Is Imitation All You Need? Generalized Decision-Making with Dual-Phase Training

Yao Wei[2,6], **Yanchao Sun**[3], **Ruijie Zheng**[3], **Sai Vemprala**[5], **Rogerio Bonatti**[1], **Shuhang Chen**[5]
Ratnesh Madaan[1], **Zhongjie Ba**[2,6], **Ashish Kapoor**[5] and **Shuang Ma**[1⋆]
[1]Microsoft [2]Zhejiang University [3]University of Maryland ,College Park
[5] Scaled Foundations [6]Jiaxing Research Institute, Zhejiang University
⋆ Project lead. Corresponding to yunyikristy@gmail.com

Abstract

We introduce DualMind, a generalist agent designed to tackle various decision-making tasks that addresses challenges posed by current methods, such as overfitting behaviors and dependence on task-specific fine-tuning. DualMind uses a novel "Dual-phase" training strategy that emulates how humans learn to act in the world. The model first learns fundamental common knowledge through a self-supervised objective tailored for control tasks and then learns how to make decisions based on different contexts through imitating behaviors conditioned on given prompts. DualMind can handle tasks across domains, scenes, and embodiments using just a single set of model weights and can execute zero-shot prompting without requiring task-specific fine-tuning. We evaluate DualMind on MetaWorld [40] and Habitat [31] through extensive experiments and demonstrate its superior generalizability compared to previous techniques, outperforming other generalist agents by over 50% and 70% on Habitat and MetaWorld, respectively. On the 45 tasks in MetaWorld, DualMind achieves over 30 tasks at a 90% success rate. Our source code is available at https://github.com/yunyikristy/DualMind.

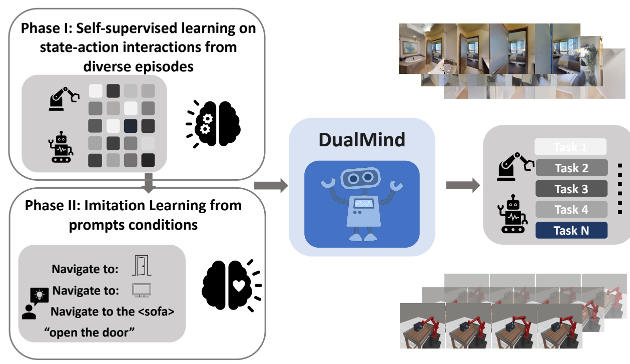

Figure 1: A high-level overview of DualMind's Dual-phase training scheme.

1. Introduction

Transformer-based models, combined with large-scale data, have shown success in generalizing across various tasks in both language and vision. Notable examples include BERT [11], GPT [28], MAE [16], CLIP [27] and Flamingo [1], etc. Recently, there has been a significant focus on developing such general-purpose models for sequential decision-making and control tasks, such as GATO [32]. The prominent approach is to train a decoder-only Transformer with Imitation Learning (IL) on massive datasets from all targeted tasks. By training with prompts, the model can perform zero-shot inference with just task prompts.

However, such IL-based approaches to general-purpose models face limitations when it comes to sequential control tasks, as highlighted below: (1) *Memorizing behaviors hinders generalization to diverse tasks*: Imitating expert behaviors can lead to memorization and over-fitting of specific behaviors that may not be applicable to new situations or variations of tasks, thus limiting the model's ability to generalize. This limitation is particularly challenging when dealing with a wide range of decision-making tasks that have vastly different configurations, transition functions, and state and action spaces. (2) *Dependence on high-quality data impedes practical application*: IL methods rely heavily on the availability of high-quality expert demonstrations, which can be difficult and expensive to obtain. When the available data is of low quality or not representative of the target task, the performance of the model may suffer.

In light of the aforementioned limitations, self-supervised pretraining has emerged as a viable solution. By focusing on learning common underlying information, a pretrained model can be better equipped to handle diverse tasks. Recently, a study known as SMART [36] has demonstrated the potential of self-supervised pretraining for multi-task decision-making.

Although SMART has shown promising results in promoting generalization, it still requires additional fine-tuning to adapt to each task. Furthermore, it has only been demonstrated on a small set of tasks on Deepmind control suite (DMC) [37]. For decision-making problems that involve numerous tasks with different configurations, finetuning the model for each task can become time-consuming and resource-intensive.

Given the limitations of both IL and self-supervised pretraining discussed earlier, a natural question arises: *How can we develop a decision-making approach that achieves a high degree of generalization without requiring task-specific fine-tuning?* In this paper, we propose DualMind, a generalist agent, to address this question, which stands for our proposed Dual-phase training scheme. The name 'DualMind' is derived from our main idea of Dual-phase training for generalized decision-making. Our approach introduces an Encoder-Decoder Control Transformer (Enc-Dec Control Transformer) that models state-action interactions from complex high-dimensional observations. To further improve computational efficiency, DualMind uses TokenLearner [34] as an attention-based Information Bottleneck (IB) [38] to compress the number of tokens so that to speed up training and inference. Building upon Enc-Dec Control Transformer, we propose a Dual-phase training scheme that initially prioritizes policy-independent transition probabilities and encourages the model to capture both short- and long-term temporal granularities. To facilitate zero-shot prompting, we train a second phase on a small fraction of model parameters to learn a generic policy by conditioning on various prompts (such as images, annotations, and language instructions) using a cross-attention mechanism (XAtten.). The Dual-phase training scheme parallels how humans learn to act in the world by first learning underlying common knowledge and subsequently making decisions based on different contexts. Our contributions are summarized below:

1. We propose DualMind, a solution for general-purpose decision-making that can handle various tasks using a single set of weights without task-specific fine-tuning.
2. We introduce a Dual-phase training scheme that overcomes limitations of IL and self-supervised learning.
3. We propose an Encoder-Decoder Transformer (Enc-Dec Control Transformer) that efficiently learns state-action transitions from high-dimensional observation spaces.
4. We conduct extensive experiments on Metaworld [40] and Habitat [31] and show that DualMind outperforms other generalist agents by over 50% and 70% on Habitat and MetaWorld, respectively. We also analyze and ablate different design choices to demonstrate the superior generalizability of DualMind.

