Foundation models in robotics: Applications, challenges, and the future

Roya Firoozi[1], Johnathan Tucker[1], Stephen Tian[1],
Anirudha Majumdar[2,3], Jiankai Sun[1], Weiyu Liu[1], Yuke Zhu[4,5],
Shuran Song[1], Ashish Kapoor[6], Karol Hausman[1,3], Brian Ichter[3],
Danny Driess[3,7], Jiajun Wu[1], Cewu Lu[8] and Mac Schwager[1]

Abstract

We survey applications of pretrained foundation models in robotics. Traditional deep learning models in robotics are trained on small datasets tailored for specific tasks, which limits their adaptability across diverse applications. In contrast, foundation models pretrained on internet-scale data appear to have superior generalization capabilities, and in some instances display an emergent ability to find zero-shot solutions to problems that are not present in the training data. Foundation models may hold the potential to enhance various components of the robot autonomy stack, from perception to decision-making and control. For example, large language models can generate code or provide common sense reasoning, while vision-language models enable open-vocabulary visual recognition. However, significant open research challenges remain, particularly around the scarcity of robot-relevant training data, safety guarantees and uncertainty quantification, and real-time execution. In this survey, we study recent papers that have used or built foundation models to solve robotics problems. We explore how foundation models contribute to improving robot capabilities in the domains of perception, decision-making, and control. We discuss the challenges hindering the adoption of foundation models in robot autonomy and provide opportunities and potential pathways for future advancements. The GitHub project corresponding to this paper can be found here: https://github.com/robotics-survey/Awesome-Robotics-Foundation-Models.

1. Introduction

Foundation models are pretrained on extensive internet-scale data and can be fine-tuned for adaptation to a wide range of downstream tasks. Foundation models have demonstrated significant breakthroughs in vision and language processing; examples include BERT (Devlin et al., 2018), GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023), CLIP (Radford et al., 2021a), DALL-E (Ramesh et al., 2021a), and PaLM-E (Driess et al., 2023). Foundation models have the potential to unlock new possibilities in robotics domains such as autonomous driving, household robotics, industrial robotics, assistive robotics, medical robotics, field robotics, and multi-robot systems. Pretrained Large Language Models (LLMs), Large Vision-Language Models (VLMs), Large Audio-Language Models (ALMs), and Large Visual-Navigation Models (VNMs) can be utilized to improve various tasks in robotics settings. The integration of foundation models into robotics is a rapidly evolving area, and the robotics community has very recently started exploring ways to leverage these large models within the robotics domain for perception, prediction, planning, and control.

Prior to the emergence of foundation models, traditional deep learning models for robotics were typically trained on limited datasets gathered for distinct tasks (Sun et al., 2022a). Conversely, foundation models are pre-trained on extensive and diverse data, which has been proven in other domains

[1]Stanford University, Stanford, CA, USA
[2]Princeton University, Princeton, NJ, USA
[3]Google DeepMind, Mountain View, CA, USA
[4]UT Austin, Austin, TX, USA
[5]NVIDIA, Santa Clara, CA, USA
[6]Scaled Foundations, Kirkland, WA, USA
[7]TU Berlin, Berlin, Germany
[8]Shanghai Jiao Tong University, Shanghai, China

Corresponding author:
Roya Firoozi, Stanford University, 496 Lomita Mall, Stanford, CA 94305, USA.
Email: rfiroozi@stanford.edu

(such as natural language processing, computer vision, and healthcare) to significantly expand adaptability, generalization capability, and overall performance (Qiu et al., 2023). Ultimately, foundation models may hold the potential to yield these same benefits in robotics. Knowledge transfer from foundation models may reduce training time and computational resources compared to task-specific models. Particularly relevant to robotics, multimodal foundation models can fuse and align multimodal heterogeneous data gathered from various sensors into compact homogeneous representations needed for robot understanding and reasoning (Sun et al., 2023e). These learned representations hold the potential to be used in any part of the autonomy stack, including perception, decision-making, and control. Furthermore, foundation models provide zero-shot capabilities, which refer to the ability of an AI system to perform tasks without prior examples or dedicated training data for that specific task. The would enable robots to generalize their learned knowledge to novel cases, enhancing adaptability and flexibility for robots in unstructured settings.

Integrating foundation models into robotic systems may enable context-aware robotic systems by enhancing the robot's ability to perceive and interact with the environment. For example, in the perception domain, Large Vision-Language Models (VLMs) have been found to provide cross-modal understanding by learning associations between visual and textual data, aiding tasks such as zero-shot image classification, zero-shot object detection (Zhang et al., 2023b), and 3D classification (Hong et al., 2023). As another example, language grounding in the 3D world (Chen et al., 2022a) (aligning contextual understanding of VLMs to the 3-dimensional (3D) real world) may enhance a robot's spatial awareness by associating words with specific objects, locations, or actions within the 3D environment.

In the decision-making or planning domain, LLMs and VLMs have been found to assist robots in task specification for high-level planning (Yang et al., 2023b). Robots can perform more complex tasks by leveraging linguistic cues in manipulation, navigation, and interaction. For example, for robot policy learning techniques like imitation learning (Mandi et al., 2022) and reinforcement learning (Palo et al., 2023), foundation models seem to offer the possibility to improve data efficiency and enhance contextual understanding. In particular, language-driven rewards can be used to guide RL agents by providing shaped rewards (Kwon et al., 2023). Also, researchers have employed language models to provide feedback for policy learning techniques (Feng et al., 2023). Some works have shown that a VLM model's visual question-answering (VQA) capability can be harnessed in robotics use cases. For example, researchers have used VLMs to answer questions related to visual content to aid robots in accomplishing their tasks (Du et al., 2023a). Also, researchers have stated utilizing VLMs to help with data annotation, by generating descriptive labels for visual content (He et al., 2023).

Despite the transformative capabilities of foundation models in vision and language processing, the generalization and fine-tuning of foundation models for real-world robotics tasks remain challenging. These challenges include: (1) **Data Scarcity:** how to obtain internet-scale data for robot manipulation, locomotion, navigation, and other robotics tasks, and how to perform self-supervised training with this data, (2) **High Variability:** how to deal with the large diversity in physical environments, physical robot platforms, and potential robot tasks while still maintaining the generality required for a foundation model, (3) **Uncertainty Quantification:** how to deal with (i) instance-level uncertainty such as language ambiguity or LLM hallucination; (ii) distribution-level uncertainty; and (iii) distribution-shift, especially resulting from closed-loop robot deployment, (4) **Safety Evaluation:** How to rigorously test for the safety of a foundation model-based robotic system (i) prior to deployment, (ii) as the model is updated throughout its lifecycle, and (iii) as the robot operates in its target environments, (5) **Real-Time Performance:** how to deal with the high inference time of some foundation models, which could hinder their deployment on robots, and how to accelerate inference in foundation models to the speed required for online decision-making, and (6) **Reproducibility:** how to reproduce research and benchmark robotic-specific foundation models developed on particular hardware setups.

