the first row refer to indices of tasks described in Appendix, Sec. B.

Model	Method	01	02	03	04	05	06	07	09	11	15	16	17
2M	Ours	**100.0**	**100.0**	**100.0**	**98.0**	**34.5**	**100.0**	**99.5**	**17.0**	**97.5**	**94.0**	**48.5**	**39.0**
	VIMA-Gato	45.5	48.0	28.0	23.0	3.0	45.5	45.0	2.5	40.5	29.5	37.0	1.0
	VIMA-Flamingo	41.5	54.5	50.5	39.5	29.0	45.0	49.5	5.5	57.5	22.5	25.0	0.0
	VIMA-GPT	48.5	50.0	5.0	7.0	2.5	47.0	45.5	2.0	69.5	22.5	5.0	0.0
4M	Ours	**99.5**	**100.0**	**100.0**	**98.0**	**44.0**	**99.5**	**99.5**	**12.0**	**92.5**	**98.5**	**47.0**	**43.5**
	VIMA-Gato	44.5	55.0	9.5	37.5	24.5	47.0	50.0	3.5	60.0	30.5	37.5	0.0
	VIMA-Flamingo	46.0	53.5	59.0	49.5	35.5	47.5	48.0	7.0	87.5	30.5	39.5	0.0
	VIMA-GPT	44.0	47.0	14.5	22.0	9.0	39.5	40.0	2.0	62.0	28.5	43.0	1.0
9M	Ours	**99.5**	**100.0**	**100.0**	**98.5**	**44.5**	**99.5**	**99.5**	**18.5**	**88.5**	**98.5**	**48.5**	**46.5**
	VIMA-Gato	44.5	53.5	42.5	52.0	28.0	46.5	51.5	6.0	67.0	35.0	23.0	0.5
	VIMA-Flamingo	44.5	53.0	53.0	48.5	33.0	41.0	45.5	8.0	72.5	27.0	44.5	0.5
	VIMA-GPT	49.0	50.5	39.0	46.5	30.5	43.0	52.0	6.5	84.0	31.5	35.0	0.5
20M	Ours	**98.0**	**100.0**	**100.0**	**98.5**	**55.5**	**100.0**	**99.5**	**15.0**	88.5	**99.5**	**44.0**	**29.5**
	VIMA-Gato	46.5	55.0	44.5	57.0	31.5	47.5	51.5	2.5	72.5	30.5	**44.0**	0.0
	VIMA-Flamingo	47.0	54.5	53.0	55.0	36.0	42.5	48.0	6.5	70.0	33.0	41.5	0.0
	VIMA-GPT	50.0	60.5	56.5	48.0	33.5	51.0	46.0	6.5	**92.5**	32.5	43.5	1.5
43M	Ours	**99.0**	**100.0**	**100.0**	**98.0**	**47.5**	**100.0**	**99.5**	**18.5**	**93.0**	**98.0**	**45.0**	**84.0**
	VIMA-Gato	44.0	55.0	59.5	58.0	34.0	49.0	54.0	7.0	74.0	40.0	35.0	0.5
	VIMA-Flamingo	47.0	54.0	56.5	52.5	37.0	46.5	44.5	6.5	69.5	27.0	43.0	0.0
	VIMA-GPT	47.5	57.0	61.0	50.0	34.5	48.0	53.5	8.0	74.0	40.5	41.5	0.5
92M	Ours	**99.0**	**99.5**	**99.5**	**97.0**	**58.0**	**100.0**	**99.0**	13.0	**94.5**	**99.0**	42.0	**82.5**
	VIMA-Gato	61.5	54.0	73.0	56.0	36.0	50.0	48.0	**17.0**	66.5	44.0	41.5	0.0
	VIMA-Flamingo	51.0	51.5	68.0	51.5	36.5	50.5	47.0	6.0	69.5	28.0	**45.5**	0.5
	VIMA-GPT	50.0	56.5	63.0	52.5	32.0	49.5	53.0	5.0	78.0	34.5	37.5	0.0
200M	Ours	**99.0**	**100.0**	**100.0**	**97.0**	**54.5**	**100.0**	**99.0**	**17.5**	**90.5**	**97.5**	**46.0**	**43.5**
	VIMA-Gato	51.0	58.0	84.5	56.5	35.5	53.5	49.0	15.0	65.0	52.0	33.0	0.0
	VIMA-Flamingo	49.0	50.0	66.5	47.0	35.0	47.5	50.0	4.0	66.0	30.5	43.5	0.5
	VIMA-GPT	52.0	51.0	55.0	49.5	40.0	46.0	50.5	5.0	82.0	37.0	38.0	1.5

Table 19: L4 level generalization results. Model indicates robot controller parameter count. Integers in the first row refer to indices of tasks described in Appendix, Sec. B.

Model	Method	08	10	13	14
2M	Ours	6.5	0.0	**0.0**	**96.5**
	VIMA-Gato	21.0	**0.5**	**0.0**	32.0
	VIMA-Flamingo	22.0	0.0	**0.0**	27.5
	VIMA-GPT	**22.5**	0.0	**0.0**	22.0
4M	Ours	**97.0**	0.0	**0.0**	**99.0**
	VIMA-Gato	17.0	**2.0**	**0.0**	34.0
	VIMA-Flamingo	17.0	0.5	**0.0**	29.0
	VIMA-GPT	19.0	0.0	**0.0**	22.5
9M	Ours	**92.0**	**0.0**	**0.0**	**96.5**
	VIMA-Gato	18.0	**0.0**	**0.0**	31.0
	VIMA-Flamingo	21.5	**0.0**	**0.0**	21.5
	VIMA-GPT	20.5	**0.0**	**0.0**	30.5
20M	Ours	**100.0**	0.0	**0.0**	**95.5**
	VIMA-Gato	20.5	0.0	**0.0**	29.0
	VIMA-Flamingo	21.0	0.0	**0.0**	27.5
	VIMA-GPT	20.5	**0.5**	**0.0**	36.0
43M	Ours	**99.0**	**0.0**	**0.0**	**97.0**
	VIMA-Gato	21.0	**0.0**	**0.0**	30.5
	VIMA-Flamingo	18.5	**0.0**	**0.0**	24.5
	VIMA-GPT	17.5	**0.0**	**0.0**	30.0
92M	Ours	**100.0**	0.0	**0.0**	**98.5**
	VIMA-Gato	22.0	0.0	**0.0**	32.0
	VIMA-Flamingo	19.5	0.0	**0.0**	25.0
	VIMA-GPT	18.5	**0.5**	**0.0**	29.5
200M	Ours	**100.0**	**0.0**	**0.0**	**94.5**
	VIMA-Gato	30.5	**0.0**	**0.0**	37.0
	VIMA-Flamingo	24.5	**0.0**	**0.0**	24.0
	VIMA-GPT	20.0	**0.0**	**0.0**	28.5