2. Related work

Pretraining Visual Representations for Policy Learning: Recent studies such as R3M [24], APV [35] VPT [4], NRNS [15], PVR [26] and MVP[29] have shown that pretrained visual representations can significantly enhance the efficiency of downstream policy learning. However, these works mainly focus on learning object-centric semantics, potentially losing essential control-relevant information. To address this issue, VIP [22] formulates the problem as an offline goal-conditioned RL problem and proposes a visual representation algorithm capable of generating dense reward functions for downstream robotics tasks. On the other hand, COMPASS [21] introduces a general-purpose pretraining pipeline that effectively integrates multimodal signals for autonomous systems.

Transformer-Based Foundational Model: The use of high-capacity transformer architectures trained on large-scale datasets has led to significant breakthroughs in various domains. Examples include language models such as BERT [11], GPT-3 [7], T5 [30], and PaLM [10], as well as vision and vision-language models such as MAE [16], Multi-MAE [3], BiT [18], MuST [13], Flamingo [1], and CLIP [27]. For decision-making tasks, recent work such as SMART [36] has proposed a self-supervised pretraining framework tailored for control tasks. For robotics control problems, PACT [5] has shown that a pretrained representation could speed up various downstream tasks of mobile agents, such as navigation and localization.

A General-Purpose Model for Control: Since the groundbreaking success of GPT [28], recent research has focused on using Transformer decoder-based models to tackle control tasks in an auto-regressive manner. Decision Transformer (DT) [9, 41] builds on the architecture of GPT to create a generalist agent for sequential decision-making tasks. This has been followed by Multi-game DT [19] and Online-DT [42], which demonstrate the potential of DTs for multi-task and online learning. GATO [32] imitates expert demonstrations from a vast dataset and showcases its ability to handle a large number of tasks. VIMA [17] is an agent that can accept multi-modal prompts for solving various robotics manipulation tasks. In real-life applications, RT-1 [6] has demonstrated the efficacy of this approach in robotic control.

3. Preliminary and Overview of DualMind

3.1. Problem formulation

We focus on a set of tasks, denoted as $\mathcal{T}$, from two representative benchmarks, namely Metaworld [40] and Habitat [31], which cover the *Manipulation* and *Navigation* domains, respectively. As shown in Table 2, our selection of these two benchmarks allows us to conduct a comprehensive study on tasks with a wide variety of characteristics.

	Self-superv.	IL-prompt	Dual-phase (ours)
Learning	Pre.: generic info. FT: task-specific policy	Cond. generic policy	I: generic info. II: cond. generic policy
Data	Pre: Multi-task large set FT: Single-task small set	Multi-task large set+prompts	I: Multi-task large set II: +prompts
Optim. weights	Pre: whole model FT: Entire/freeze+Task heads	Entire model	I: Entire model II: Partial/freeze+XAtten.
Inference task	Single	Multiple	Multiple
No need FT	✗	✓	✓
Zero-shot promp.	✗	✓	✓
Final utilization	Many models for each task	Single model	Single model

Table 1: Comparisons of different training approaches.

Bench.	Dom.	Sce.	Emb.	Prom.	Tasks	Epis.
Meta.	Man.	1	1	inst.	50	50K
Habit.	Nav.	933	1	Obj. / Img	27	50K
Total	2	934	2	3	77	100K

Table 2: Dataset summerization Dom.: domains, Sce.: number scenes, Emb.: number of embodiments, Prom.: types of prompts, Epis.: number of episodes.

Here, we define a task as a partially observable Markov decision process (POMDP). The tasks we consider span across several factors, as defined below:

- *Domain*: This refers to tasks with different state/action spaces and application scenarios. In our study, *Manipulation* and *Navigation* are two domains we focus on.
- *Embodiment*: This factor is used to differentiate tasks that have different physics and action spaces. For instance, a robot arm and an embodied agent in MetaWorld and Habitat are considered as different embodiments. Differences can also exist in the same domain, such as arms with distinct joint torques and/or hardware configurations.
- *Scene*: This refers to tasks that are performed in different observation spaces, state spaces, and world structures. For example, in Habitat, agents that navigate in different rooms should adapt to various visual appearances and geometry structures.
- *Prompt*: This factor captures different forms of prompt conditions. In MetaWorld, prompts are natural language instructions, while in Habitat, we use a single RGB image or an object annotation as the navigation goal to prompt our model.

3.2. Overview of Dual-phase training scheme

In this section, we provide a brief overview of DualMind and compare it with two other prominent approaches: self-supervised pretraining (Self-superv.) and Imitation Learning with prompt conditions (IL-prompt). We also provide insights into the central idea behind our proposed approach. A summarized comparison of these approaches is shown in Table 1.