In this survey, we study the existing literature on the use of foundation models in robotics. We study current approaches and applications, present current challenges, suggest directions for future research to address these challenges, and identify potential risks exposed by integrating foundation models into robot autonomy. Another survey on foundation models in robotics appeared simultaneously with ours on arXiv (Xiao et al., 2023b). In comparison with that paper, ours emphasizes future challenges and opportunities, including safety and risk, and ours has a stronger emphasis on comparisons in applications, algorithms, and architectures among the existing papers in this space. In contrast to some existing surveys that focus on a specific in-context instruction, such as prompts (Liu et al., 2023b), vision transformers (Khan et al., 2022), or decision-making (Wen et al., 2023; Yang et al., 2023b;), we provide a broader perspective to connect distinct research threads in foundation models organized around their relevance to and application to robotics. Conversely, our scope is much narrower than the paper (Bommasani et al., 2021), which explores the broad application of foundation models across many disciplines, of which robotics is one. In another concurrent work (Hu et al., 2023b), the authors survey general-purpose robotics using foundation models. Beginning with discussions on classical robotics, including perception, planning, and control, this work addresses the primary limitations of conventional robotics through the use of foundation models. Compared to this work, our survey excludes classical methods and instead offers a more comprehensive discussion on various components of the autonomy stack, including in-depth examinations of open-vocabulary perception and embodied AI. Additionally, our survey provides detailed discussions on challenges and future opportunities, such as language ambiguity, real-time inference, and safety evaluations. We hope this paper can

provide clarity regarding areas of recent progress and existing deficiencies in the research, and point the way forward to future opportunities and challenges facing this research area. Ultimately, we aim to give a resource for robotics researchers to learn about this exciting new area.

We limit the scope of this survey to papers that fall into one of the following categories:

1. **Background Papers:** Papers that do not explicitly link to robotics, but are nonetheless required for understanding foundation models. These papers are discussed in the background section (Section 2) of the survey paper.
2. **Robotics Papers:** Papers that integrate a foundation model into a robotic system in a plug-and-play fashion, papers that adapt or fine-tune foundation models for robotic systems, or papers that build new robotic-specific foundation models.
3. **Robotics-Adjacent Papers:** Papers that present methods or techniques applied to areas adjacent to robotics (e.g., computer vision, embodied AI), with a clear path to future application in robotics.

This survey is organized as follows: In Section 2, we provide an introduction to foundation models, including LLMs, vision transformers, VLMs, embodied multimodal language models, and visual generative models. In addition, in the last part of this section, we discuss different training methods used to train foundation models. In Section 3, we present a review of how foundation models are integrated into different tasks for decision-making in robotics. First, we discuss robot policy learning using language-conditioned imitation learning, and language-assisted reinforcement learning. Then, we discuss how to use foundation models to design a language-conditioned value function that can be used for planning purposes. Next, robot task specification and code generation for task planning using foundation models are presented. In Section 4, we study various perception tasks in robotics that have the potential to be enhanced by employing foundation models. These tasks include semantic segmentation, 3D scene representation, zero-shot 3D classification, affordance prediction, and dynamics prediction. In Section 5, we present papers about Embodied AI agents, generalist AI agents, as well as simulators and benchmarks developed for embodied AI research. In Section 6, we conclude the survey by discussing different challenges for employing foundation models in robotic systems and proposing potential avenues for future research. Finally, in Section 7, we offer the concluding remarks. Figure 1 summarizes some applications of foundation models in robotics.
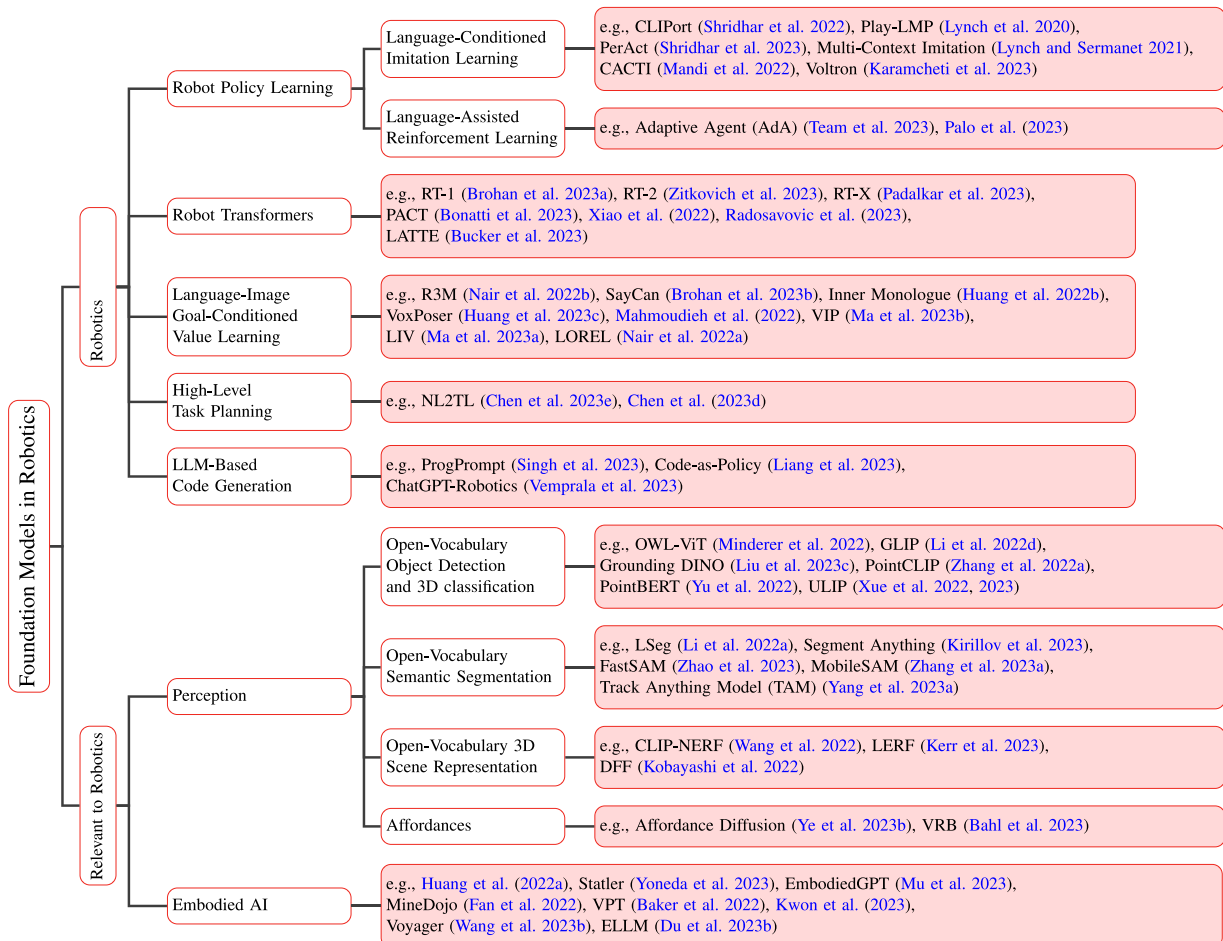

Figure 1. Overview of robotics tasks leveraging foundation models.

2. Foundation models background

Foundation models have billions of parameters and are pretrained on massive internet-scale datasets. Training models of such scale and complexity involve substantial costs. Acquiring, processing, and managing data can be costly. The training process demands significant computational resources, requiring specialized hardware such as GPUs or TPUs, as well as software and infrastructure for model training which requires financial resources. Additionally, training a foundation model is time-intensive, which can translate to even higher costs. Hence these models are often used as plug-and-play modules (which refers to the integration of foundation models into various applications without the need for extensive customization). Table 1 provides details about commonly used foundation models. In the rest of this section, we introduce LLMs, vision transformers, VLMs, embodied multi-modal language models, and visual generative models. In the last part of this section, we introduce different training methods that are used to train foundation models.

2.1. Terminology and mathematical preliminaries

In this section, we first introduce common terminologies in the context of foundation models and describe basic mathematical details and training practices for various types of foundation models.

2.1.1. Tokenization. Given a sequence of characters, tokenization is the process of dividing the sequence into smaller units, called tokens. Depending on the tokenization strategy, tokens can be characters, segments of words, complete words, or portions of sentences. Tokens are represented as 1-hot vectors of dimension equal to the size of the total vocabulary and are mapped to lower-dimensional vectors of real numbers through a learned embedding matrix. An LLM takes a sequence of these embedding vectors as raw input, producing a sequence of embedding vectors as raw output. These output vectors are then mapped back to tokens and hence to text. GPT-3, for example, has a vocabulary of 50,257 different tokens, and an embedding dimension of 12,288.

Table 1. Large Pretrained Models.

Model	Architecture	Size	Training Data	What to Pretrain	How to Pretrain	Hardware
CLIP (Radford et al., 2021a)	ViT-L/14@336px and a text encoder (Radford et al., 2019)	0.307B	400M image-text pairs	Zero-shot image classification	Contrastive pre-training	Fine-tuned CLIP model is trained for 12 days on 256 V100 GPUs
GPT-3 (Brown et al., 2020)	Transformer (slight modification of GPT-2)	175B	Common Crawl (about a trillion words)	Text output	Autoregressive model	NPA[a]
GPT-4 (OpenAI, 2023)	NPA	NPA	NPA	Text output	NPA	NPA
PaLI-X (Chen et al., 2023c)	Encoder-decoder	55B	10B image-text pairs from WebLI (Chen et al., 2022b) and auxiliary tasks	Text and image to text output	Autoregressive model	Runs on multi-TPU cloud service
DALL-E (Ramesh et al., 2021b)	Decoder-only transformer	12B	250 M text-image pairs	Zero-shot text-to-image generation	Autoregressive model	NPA
DALL-E2 (Ramesh et al., 2022)	A prior based on CLIP+ a decoder	3.5 B	CLIP and DALL-E (Ramesh et al., 2021b)	Zero-shot text-to-image generation	Diffusion	NPA
DINOv2 (Oquab et al., 2023)	ViT-g/14	1.1B	LVD-142M (Oquab et al., 2023)	Visual-features (image-level and pixel-level)	Discriminative	20 nodes equipped with 8 V100-32GB GPUs
SAM (Kirillov et al., 2023)	MAE (He et al., 2022) vision transformer + CLIP (Radford et al., 2021b) text encoder	632M for ViT-H + 63M for CLIP text encoder	SA-1B dataset (Kirillov et al., 2023) that includes 1.1B segmentation masks on 11M images	Zero-shot promptable image segmentation	Supervised learning	256 A100 GPUs for 68 hours

[a]NPA stands for not publicly available.

The token decoding (from low-dimension real-valued embedding vectors to high-dimension 1-hot vectors) is not deterministic, resulting in a weighting for each possible token in the vocabulary. These weightings are often used by LLMs as probabilities over tokens, to introduce randomness in the text generation process. For example, the temperature parameter in GPT-3 blends between always choosing the top-weighted token (temperature of 0) and drawing the token based on the probability distribution suggested by the weights (temperature of 1). This source of randomness is only in the token decoding process, not in the LLM itself. To the authors' knowledge, this is, in fact, the only source of randomness in the GPT family of models.

One of the most common tokenization schemes, which is used by the GPT family of models, is called byte-pair encoding (Sennrich et al., 2016). Byte-pair encoding starts with a token for each individual symbol (e.g., letter, punctuation), then recursively builds tokens by grouping pairs of symbols that commonly appear together, building up to assign tokens to larger and larger groups (pairs of pairs, etc.) that frequently appear together in a text corpus. The tokenization process can extend beyond text data to diverse contexts, encompassing various data modalities like images, videos, and robot actions. In these scenarios, the respective data modalities can be treated as sequential data and tokenized similarly to train generative models. For example, just as language constitutes a sequence of words, an image comprises a sequence of image patches, force sensors yield a sequence of sensory inputs at each time step, and a series of actions represent the sequential nature of tasks for a robot.

2.1.2. Generative Models.
A generative model is a model that learns to sample from a probability distribution to create examples of data that seem to be from the same distribution as the training data. For example, a face generation model can produce images of faces that cannot be distinguished from the set of real images used to train the model. These models can be trained to be conditional, meaning they generate samples from a conditional distribution conditioned on a wide range of possible conditioning information. For example, a gender conditional face generator can generate images of female or male faces, where the desired gender is given as a conditioning input to the model.

2.1.3. Discriminative Models.
Discriminative models are used for regression or classification tasks. In contrast to generative models, discriminative models are trained to distinguish between different classes or categories. Their emphasis lies in learning the boundaries between classes within the input space. While generative models learn to sample from the distribution over the data, discriminative models learn to evaluate the probability distribution of the output labels given the input features, or (depending on how the model is trained) learn to evaluate some statistic of the probability distribution over the outputs, such as the expected output given an input.

2.1.4. Transformer architecture.
Most foundation models are built on the transformer architecture, which has been instrumental in the rise of foundation models and large language models. The following discussion was synthesized from Vaswani et al. (2017), as well as online blogs, unpublished reports, and Wikipedia (Dugas, 2023; Thickstun, 2023; Wikipedia, 2023). A transformer acts simultaneously on a collection of embedded token vectors $(x_1, \ldots, x_N)$ known as a context window. The key enabling innovation of the Transformer architecture is the multi-head self-attention mechanism originally proposed in the seminal work (Vaswani et al., 2017). In this architecture, each attention head computes a vector of importance weights that corresponds to how strongly a token in the context window x_i correlates with other tokens in the same window x_j. Each attention head mathematically encodes different notions of similarity, through different projection matrices used in the computation of the importance weights. Each head can be trained (backward pass) and evaluated (forward pass) in parallel across all tokens and across all heads, leading to faster training and inference when compared with previous models based on RNNs or LSTMs.

Mathematically, an attention head maps each token x_i in the context window to a "query" $q_i = W_q^T x_i$, and each other token in the context head x_j to a "key" $k_j = W_k^T x_j$. The similarity between query and key is then measured through a scaled dot product, $q_i^T k_j / \sqrt{d}$, where d is the dimension of the query and key vectors. A softmax is then taken over all j to give weights α_{ij} representing how much x_i "attends to" x_j. The tokens are then mapped to "values" with $v_j = W_v^T x_j$, and the output of the attention for position i is then given as a sum over values weighted by attention weights, $\sum_j \alpha_{ij} v_j$. One of the key reasons for the success of the transformer attention model is that it can be efficiently computed with GPUs and TPUs by parallelizing the preceding steps into matrix computations,

$$\text{attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \qquad (1)$$

where $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are matrices with rows q_i^T, k_i^T, and v_i^T, respectively. Each head in the model produces this computation independently, with different W_q, W_k, W_v matrices to encode different kinds of attention. The outputs from each head are then concatenated, normalized with a skip connection, passed through a fully connected ReLU layer, and normalized again with a skip connection to produce the output of the attention layer. Multiple layers are arranged in various ways to give "encoders" and "decoders," which together make up a transformer.

The size of a transformer model is typically quantified by (i) the size of the context window, (ii) the number of attention heads per layer, (iii) the size of the attention vectors in each head, and (iii) the number of stacked attention layers. For example, GPT-3's context window is 2048 tokens (corresponding to about 1500 words of text), each attention layer has 96 heads, each head has attention vectors

of 128 dimensions, and there are 96 stacked attention layers in the model.