As shown in Fig. 2, In Phase I, we train the entire Enc-Dec Control Transformer (Sec. 4.1) with a self-supervised training objective to capture generic information of state-action transitions. In Phase II, we train only a small part of Enc-Dec Control Transformer attached with XAtten. on a diverse set of prompts for a conditional generic policy. After the Dual-phase training, we can obtain one model with a single set of weights that can be directly applied to a large number of tasks with corresponding prompts.

Compared to other generalist agents like GATO [32], which trains an imitating policy directly, DualMind demonstrates superior generalization capability. Moreover, our Phase II requires training only a small fraction of model weights while freezing the remaining parts, resulting in faster learning and reduced training cost when optimizing the model with the same number of iterations. Additionally, compared to self-supervised learning approaches such as SMART [36], DualMind is simple and effective, making it suitable for a wide range of application scenarios.

3.3. Insights

The central idea behind our Dual-phase is to mimic how humans learn to act in the world, first by learning underlying common knowledge and then by learning to make decisions based on different contexts. Our approach relates to InstructGPT [25], which aims to align language models with user intent by fine-tuning them with human feedback. In analogy to InstructGPT, our Phase I can be considered as learning a general model that captures the common essential information. However, as stated in InstructGPT, this is different from the objective of "following task instructions (i.e. prompt conditions)," and thus such a model is *misaligned*. Therefore, in the second phase, we leverage conditional IL to align the model so that it can perform well for any given prompts.

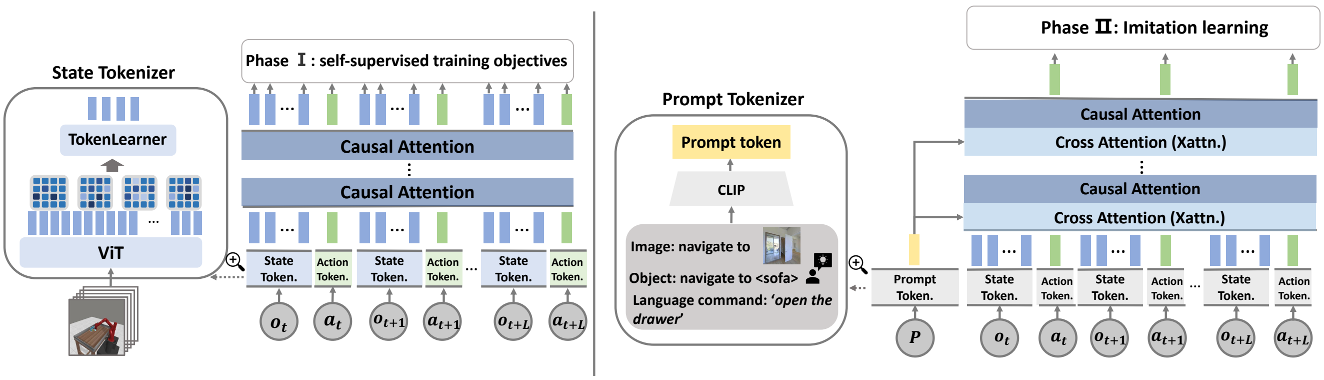

Figure 2: The architecture diagram of DualMind. **Left: Phase I.** Agent is trained with self-supervised learning objectives. During this phase, Transformer encoder and decoder jointly trained. **Right: Phase II.** Agent is trained with prompt conditional imitation learning. We tokenize task prompts with a pretrained CLIP encoder, and condition the Transformer decoder on the prompt through XAtten. layers. The gray color indicates frozen modules. (Detailed training objectives are in Sec. 4.2.)

4. Approach

In this section, we introduce our proposed DualMind. We present the model architecture in Section 4.1, and illustrate the training objective for DualMind in Section 4.2.

4.1. Model Architecture

We propose an Encoder-Decoder Control Transformer to process state-action interaction sequences, as illustrated in Figure 2. The implementation details of each component in the Enc-Dec Control Transformer are outlined below.

State tokenizer. We utilize a ViT model [12] to tokenize raw pixel states. To reduce the computational burden of dealing with sequential decision-making tasks, we leverage an attention-based Information Bottleneck (IB) to further compress the number of tokens so as to speed up training and inference (Fig. 2-left). Specifically, we use TokenLearner [34] which is an element-wise attention module that learns to soft-select image tokens, passing only the important ones to subsequent layers. The inclusion of TokenLearner sub-samples the 196 state tokens that come out of ViT to just 8 tokens that are then passed to the Transformer decoder layers.

Action tokenizer. To handle both continuous and discrete action spaces in our two domains, we adapt a strategy similar to GATO [32] by discretizing continuous actions into bins. We first flatten the actions into sequences of floating point values in row-major order, and then discretizing them into 256 uniform bins. Discrete actions are tokenized into 256 bins in the same way.

Transformer decoder. Our transformer decoder architecture is similar to Control Transformer [36], but with a modification. In our approach, we encode each state into 8 tokens, which is different from SMART's single token representation. This modification enables richer representation learning, making it suitable for more complex visual control environments.