The basic multi-head attention mechanism does not impose any inherent sense of sequence or directionality in the data. However, transformers—especially in natural language applications—are often used as sequence predictors by imposing a positional encoding on the input token sequence. They are then applied to a token sequence autoregressively, meaning they predict the next token in the sequence, add that token to their context window, and repeat. This concept is elaborated below.

2.1.5. Autoregressive models. The concept of autoregression has been applied in many fields as a representation of random processes whose outputs depend causally on the previous outputs. Autoregressive models use a window of past data to predict the next data point in a sequence. The window then slides one position forward, recursively ingesting the predicted data point into the window and expelling the oldest data point from the window. The model again predicts the next data point in the sequence, repeating this process indefinitely. Classical linear autoregressive models such as Auto-Regressive Moving Average (ARMA) and Auto-Regressive Moving Average with eXogenous input (ARMAX) models are standard statistical tools dating back to at least the 1970s (Box et al., 2015). These modeling concepts were adapted to deep learning models first with RNNs, and later LSTMs, which are both types of learnable nonlinear autoregressive models. Transformer models, although they are not inherently autoregressive, are often adapted to an autoregressive framework for text prediction tasks.

For example, the GPT family (Radford et al., 2018) builds on the original transformer model by using a modification introduced in Liu et al. (2018) that removes the transformer encoder blocks entirely, retaining just the transformer decoder blocks. This has the advantage of reducing the number of model parameters by close to half while reducing redundant information that is learned in both the encoder and decoder. During training, the GPT model seeks to produce an output token from the tokenized corpus $\mathcal{X} = (x_1, \ldots, x_n)$ to minimize the negative log-likelihood within the context window of length N,

$$\mathcal{L}_{\text{LLM}} = -\sum_i \log P(x_i \mid x_{i-N}, \ldots, x_{i-1}). \qquad (2)$$

This results in a large pretrained model that autoregressively predicts the next likely token given the tokens in the context window. Although powerful, the unidirectional autoregressive nature of the GPT family means that these models may lag in performance on bidirectional tasks such as reading comprehension.

2.1.6. Masked Auto-Encoding. To address the unidirectional limitation of the GPT family and allow the model to make bidirectional predictions, works such as BERT (Devlin et al., 2018) use masked auto-encoding. This is achieved through an architectural change, namely the addition of a bidirectional encoder, as well as a novel pretraining objective known as masked language modeling (MLM). The MLM task simply masks a percentage of the tokens in the corpus and requires the model to predict these tokens. Through this procedure, the model is encouraged to learn the context that surrounds a word rather than just the next likely word in a sequence.

2.1.7. Contrastive learning. Visual-language foundation models such as CLIP (Radford et al., 2021a) typically rely on different training methods from the ones used with large language models, which encourage explicitly predictive behavior. Visual-language models use contrastive representation learning, where the goal is to learn a joint embedding space between input modalities where similar sample pairs are closer than dissimilar ones. The training objective for many VLMs is some variation of the objective function,

$$\ell_i^{(v \to u)} = -\log \frac{\exp(\text{sim}(\mathbf{v}_i, \mathbf{u}_i)/\tau)}{\sum_{k=1}^{N} \exp(\text{sim}(\mathbf{v}_i, \mathbf{u}_k)/\tau)}, \qquad (3)$$

$$\ell_i^{(u \to v)} = -\log \frac{\exp(\text{sim}(\mathbf{u}_i, \mathbf{v}_i)/\tau)}{\sum_{k=1}^{N} \exp(\text{sim}(\mathbf{u}_i, \mathbf{v}_k)/\tau)}, \qquad (4)$$

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(\lambda \ell_i^{(v \to u)} + (1 - \lambda)\ell_i^{(u \to v)} \right). \qquad (5)$$

This objective function was popularized for multimodal input by ConVIRT (Zhang et al., 2022b) and first presented in prior works (Chen et al., 2020; Oord et al., 2018; Sohn, 2016; Wu et al., 2018). This objective function trains the image and text encoders to preserve mutual information between the true text and image pairs. In these equations, $\mathbf{u}_i$ and $\mathbf{v}_i$ are the ith encoded text and image respectively from $i \in 1, \ldots, N$ image and text pairs. The *sim* operation is the cosine similarity between the text and image embeddings, and τ is a temperature term. In CLIP (Radford et al., 2021a) the authors use a symmetric cross-entropy loss, meaning the final loss is an average of the two loss components where each is equally weighted (i.e., $\lambda = 0.5$).

2.1.8. Diffusion models. Outside of large language models and multi-modal models such as VLMs, diffusion models for image generation (e.g., DALL-E2) (Ramesh et al., 2022) are another class of foundation models considered in this survey. Although diffusion models were established in prior work (Sohl-Dickstein et al., 2015; Song and Ermon, 2019) the diffusion probabilistic model presented in Ho et al. (2020) popularized the method. The diffusion probabilistic model is a deep generative model that is trained in an iterative forward and reverse process. The forward process adds Gaussian noise to an input x_0 in a Markov chain until x_T when the

result is zero mean isotropic noise. This means the forward process produces a trajectory of noise $q(x_{1:T} \mid x_0)$ as,

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}). \tag{6}$$

At each time step $q(x_t|x_{t-1})$ is described by a normal distribution with mean $\sqrt{1-\beta_t}\mathbf{x}_{t-1}$ and covariance $\beta_t\mathbf{I}$ where β_t is scheduled or a fixed hyperparameter.

The reverse process requires the model to learn to the transitions that will de-noise the zero-mean Gaussian and produce the input image. This process is also defined as a Markov chain where the transition distribution at time t is $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$. For completeness, the reverse process Markov chain is given by,

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t). \tag{7}$$

Diffusion models are trained using a reduced form of the evidence lower bound loss function that is typical of variational generative models like variational autoencoders (VAEs). The reduced loss function used for training is

$$\begin{aligned} \mathcal{L} = \mathbb{E}_q \Big[&D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T)) \\ &+ \sum_{t>1} D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)) \Big], \end{aligned} \tag{8}$$

where $D_{\mathrm{KL}}(q\|p)$ denotes Kullback–Leibler divergence, which is a measure of how different a distribution q is from a distribution p.

2.2. Large Language Model (LLM) examples and historical context

Large Language Models (LLMs) have billions of parameters and are trained on trillions of tokens. This large scale has allowed models such as GPT-2 (Radford et al., 2019) and BERT (Devlin et al., 2018) to achieve state-of-the-art performance in the Winograd Schema challenge (Levesque et al., 2012) and the General Language Understanding Evaluation (GLUE) (Wang et al., 2018) benchmarks, respectively. Their successors include GPT-3 (Brown et al., 2020), LLaMA (Touvron et al., 2023a), and PaLM (Chowdhery et al., 2022) has grown considerably in the number of parameters (typically now over 100 billion), the size of the context window (typically now over 1000 tokens), and the size of the training data set (typically now 10s of terabytes of text). GPT-3 is trained on the Common Crawl dataset. Common Crawl contains petabytes of publicly available data over 12 years of web crawling and includes raw web page data, metadata, and text extracts. LLMs can also be multi-lingual. For example, GLM-130B (Zeng et al., 2023a) is a bilingual (English and Chinese) pretrained language model with 130 billion parameters. LLMs can also be fine-tuned, a process by which the model parameters are adjusted with domain-specific data to align the performance of the LLM to a specific use case. For example, GPT-3 and GPT-4 (OpenAI, 2023) have been fine-tuned using reinforcement learning with human feedback (RLHF).

2.3. Vision transformers

A Vision Transformer (ViT) (Dosovitskiy et al., 2021; Han et al., 2022; Khan et al., 2022) is a transformer architecture for computer vision tasks including image classification segmentation, and object detection. A ViT treats an image as a sequence of image patches referred to as tokens. In the image tokenization process, an image is divided into patches of fixed size. Then, the patches are flattened into a one-dimensional vector, which is referred to as linear embedding. To capture the spatial relationships between image patches, positional information is added to each token. This process is referred to as position embedding. The image tokens incorporated with position encoding are fed into the transformer encoder, and the self-attention mechanism enables the model to capture long-term dependencies and global patterns in the input data. In this paper, we focus only on those ViT models with a large number of parameters. ViT-G (Zhai et al., 2022) scales up the ViT model and has 2B parameters. Additionally, ViT-e (Chen et al., 2022b) has 4B parameters. ViT-22 B (Dehghani et al., 2023) is a vision transformer model at 22 billion parameters, which is used in PaLM-E and PaLI-X (Chen et al., 2023c) and helps with robotics tasks.

DINO (Caron et al., 2021) is a self-supervised learning method, for training ViT. DINO is a form of knowledge distillation with no labels. Knowledge distillation is a learning framework where a smaller model (student network) is trained to mimic the behavior of a larger more complex model (teacher network). Both networks share the same architecture with different sets of parameters. Given a fixed teacher network, the student network learns its parameters by minimizing the cross-entropy loss w.r.t. the student network parameters. The neural network architecture is composed of ViT or ResNet (He et al., 2015) backbone and a projection head that includes layers of multi-layer perception (MLP). Self-supervised ViT features learned using DINO contain explicit information about the semantic segmentation of an image, including scene layout and object boundaries, with such clarity that is not achieved using supervised ViTs or ConvNets.

DINOv2 (Oquab et al., 2023) provides a variety of pretrained visual models that are trained with different vision transformers (ViT) on the LVD-142M dataset introduced in Oquab et al. (2023). It is trained using a discriminative self-supervised method on a compute cluster of 20 nodes equipped with 8 V100-32 GB GPUs. DINOv2 provides various visual features at the image (e.g., detection) or pixel level (e.g., segmentation). SAM (Kirillov

et al., 2023) provides zero-shot promptable image segmentation. It is discussed in more detail in Section 4.

2.4. Multimodal vision-language models (VLMs)

Multimodal refers to the ability of a model to accept different "modalities" of inputs, for example, images, texts, or audio signals. Visual-language models (VLM) are a type of multi-modal model that takes in both images and text. A commonly used VLM in robotics applications is Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021a). CLIP offers a method to compare the similarity between textual descriptions and images. CLIP uses internet-scale image-text pairs data to capture the semantic information between images and text. CLIP model architecture contains a text encoder (Radford et al., 2019) and an image encoder (a modified version of vision transformer ViT) that are trained jointly to maximize the cosine similarity of the image and text embeddings. CLIP uses contrastive learning together with language models and visual feature encoders to incorporate models for zero-shot image classification.

BLIP (Li et al., 2022c) focuses on multimodal learning by jointly optimizing three objectives during pretraining. These objectives include Image-Text Contrastive Loss, Image-Text Matching Loss, and Language Modeling Loss. The method leverages noisy web data by bootstrapping captions, enhancing the training process. CLIP^2 (Zeng et al., 2023b) aims to build well-aligned and instance-based text-image-point proxies. It learns semantic and instance-level aligned point cloud representations using a cross-modal contrastive objective. FILIP (Yao et al., 2022) focuses on achieving finer-level alignment in multimodal learning. It incorporates a cross-modal late interaction mechanism that utilizes token-wise maximum similarity between visual and textual tokens. This mechanism guides the contrastive objective and improves the alignment between visual and textual information. FLIP (Li et al., 2023) proposes a simple and more efficient training method for CLIP. FLIP randomly masks out and removes a significant portion of image patches during training. This approach aims to improve the training efficiency of CLIP while maintaining its performance.

2.5. Embodied multimodal language models

An embodied agent is an AI system that interacts with a virtual or physical world. Examples include virtual assistance or robots. Embodied language models are foundation models that incorporate real-world sensor and actuation modalities into pretrained large language models. Typical vision-language models are trained on general vision-language tasks such as image captioning or visual question answering. PaLM-E (Driess et al., 2023) is a multimodal language model that has been trained on not only internet-scale general vision-language data, but also on embodied, robotics data, simultaneously. In order to connect the model to real-world sensor modalities, PaLM-E's architecture injects (continuous) inputs such as images, low-level states, or 3D neural scene representations into the language embedding space of a decoder-only language model to enable the model to reason about text and other modalities jointly. The main PaLM-E version is built from the PaLM LLM (Chowdhery et al., 2022) and a ViT (Dehghani et al., 2023). The ViT transforms an image into a sequence of embedding vectors, which are projected into the language embedding space via an affine transformation. The whole model is trained end-to-end, starting from a pre-trained LLM and ViT model. The authors also explore different strategies, such as freezing the LLM and just training the ViT, which leads to worse performance. Given multimodal inputs, the output of PaLM-E is text decoded auto-regressively. In order to connect this output to a robot for control, language conditioned short-horizon policies can be used. In this case, PaLM-E acts as a high-level control policy. Experiments show that a single PaLM-E, in addition to being a vision-language generalist, is able to perform many different robotics tasks over multiple robot embodiments. The model exhibits positive transfer; that is, simultaneously training on internet-scale language, general vision-language, and embodied domains leads to higher performance compared to training the model on single tasks.

2.6. Visual generative models

Web-scale diffusion models such as OpenAI's DALL-E (Ramesh et al., 2021b) and DALL-E2 (Ramesh et al., 2022) provide zero-shot text-to-image generation. They are trained on hundreds of millions of image-caption pairs from the internet. These models learn a language-conditioned distribution over images from which an image can be generated using a given prompt. The DALL-E2 architecture includes a prior that generates a CLIP image embedding from a text caption, and a decoder that generates an image conditioned on the image embedding.

3. Robotics

In this section, we delve into robot decision-making, planning, and control. Within this realm, Large Language Models (LLMs) and Visual Language Models (VLMs) may hold the potential to serve as valuable tools for enhancing robotic capabilities. For instance, LLMs may facilitate the process of task specification, allowing robots to receive and interpret high-level instructions from humans. VLMs may also promise contributions to this field. VLMs specialize in the analysis of visual data. This visual understanding is a critical component of informed decision-making and complex task execution for robots. Robots can now leverage natural language cues to enhance their performance in tasks involving manipulation, navigation, and interaction. Vision-language goal-conditioned policy learning, whether through imitation learning or reinforcement learning, holds promise for improvement using foundation models. Language

models also play a role in offering feedback for policy learning techniques. This feedback loop fosters continual improvement in robotic decision-making, as robots can refine their actions based on the feedback received from an LLM. This section underscores the potential contributions of LLMs and VLMs in robot decision-making. Assessing and comparing the contributions of papers in this section presents greater challenges compared to the other sections like the Perception Section (4) or the Embodied AI Section (5). This is due to the fact that most papers in this section either rely on hardware experiments, using custom elements in the low-level control and planning stack that are not easily transferred to other hardware or other experimental setups, or they utilize non-physics-based simulators, which allow these low-level parts of the stack to be ignored, but leaving open the issue of non-transferability between different hardware implementations. In Section 6, we discuss the lack of benchmarking and reproducibility that needs to be addressed in future research.