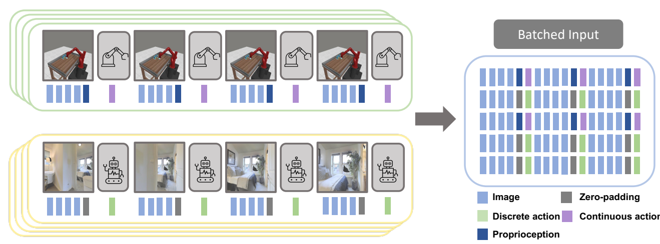

Figure 3: Batch input when training on multiple domains.

Prompt tokenizer. We tokenize prompts using a pretrained CLIP encoder [27]. For "image goal" prompts in Habitat, we use the CLIP image encoder, while for "object goal" prompts in Habitat and "language instruction" prompts in MetaWorld, we use the CLIP text encoder. A learnable linear layer is added on top of the CLIP encoders to map all prompts to prompt tokens with the same dimensions. During training in both phases, we freeze the CLIP encoders.

XAtten. layer. We condition the Transformer decoder by training it to learn from the prompt sequence through a series of cross-attention layers. The output sequence from each cross-attention layer is computed by $\mathrm{softmax}(\frac{q_H k_P^T}{\sqrt{d}})v_P$, where H is the sequence of episodes, P is prompt, and d is the embedding dimension. This design builds a stronger connection between the prompts and the demonstrations, which is an improvement over prefix-style prompting approaches [32]. We will show the benefits of this design in Sec. 5.4.

4.2. Training objectives

Phase I: self-supervised SMART training. The goal of this phase is to learn a good representation that captures control-relevant information shared across tasks. In this phase, we jointly train the encoder and the decoder following the self-supervised training objectives of SMART [36]. We use F_θ to denote the learned model with parameterization θ, such that $F_\theta(o_{i:j}, a_{i:j})$ refers to the output tokens of the decoder corresponding to raw inputs $o_{i:j}$ and $a_{i:j}$, the observation and action sequence from step i to step j. For a sequence of observations and actions denoted as $\{o_t, a_t, \cdots, o_{t+L}, a_{t+L}\}$ with context length L, we minimize the following objective.

$$\mathcal{L}_{P1} := \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3, \text{ where} \tag{1}$$

$$\mathcal{L}_1 := \sum\nolimits_{i=0}^{L-1} l\left(f_1(F_\theta(o_{t:t+i}, a_{t:t+i})), \bar{\phi}(o_{t+i+1})\right), \tag{2}$$

$$\mathcal{L}_2 := \sum\nolimits_{i=1}^{L} l\left(f_2(F_\theta(o_{t:t+i}, a_{t:t+i-1}), a_{t+i}\right), \tag{3}$$

$$\mathcal{L}_3 := \sum\nolimits_{i=1}^{L-1} l\left(f_3(F_\theta(\mathsf{Mask}(o_{t:t+L}, a_{t:t+L})), a_{t+i}\right). \tag{4}$$

Here l is a loss function that is selected by the variable type. For latent states, we use a mean squared error, while for discrete actions, we use the cross-entropy loss. $\mathcal{L}_1$ is to learn a forward prediction head f_1 that can predict the next state representation based on the historical interactions. Since the groundtruth state representation is unknown, we use the learned state embedding from the ViT model to encode the next observation, denoted as $\bar{\phi}$ where the overline stands for gradient stopping. $\mathcal{L}_2$ aims to recover the action token in each step conditioning on the history and the next state. $\mathcal{L}_3$ masks a proportion of input tokens and learns to recover the masked actions, which can extract long-term temporal dependence for control.

Phase II: Imitation learning with prompt conditions. In this phase, we train the model to follow prompt conditions. We formulate various tasks as a conditional generation problems, where the conditions can be goals, commands, prompts, etc. During Phase II, we let the agent learn a conditional policy, using expert trajectories with associated prompts. Let ψ be the prompt tokenizer, and π be the learned policy whose inputs are the representation tokens given by the decoder. For an expert sequence $\{o_t, a_t, \cdots, o_{t+L}, a_{t+L}\}$ with prompt P, we minimize loss

$$\mathcal{L}_{P2} := \sum\nolimits_{i=0}^{L-1} l(\pi(F_\theta(o_{t:t+i}, a_{t:t+i}; \psi(P))), a_{t+i+1}). \tag{5}$$

Note that in this phase, we do not train the entire model F_θ, and instead only re-train a small fraction of it. More discussion is in Sec. 5.4.

5. Experiments

5.1. Experimental setup

Data. We evaluate and train DualMind on two benchmarks, Habitat [31] and MetaWorld [40]. Habitat is a photorealistic simulation platform for research in Embodied AI, emphasizing active perception and long-term planning, ·while MetaWorld is a simulated benchmark for multi-task learning and meta-reinforcement learning, comprising 50 distinct robotic manipulation environments. Training on datasets collected from both these benchmarks allows us to demonstrate the model's generalizability across domains, embodiments, scenes, and prompts. We provide a detailed introduction to these factors in Sec. 3.1 and summarize them in Table 2. Additionally, we use 10 tasks as an out-of-distribution testbed to showcase the model's generalization capability. More details about our data collection process can be found in Appendix A.

Comparing baselines. We compare DualMind with existing transformer-based approaches and present results from two versions of our model: a generalist agent trained on the full dataset (`DualMind`) and a single-domain specialist trained only on data from either MetaWorld or Habitat (`DualMind/single`). To ensure fair comparisons, we implemented related works ourselves and trained and evaluated them on the same data and model architecture. We provide information on each baseline below:

- `IL-only` is a model trained only through prompt-conditioned imitation learning, which is related to GATO but uses a different prompting conditioning method.
- `SMART-only` is a model trained only using SMART training objectives (purely self-supervised).
- `Jointly` is a model jointly trained with both SMART objectives and prompt-conditioned Imitation Learning loss.
- `GATO*` is the model described in the original paper. We include its reported performance on the Metaworld benchmark for reference. Notably, this model has 1.18 billion parameters and was trained on massive datasets, including 94.6k episodes from Metaworld. In comparison, `DualMind` has 175 million parameters and was trained on a smaller dataset consisting of 100k episodes, of which 50k are from MetaWorld.
- `GATO-CT` is a model we implemented ourselves, reproducing the main technical approaches presented in the original paper. For a fair comparison, we used the same base model architecture (Enc-Dec Control Transformer), but replaced our XAtten.-based prompting approach with their proposed prefix prompting approach. Similar to `IL-only`, only imitation learning loss is used to train with prompt conditions. We provide a detail description in Appendix. A.

Implementation details. Our implementation of Dual-

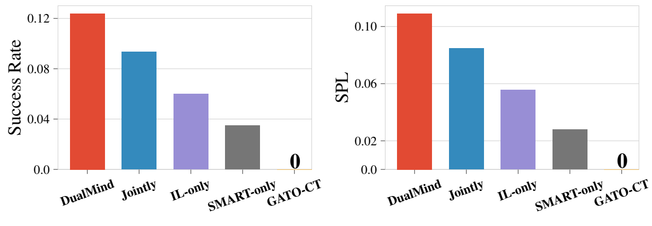

Figure 4: Comparisons of **generalist agents** on *Habitat 4 scenes* with 3 difficulty levels per scene. We roll out the agents 3 times on each scene and average the defined scores, and compare agents by Success Rate (SR) (left) and Success weighted by Path Length (SPL) (right).

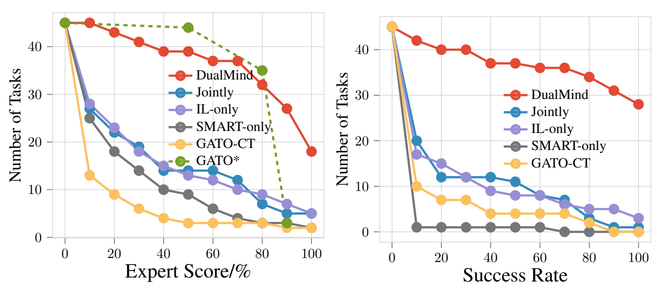

Figure 5: Comparisons of **generalist agents** on *MetaWorld 45 tasks* on Percentage of Expert Score (PES) (left) and Success Rate (SR) (right).

Mind uses a Transformer-based architecture consisting of a ViT-B [14] model, a TokenLearner [34], and a GPT model [28] as the encoder and decoder, respectively. The decoder consists of 8 layers and 8 attention heads, with a context length of L=6 and an embedding size of d=512[1]. We trained our model with the AdamW optimizer [20] and a learning rate of 5e-5 for both training phases. In Phase i, we trained the model for around 40 hours with BS=16 on 5x8xV100 GPUs. In Phase ii, the model was trained for about 12 hours with BS=128 on 2x8xV100 GPUs. Further implementation details are provided in Appendix A.

5.2. Capabilities of DualMind

In this section, we aim to demonstrate the capabilities of `DualMind` on all tasks. Note that, as a generalist agent, the performance on both MetaWorld and Habitat are achieved by a single model. The performance is shown in Fig. 4 and Fig. 5. To provide a reference for readers, we follow GATO's evaluation protocol and report the Percentage Expert Score (PES), which measures the number of distinct tasks for which each model performs above a given score threshold relative to the expert performance. For each task, we roll out the model 10 times and average the defined

[1]We found that longer context lengths can produce better performance, particularly on tasks that rely on long-range temporal dependencies. See Appendix B for more ablations.

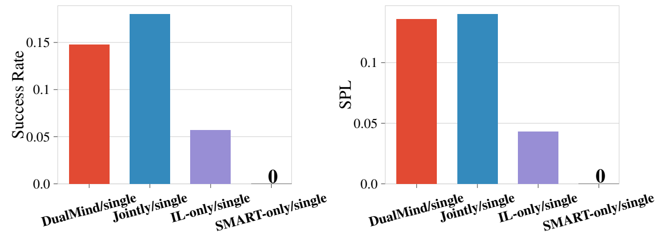

Figure 6: Comparisons of **single-domain specialist** on *Habitat 12 scenes*, as measured by SR (left) and SPL (right).