3.1. Robot policy learning for decision making and control

In this section, we discuss robot policy learning, including language-conditioned imitation learning and language-assisted reinforcement learning.

3.1.1. Language-conditioned imitation learning for manipulation.
In language-conditioned imitation learning, a goal-conditioned policy $\pi_\theta(a_t|s_t, l)$ is learned that outputs actions $a_t \in \mathcal{A}$ conditioned on the current state $s_t \in \mathcal{S}$ and language instruction $l \in \mathcal{L}$. The loss function is defined as the maximum likelihood goal-conditioned imitation objective:

$$\mathcal{L}_{\text{GCIL}} = \mathbb{E}_{(\tau, l) \sim \mathcal{D}} \sum_{t=0}^{|\tau|} \log \pi_\theta(a_t|s_t, l), \qquad (9)$$

where $\mathcal{D}$ is the language-annotated demonstration dataset $\mathcal{D} = \{\tau_i\}_i^N$. Demonstrations can be represented as trajectories, or sequences of images, RGB-D voxel observations, etc. Language instructions are paired with demonstrations to be used as the training dataset. Each language-annotated demonstration τ_i consists of $\tau_i = \{(s_1, l_1, a_1), (s_2, l_2, a_2), \ldots\}$. At test time, the robot is given a series of instructions, and the language-conditioned visuomotor policy π_θ provides actions a_t in a closed loop given the instruction at each time step. The main challenges in this domain are: (i) obtaining a sufficient volume of demonstrations and conditioning labels to train a policy and (ii) distribution shift under the closed-loop policy—the feedback of the policy can lead the robot into regions of the state space that are not well-covered in the training data, negatively impacting performance. (All the following papers in this subsection focus on robot manipulation tasks).

Since generating language-annotated data by pairing demonstrations with language instruction is an expensive process, the authors in Play-LMP (Lynch et al., 2020) propose learning from teleoperated play data. In this setting, reusable latent plan representations are learned from unlabeled play data. Also, a goal-conditioned policy is learned to decode the inferred plan to perform the task specified by the user. In addition, the distributional shift in imitation learning is analyzed and it is shown in this setting that the play data is more robust with respect to perturbation compared to expert positive demonstrations. Note that language goal l in (9) can be substituted with any other type of goal for example goal image, which is another common choice of goal in goal-conditioned imitation learning.

In a follow-up work (Lynch and Sermanet, 2021), the authors present multi-context imitation (MCIL), which uses language-conditioned imitation learning over unstructured data. The multi-Context imitation framework is based on relabeled imitation learning and labeled instruction following. MCIL assumes access to multiple contextual imitation datasets, for example, goal image demonstrations, language goal demonstrations, or one-hot task demonstrations. MCIL trains a single latent goal-conditioned policy over all datasets simultaneously by encoding contexts in the shared latent space using the associated encoder for each context. Then a goal-conditioned imitation loss is computed by averaging over all datasets. The policy and goal-encoders are trained end-to-end.

Another approach to tackle the data annotation challenge in language-conditioned imitation learning involves utilizing foundation models to offer feedback by labeling demonstrations. In Yuying et al. (2023), the authors propose to use pretrained foundation models to provide feedback. To deploy a trained policy to a new task or new environment, the policy is played using randomly generated instructions, and a pretrained foundation model provides feedback by labeling the demonstration. Also, this paired instruction-demonstration data can be used for policy fine-tuning. CLIPort (Shridhar et al., 2022) also presents a language-conditioned imitation learning for vision-based manipulation. A two-stream architecture is presented that combines the semantic understanding of CLIP with the spatial precision of Transporter (Zeng et al., 2020). This end-to-end framework solves language-specified manipulation tasks without any explicit representation of the object poses or instance segmentation. CLIPort grounds semantic concepts in precise spatial reasoning, but it is limited to 2D observation and action spaces.

To address this limitation, the authors of PerAct (Perceiver-Actor) (Shridhar et al., 2023) propose to represent observation and action spaces with 3D voxels and employ the 3D structure of voxel patches for efficient language-conditioned behavioral cloning with transformers to imitate 6-DoF manipulation tasks from just a few demonstrations. While 2D behavioral cloning methods such as CLIPort are limited to single-view observations, 3D approaches such as PerAct allow for multi-view observations as well as 6-DoF action spaces. PerAct uses only CLIP's language encoder to encode the language goal. PerAct takes language goals and RGB-D voxel observations

as inputs to a Perceiver Transformer and outputs discretized actions by detecting the next best voxel action. PerAct is trained through supervised learning with discrete-time input actions from the demonstration dataset. The demonstration dataset includes voxel observations paired with language goals and keyframe action sequences. An action consists of a 6-DoF pose, gripper open state, and collision avoidance action. During training, a tuple is randomly sampled and the agent predicts the keyframe action given the observation and goal.

Grounding semantic representations into a spatial environment is essential for effective robot interaction. CLIPort and PerAct utilize CLIP (which is trained based on contrastive learning) for semantic reasoning and Transporter and Perceiver for spatial reasoning. Voltron (Karamcheti et al., 2023) presents a framework for language-driven representation learning in robotics. Voltron captures semantic, spatial, and temporal representations that are learned from videos and captions. Contrastive learning captures semantic representations but loses spatial relationships, and in contrast, masked autoencoding captures spatial and not semantic representations. Voltron trades off language-conditioned visual reconstruction for local spatial representations and visually grounded language generation to capture semantic representations. This framework includes grasp affordance prediction, single-task visuomotor control, referring expression grounding, language-conditioned imitation, and intent-scoring tasks. Voltron models take videos and their associated language captions as input to a multimodal encoder whose outputs are then decoded to reconstruct one or more frames from a masked context. Voltron starts with a masked autoencoding backbone and adds a dynamic component to the model by conditioning the MAE encoder on a language prefix. Temporal information is captured by conditioning on multiple frames.

Deploying robot policy learning techniques that leverage language-conditioned imitation learning with real robots presents ongoing challenges. These models rely on end-to-end learning, where the policy maps pixels or voxels to actions. As they are trained through supervised learning on demonstration datasets, they are susceptible to issues related to generalization and distribution shifts. To improve robustness and adaptability, techniques such as data augmentation and domain adaptation can make the policies more robust to the distribution shift. CACTI (Mandi et al., 2022) is a novel framework designed to enhance scalability in robot learning using foundation models such as Stable Diffusion (Rombach et al., 2022). CACTI introduces the four stages of data collection, data augmentation, visual representation learning, and imitation policy training. In the data collection stage, limited in-domain expert demonstration data is collected. In the data augmentation stage, CACTI employs visual generative models such as Stable Diffusion (Rombach et al., 2022) to boost visual diversity by augmenting the data with scene and layout variations. In the visual representation learning stage, CACTI leverages

pretrained zero-shot visual representation models trained on out-of-domain data to improve training efficiency. Finally, in the imitation policy training stage, a general multi-task policy is learned using imitation learning on the augmented dataset with compressed visual representations as input. CACTI is trained for multi-task and multi-scene manipulation in kitchen environments, both in simulation and the real world. The use of these techniques enhances the generalization ability of the framework and enables it to learn from a wide range of environments.

As an alternative to costly expert demonstrations for imitation policy training, Ehsani et al. (2024) leverage shortest path plans for navigation and manipulation as expert demonstrations in simulation. The expert, with access to privileged simulator information, navigates using the shortest path planner on a collision mesh and manipulates objects by minimizing the distance between the known pose of the object and the end effector. Expert trajectories are generated for ten different tasks across 200,000 procedurally generated homes. The imitation agent consists of input encoders, a transformer encoder to aggregate the goal and visual knowledge, and a transformer decoder policy network. The resulting agent can generalize across the ten household robotics tasks in simulation and enjoys similar task success rates in real-world experiments.