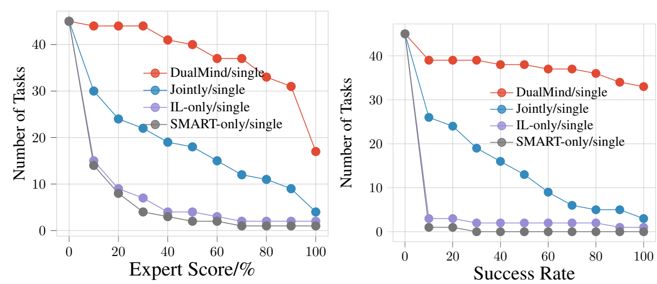

Figure 7: Comparisons of **single-domain specialist** on *MetaWorld 45 tasks*, as measured by Percentage of Expert Score (PES) (left) and Success Rate (SR) (right).

scores. As shown in Fig. 5, `DualMind` achieves over 90% expert score threshold across more than 27 tasks, outperforming `GATO*` by a large margin, which only has three tasks above the threshold. On lower expert score thresholds, for example, 80% and 50%, `DualMind` can also achieve comparable performance. However, it should be noted that GATO's performance was achieved by their 1.18B model trained on massive datasets. Therefore, this is just a reference for readers, and a fully fair comparison with GATO cannot be performed without access to both the model and data. To provide a more fair comparison, we compare `DualMind` with a self-implemented GATO (`GATO-CT`), which will be discussed in more detail in Sec. 5.1. We also report the number of tasks for which our model performs above a given Success Rate (SR). `DualMind` achieves 39 tasks at over 0.5 SR and can maintain good performance on higher SRs, with 34 tasks at over 0.8 SR and 28 tasks at over 1 SR. We present the performance of `DualMind` on Habitat by averaging across all 12 testing scenes and reporting the success rate (SR) and success weighted by path length (SPL) [2] evaluation metrics. As shown Fig. 4, `DualMind` outperforms the other baseline models by a large margin under both evaluation metrics. (See performance on each task in Appendix B.)

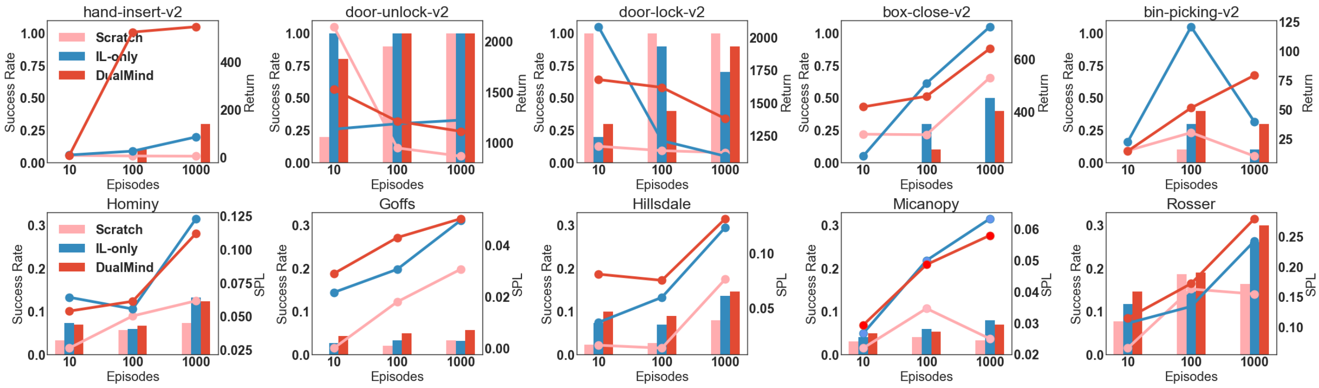

Figure 8: Few-shot comparisons of generalist agents on out-of-distribution tasks. The performance of success rate (left axis, bar-chart) or Return/SPL (right axis, line-chart) on different tasks after we performed 10-, 100-, and 1000-shot learning on DualMind(red), IL-only(blue), and Scratch(pink). Specifically, the bar charts (on the left axis) represent success rates, while the line charts (on the right axis) represent SPL in Habitat or Return in Metaworld.

5.3. Analysis

5.3.1 Different training regimes

Is imitation learning all you need for a generalist agent?

To answer the question, in this experiment, we compare DualMind with its counterpart trained only with Imitation Learning objective, i.e., IL-only. In Fig. 4 and Fig. 5, we present the comparison results between the generalist multi-domain agents. As shown in the figures, DualMind outperforms its IL-only counterpart by over 50% and 70% on Habitat and MetaWorld, respectively. Specifically, DualMind performs well on 39 out of 45 tasks over the 50% expert score threshold, while IL-only only performs well on 13 tasks. As the difficulty of the tasks increases, DualMind still maintains good performance, achieving 18 tasks and 28 tasks at the 100% expert score and SR, respectively, while IL-only only achieves 5 tasks. Similar observations can also be made when comparing the single-domain specialist agents (Fig. 7 and Fig. 6). (See performance on each tasks in Appendix B.)

We can infer from this that Imitation Learning alone may not suffice to build a truly general-purpose model, particularly when aiming to tackle tasks that span a broad range of domains. Even within a single domain, variations in embodiments, scenes, and instructions can pose significant challenges. We conducted additional investigations into the generalization capabilities by comparing different approaches on out-of-distribution tasks, as demonstrated in Section 5.3.2.

Can self-supervised learning well-align with instructions without FT?

To address this inquiry, we compare DualMind with its self-supervised equivalent, SMART-only, while also evaluating both single- and multi-domain agents. As depicted in Fig. 4 and Fig. 5, DualMind exhibits superior performance compared to SMART-only, with over 75% and 78% better results on Habitat and MetaWorld, respectively. Notably, SMART-only is unable to succeed on any tasks when applied to single-domain agents, whereas DualMind maintains a significant advantage, particularly on MetaWorld.

Our hypothesis is that SMART, being a pretrain-finetune pipeline, is unlikely to attain the desired performance without post-finetuning. Even when training SMART-only by providing prompts in the same manner as DualMind, zero-shot prompting may not be achievable due to limitations in the self-supervised training objective not being well-aligned with task instructions, as detailed in Section 3. Additionally, we noted that SMART-only surpasses its single-domain equivalent, suggesting its effectiveness in capturing shared knowledge across diverse data.