Beyond language, recent works have investigated other forms of task specification. Notably, MimicPlay (Wang et al., 2023a) presents a hierarchical imitation learning algorithm that learns high-level plans in latent spaces from human play data and low-level motor commands from a small number of teleoperated demonstrations. By harnessing the complementary strengths of these two data sources, this algorithm can significantly reduce the cost of training visuomotor policies for long-horizon manipulation tasks. Once trained, it is capable of performing new tasks based on one human video demonstration at test time. MUTEX (Shah et al., 2023c) further explores learning a unified policy across multimodal task specifications in video, image, text, and audio, showing improved policy performances over single-modality baselines through cross-modal learning.

3.1.2. Language-assisted reinforcement Learning.
Reinforcement learning (RL) is a family of methods that enable a robot to optimize a policy through interaction with its environment by optimizing a reward function. These interactions are usually in a simulation environment, sometimes augmented with data from physical robot hardware for sim-to-real transfer. RL has close ties to optimal control. Unlike imitation learning, RL does not require human demonstrations, and (in theory) has the potential to attain super-human performance. In the RL problem, the expected return of a policy is maximized using the collected roll-outs from interactions with the environment. The feedback received from the environment in the form of a reward signal guides the robot to learn which actions lead to favorable results and which do not. In this section, we discuss works

that have incorporated foundation models (LLMs, VLMs, etc.) into RL problems.

Fast and flexible adaptation is a desired capability of artificial agents and is essential for progress toward general intelligence. In Adaptive Agent (AdA) (Team et al., 2023), the authors present an RL foundation model that is an agent pretrained on diverse tasks and is designed to quickly adapt to open-ended embodied 3D problems by using fast in-context learning from feedback. This work considers navigation, coordination, and division of labor tasks. Given a few episodes within an unseen environment at test time, the agent engages in trial-and-error exploration to refine its policy toward optimal performance. In AdA a transformer architecture is trained using model-based RL[2] (Duan et al., 2016) to train agents with large-scale attention-based memory, which is required for adaptation. Transformer-XL (Dai et al., 2019) with some modification is used to enable long and variable-length context windows to increase the model memory to capture long-term dependencies. The agent collects diverse data in the XLand environment that includes 10^{40} possible tasks (Team et al., 2021), in an automated curriculum. In addition, distillation is used to enable scaling to models with more than 500M parameters.

Palo et al. (2023) propose an approach to enhance reinforcement learning by integrating Large Language Models (LLMs) and Visual-Language Models (VLMs) to create a more unified RL framework. This work considers robot manipulation tasks. Their approach addresses core RL challenges related to exploration, experience reuse and transfer, skills scheduling, and learning from observation. The authors use an LLM to decompose complex tasks into simpler sub-tasks, which are then utilized as inputs for a transformer-based agent to interact with the environment. The agent is trained using a combination of supervised and reinforcement learning, enabling it to predict the optimal sub-task to execute based on the current state of the environment.

3.2. Robot transformers

Foundation models can be used for end-to-end control of robots by providing an integrated framework that combines perception, decision-making, and action generation. Xiao et al. (2022) demonstrate the effectiveness of self-supervised visual pretraining using real-world images for learning motor control tasks directly from pixel inputs. This work is focused on robot manipulation tasks. They show that without any task-specific fine-tuning of the pretrained encoder, the visual representations can be utilized for various motor control tasks. This approach highlights the potential of leveraging self-supervised learning from real-world images to acquire general visual representations that can be applied across different motor control tasks. Similarly, Radosavovic et al. (2023) investigate the use of self-supervised visual pretraining on diverse, in-the-wild videos for real-world robotic tasks. This work considers robot

manipulation tasks. They find that the pretrained representations obtained from such videos are effective in a range of real-world robotic tasks, considering different robotic embodiments. This suggests that the learned visual representations generalize well across various tasks and robot platforms, demonstrating the broad applicability of self-supervised pretraining for real-world robotic applications. Both studies emphasize the advantages of self-supervised visual pretraining, where models are trained on large amounts of unlabeled data to learn useful visual representations. By leveraging real-world images and videos, these approaches enable learning from diverse and unstructured visual data, leading to more robust and transferable representations for motor control tasks in robotic systems.

Another example of a Transformer-based policy model is the work on Robotics Transformer (RT-1) (Brohan et al., 2023a), where the authors demonstrate a model that shows promising scalability properties. To train the model, the authors use a large dataset of over 130k real-world robotic experiences, comprising more than 700 tasks, that was collected over 17 months using a fleet of 13 robots. RT-1 receives images and natural language instructions as inputs and outputs discretized base and arm actions. It can generalize to new tasks, maintain robustness in changing environments, and execute long-horizon instructions. The authors also demonstrate the model's capability to effectively absorb data from diverse domains, including simulations and different robots.

The follow-up work, called Robotic Transformer 2 (RT-2) (Zitkovich et al., 2023), demonstrates a vision-language-action (VLA) model that takes a step further by learning from both web and robotics data. The model effectively utilizes this data to generate generalized actions for robotic control. To do so, the authors use pre-existing vision-language models and directly co-fine-tune them on robot trajectories resulting in a single model that operates as a language model, a vision-language model, and a robot policy. To make co-fine-tuning possible, the actions are represented as simple text strings which are then tokenized using an LLM tokenizer into text tokens. The resulting model, RT-2, enables vision-language models to output low-level closed-loop control. Similarly to RT-1, actions are produced based on robot instructions paired with camera observations and the action space includes 6-DoF positional and rotational displacement of the robot end-effector, gripper extension, and episode termination command. Via extensive experiments, the authors show that utilizing VLMs aids in the enhancement of generalization across visual and semantic concepts and enables the robots to respond to the so-called chain of thought prompting, where the agent performs more complex, multi-stage semantic reasoning. Both RT-1 and RT-2 consider robot manipulation and navigation tasks using a real-world mobile manipulator robot from Everyday Robots. One key limitation of RT-2 and other related works in robotics is the fact that the range of physical skills exhibited by the robot is limited to the

distribution of skills observed within the robot's data. While one way to approach this limitation is to collect more diverse and dexterous robotic data, there might be other intriguing research directions, such as using motion data in human videos, robotic simulations, or other robotic embodiments.

The next work utilizing the Transformer architecture indeed focuses on learning from data that combines multiple robotic embodiments. In RT-X (Padalkar et al., 2023), the authors provide a number of datasets in a standardized data format and models to make it possible to explore the possibility of training large cross-embodied robotic models in the context of robotic manipulation. In particular, they assembled a dataset from 22 different robots collected through a collaboration between 21 institutions, demonstrating 527 skills (160266 tasks). With this unified dataset, RT-X demonstrates that RT-1- and RT-2-based models trained on this multi-embodiment, diverse data exhibit positive transfer across robotic domains and improve the capabilities of multiple robots by leveraging experience from other platforms.