Do we need to train them in two phases?

As DualMind is trained using different objectives in two phases, one may question the necessity of such an approach. Firstly, from an optimization standpoint, training all four losses jointly may present more challenges in terms of steady optimization. Different optimization directions could potentially conflict with each other, and varying convergence rates could hinder all objectives from being trained to reach optimality. Furthermore, in terms of computational costs, DualMind only needs to optimize a small portion of the model weights in phase 2 (as demonstrated in the ablations presented in Section 5.4). This makes the training process more efficient and cost-effective compared to its jointly trained counterpart. In this experiment, we provide further empirical evidence to support this claim.

As illustrated in Fig. 4, Fig. 5, Fig. 6, and Fig. 7, Jointly outperforms IL-only and SMART-only, thereby confirming the necessity of utilizing all training objectives. However, it lags behind DualMind by a considerable margin in both multi- and single-domain comparisons. Interestingly, `Jointly` slightly outperforms `DualMind` in single-domain comparisons. We hypothesize that the optimization challenges may not be as significant as those encountered when training on data from the same domain.

5.3.2 Out-of-distribution tasks

The objective of this experiment is to assess the ability of our model to solve novel tasks. To achieve this, we evaluate our models on 10 held-out tasks from two domains, namely MetaWorld and Habitat. The MetaWorld tasks consist of "hand-insert-v2", "door-unlock-v2", "door-lock-v2", "box-close-v2", and "bin-picking-v2", whereas the Habitat tasks include "Goffs", "Hominy", "Hillsdale", "Micanopy", and "Rosser". To evaluate the performance of our models, we follow the evaluation protocol with GATO, which involves finetuning each agent on a limited number of demonstrations. Specifically, we conduct 10-, 100-, and 1000-shot learning and finetune all models for 10000 gradient steps. Further details on the evaluation protocol can be found in Appendix A.

We compare the performance of three models, namely `DualMind`, `IL-only`, and `Scratch`. `DualMind` and `IL-only` are finetuned with few-shot demonstrations from the base model of DualMind and `IL-only`, respectively. In contrast, `Scratch` refers to the model that is trained on few-shot demonstrations from randomly initialized model weights. Fig. 8 illustrates the success rate (shown on the left axis using a bar chart) and Return/SPL (shown on the right axis using a line chart) across different tasks after implementing 10-shot, 100-shot, and 1000-shot learning on these models. As demonstrated in Fig. 8, `Scratch` performs the worst among the three models in most cases. Upon comparing DualMind with `IL-only`, we observe that DualMind exhibits superior performance across various shot settings. Specifically, in terms of the SR metric, DualMind outperforms `IL-only` on 8 out of 10 tasks at 10-shot and on 7 tasks at 100- and 1000-shot demonstrations. Furthermore, with respect to the SPL and PES metrics, DualMind achieves better results than `IL-only` on 9 tasks in the 10-shot experiment. These results provide further evidence that the proposed Dual-phase training approach can enhance the generalization ability of models even when dealing with novel tasks and limited demonstrations.

5.3.3 Attention visualization

To gain insight into how `DualMind` is able to perform diverse tasks, we conduct attention visualization. We present attention maps for tasks from both Habitat and MetaWorld, where we display a sequence of frames from the episode for each task.

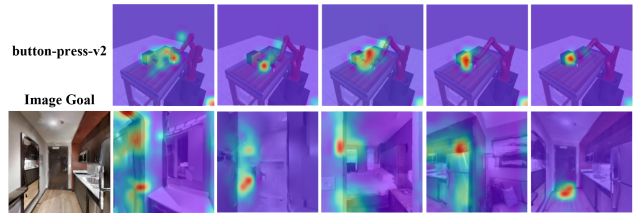

Figure 9: Attention map visualization.

The attention maps reveal that when performing manipulation tasks in MetaWorld, such as and "button-press-v2", the model initially focuses on the execution context and then shifts its attention to the targeting instance, such as the "button", until the task is completed. Notably, for navigation tasks in Habitat, `DualMind` learns to explore the scene to locate the goal. For example, as shown in Fig. 9, given an image goal, the agent first attends to the entrance to navigate into the restroom. Upon realizing that the goal is not there, it steps out and searches for another room to enter. After spotting the refrigerator, which appears in the image goal, the agent quickly locks onto the goal and completes the task. These attention maps provide insight into how `DualMind` leverages its generalization ability to solve new tasks.

5.4. Ablation study

Training parts in Phase II

In this section, we ablate DualMind by varying model weights that been trained in Phase II, as listed below:

- ①: Freeze the entire Enc-Dec Control Transformer arch by only train the cross-attention layers.
- ②: Freeze the Transformer Encoder (State tokenizer) and the first 4 layers of Transformer Decoder.
- ③: Freeze the Transformer Encoder.
- ④: No frozen part, optimize the entire model in Phase II.

As shown in Fig. 10, ② and ③ perform the best in most cases. For our experiments, we use ③. However, for future scaled-up models and data, we would recommend using ② since it saves more computational cost. When training each setting with the same number of iterations, ④ performs poorly, which may be due to slow convergence with more model weights. This result also suggests that after training in phase I, our model has learned useful information, but insufficient re-training in phase II may lead to performance deterioration due to potential forgetting issues.

Prompt conditioning

We conducted an ablation study on DualMind by comparing two prompt conditioning approaches: prefix and XAtten. prompting. We used the average success rate of

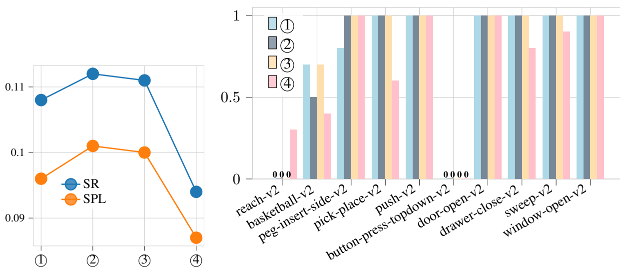

Figure 10: Comparisons of frozen parts in Phase II.

ML10 training tasks as the comparison metric for Metaworld. Results show that XAtten. prompting achieves a 0.76 SR on Metaworld and an 0.11 SR on Habitat, while prefix prompting only achieves 0.29 SR and 0 SR, respectively. The cross-attention mechanism in XAtten. prompting allows the agent to establish a strong connection between prompts and demonstrations, which is particularly useful for goal-conditioned tasks. (See more details and discussion in Appendix B.)

6. Conclusion

This paper presents a new training approach for generalist agents called DualMind, which consists of two phases: self-supervised learning of basic and generic knowledge across various tasks, followed by imitation of expert behaviors with different types of prompt conditioning. By utilizing a carefully designed Transformer Encoder-Decoder architecture and a dual-phase training scheme, DualMind is scalable, versatile, and generalizable. Empirical evaluation on two challenging domains, Habitat and MetaWorld, shows that DualMind outperforms previous generalist learning methods and pretraining approaches. Further analysis and ablations demonstrate the effectiveness of the dual-phase design.

Future work includes expanding DualMind to more domains and tasks, finding efficient solutions for handling longer context lengths in demonstrations, and enabling practical training in online interactive scenarios.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 1, 2

[2] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 6, 15

[3] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*, pages 348–367. Springer, 2022. 2, 13

[4] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022. 2

[5] Rogerio Bonatti, Sai Vemprala, Shuang Ma, Felipe Frujeri, Shuhang Chen, and Ashish Kapoor. Pact: Perception-action causal transformer for autoregressive robotics pre-training. *arXiv preprint arXiv:2209.11133*, 2022. 2

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 2

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. 2

[8] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 14

[9] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021. 2

[10] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 2

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. 1, 2

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner,

Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4, 12

[13] Golnaz Ghiasi, Barret Zoph, Ekin D. Cubuk, Quoc V. Le, and Tsung-Yi Lin. Multi-task self-training for learning general representations. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8836–8845, 2021. 2

[14] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. 6

[15] Meera Hahn, Devendra Singh Chaplot, Shubham Tulsiani, Mustafa Mukadam, James M Rehg, and Abhinav Gupta. No rl, no simulation: Learning to navigate without navigating. *Advances in Neural Information Processing Systems*, 34:26661–26673, 2021. 2

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1, 2

[17] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2022. 2, 17

[18] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*, page 491–507, Berlin, Heidelberg, 2020. Springer-Verlag. 2

[19] Kuang-Huei Lee, Ofir Nachum, Sherry Yang, Lisa Lee, C. Daniel Freeman, Sergio Guadarrama, Ian Fischer, Winnie Xu, Eric Jang, Henryk Michalewski, and Igor Mordatch. Multi-game decision transformers. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019*, 2019. 6, 13

[21] Shuang Ma, Sai Vemprala, Wenshan Wang, Jayesh K Gupta, Yale Song, Daniel McDufft, and Ashish Kapoor. Compass: Contrastive multimodal pretraining for autonomous systems. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1000–1007. IEEE, 2022. 2

[22] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[23] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229, 2019. 13

[24] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In Karen Liu, Dana Kulic, and Jeffrey Ichnowski, editors, *Conference on Robot Learning, CoRL 2022*, volume 205 of *Proceedings of Machine Learning Research*, pages 892–909. PMLR, 2022. 2

[25] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 3

[26] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *International Conference on Machine Learning*, pages 17359–17371. PMLR, 2022. 2

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 4, 15

[28] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 1, 2, 6

[29] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023. 2

[30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 2

[31] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 1, 2, 5, 14

[32] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022. 1, 2, 3, 4, 12, 14, 16

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10684–10695, 2022. 17

[34] Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12786–12797. Curran Associates, Inc., 2021. 2, 4, 6, 12

[35] Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 19561–19579. PMLR, 17–23 Jul 2022. 2

[36] Yanchao Sun, Shuang Ma, Ratnesh Madaan, Rogerio Bonatti, Furong Huang, and Ashish Kapoor. SMART: self-supervised multi-task pretraining with control transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. 1, 2, 3, 4, 5, 12

[37] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018. 2

[38] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 2

[39] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079, 2018. 14

[40] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020. 1, 2, 5, 15

[41] Qinqing Zheng, Amy Zhang, and Aditya Grover. Online decision transformer. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 27042–27059. PMLR, 17–23 Jul 2022. 2

[42] Qinqing Zheng, Amy Zhang, and Aditya Grover. Online decision transformer. In *International Conference on Machine Learning*, pages 27042–27059. PMLR, 2022. 2