Other works have investigated general pretrained transformers for robot control, trained with self-supervised trajectory data from multiple robots. For example, Perception-Action Causal Transformer (PACT) (Bonatti et al., 2023) is a generative transformer architecture that builds representations from robot data with self-supervision. This work considers robot navigation tasks. PACT pretrains a representation useful for multiple tasks on a given robot. Similar to how large language models learn from extensive text data, PACT is trained on abundant safe state-action data (trajectories) from a robot, learning to predict appropriate safe actions. By predicting states and actions over time in an autoregressive manner, the model implicitly captures dynamics and behaviors specific to a robot. PACT was tested in experiments involving mobile agents: a wheeled robot with a LiDAR sensor (MuSHR) and a simulated agent using first-person RGB images (Habitat). The results show that this robot-specific representation can serve as a starting point for tasks like safe navigation, localization, and mapping. Additionally, the experiments demonstrated that fine-tuning smaller task-specific networks on the pre-trained model leads to significantly better performance compared to training a single model from scratch for all tasks simultaneously, and comparable performance to training a separate large model for each task independently.

Another work in this space is Self-supervised Multi-task pretrAining with contRol Transformer (SMART) (Sun et al., 2023f), which introduces a self-supervised multi-task pertaining to control transformers, providing a pretraining-finetuning approach tailored for sequential decision-making tasks. During the pretraining phase, SMART captures information essential for both short-term and long-term control, facilitating transferability across various tasks. Subsequently, the finetuning process can adapt to a wide variety of tasks spanning diverse domains. Experimentation underscores SMART's ability to enhance learning efficiency across tasks and domains. This work considers cart pole-swing-up, cart pole-balance, hopper-hop, hopper-stand, cheetah-run, walker-stand walker-run, and walker-walk tasks. The approach demonstrates robustness against distribution shifts and proves effective with low-quality pretraining datasets.

Some works have investigated transformer models in conjunction with classical planning and control layers as part of a modular robot control architecture. For example, in Bucker et al. (2023), a multi-modal transformer (LATTE) is presented that allows a user to reshape robot trajectories using language instructions. This work considers both robot manipulation and navigation tasks. LATTE transformer takes as input geometrical features of an initial trajectory guess along with the obstacle map configuration, language instructions from a user, and images of each object in the environment. The model's output is modified for each waypoint in the trajectory so that the final robot motion can adhere to the user's language instructions. The initial trajectory plan can be generated using any geometric planner such as A*, RRT*, or model predictive control. Subsequently, this plan is enriched with the semantic objectives within the model. LATTE leverages pretrained language and visual-language models to harness semantic representations of the world.

One key obstacle to incorporating foundation models into robotics research is the reliance on real-world hardware experiments. These experiments challenge reproducibility because replicating the results may require the exact equipment used in prior studies. Conversely, many recent studies have employed simulators that do not faithfully replicate physical interactions, focusing instead on high-level, long-term tasks and visual modeling. A blend of open hardware configurations and benchmarking in high-fidelity physics simulators can enhance reproducibility and address associated issues.

3.3. Language-image goal-conditioned value learning

In value learning, the aim is to construct a value function that aligns goals in different modalities and preserves temporal coherence due to the recursive nature of the value function. Reusable Representation for Robotic Manipulations (R3M) (Nair et al., 2022b) provides pretrained visual representation for robot manipulation using diverse human video datasets such as Ego4D and can be used as a frozen perception module for policy learning in robot manipulation tasks. R3M's pretrained visual representation is demonstrated on Franka Emika Panda's arm and enables different downstream manipulation tasks. R3M is trained using time-contrastive learning to capture temporal dependencies, video-language alignment to capture semantic features of the scene (such as objects and their relationships), and $L1$ penalty to encourage sparse and compact representation. For a batch of videos, using time-contrastive loss, an encoder is trained to generate a representation wherein the

distance between images that are temporally closer is minimized compared to images that are farther apart in time or from different videos.

Similar to R3M, Value-Implicit Pretraining (VIP) (Ma et al., 2023b) employs time-contrastive learning to capture temporal dependencies in videos, but it does not require video-language alignment. VIP is also focused on robot manipulation tasks. VIP is a self-supervised approach for learning visual goal-conditioned value functions and representations from videos. VIP learns visual goal-based rewards for downstream tasks and can be used for zero-shot reward specification. The reward model is derived from pretrained visual representations. Pretraining involves using unlabeled human videos. Human videos do not contain any action information to be used for robot policy learning; therefore, the learned value function does not explicitly depend on actions. VIP introduces a novel time contrastive objective that generates a temporally smooth embedding. The value function is implicitly defined via distance embedding. The proposed implicit time contrastive learning attracts the representation of the initial and goal frames in the same trajectory and repels the representation of intermediate frames by recursive one-step temporal difference minimization. This representation captures long-term temporal dependencies across task frames and local temporal smoothness among adjacent frames.

Language-Image Value Learning (LIV) (Ma et al., 2023a) is a control-centric vision-language representation. LIV generalizes the prior work VIP by learning multi-modal vision-language value functions and representations using language-aligned videos. For tasks specified as language goals or image goals, a multi-model representation is trained that encodes a universal value function. LIV is also focused on robot manipulation tasks. LIV is a pretrained control-centric vision-language representation based on large human video datasets such as EPIC-KITCHENS (Damen et al., 2018). The representations are kept frozen during policy learning. A simple MLP is used on top of pretrained representations for the policy network. Policy learning is decoupled from language-visual representation pretraining. The LIV model is pretrained on arbitrary video activity datasets with text annotation, and the model can be fine-tuned on small datasets of in-domain robot data to ground language in a context-specific way. LIV uses a generalization of the mutual information-based image-text contrastive representation learning objective as used in CLIP, so LIV can be considered as a combination of CLIP and VIP. Both VIP and LIV learn a self-supervised goal-conditioned value-function objective using contrastive learning. The LIV extends the VIP framework to multi-modal goal specifications. LOREL (Nair et al., 2022a) learns a language-conditioned reward from offline data and uses it during model predictive control to complete language-specified tasks.

Value functions can be used to help ground semantic information obtained from an LLM to the physical environment in which a robot is operating. By leveraging value functions, a robot can associate the information processed by the LLM with specific locations and objects in its surroundings. In SayCan (Brohan et al., 2023b), researchers investigate the integration of large language models with the physical world through learning. They use the language model to provide task-grounding (Say), enabling the determination of useful sub-goals based on high-level instructions, and a learned affordance function to achieve world-grounding (Can), enabling the identification of feasible actions to execute the plan. Inner Monologue (Huang et al., 2022b) studies the role of grounded environment feedback provided to the LLM, thus closing the loop with the environment. The feedback is used for robot planning with large language models by leveraging a collection of perception models (e.g., scene descriptors and success detectors) in tandem with pretrained language-conditioned robot skills. Feedback includes task-specific feedback, such as success detection, and scene-specific feedback (either "passive" or "active"). In both SayCan and Inner Monologue, robot manipulation and navigation tasks are considered using a real-world mobile manipulator robot from Everyday Robots. Text2Motion (Lin et al., 2023c) is a language-based planning framework for long-horizon robot manipulation. Similar to SayCan and Inner Monologue, Text2Motion computes a score (S_{LMM}) associated with each skill at each time step. The task planning problem is to find a sequence of skills by maximizing the likelihood of a skill sequence given a language instruction and the initial state. In Text2Motion, the authors propose to verify that the generated long-horizon plans are symbolically correct and geometrically feasible. Hence, a geometric feasibility score (S_{geo}) is defined as the probability that all the skills in the sequence achieve rewards. To compute the overall score, the LLM score is multiplied by the geometric feasibility score ($S_{\text{Skill}} = S_{\text{LMM}} \cdot S_{\text{geo}}$).

VoxPoser (Huang et al., 2023c) builds 3D value maps to ground affordances and constraints into the perceptual space. VoxPser considers robot manipulation tasks. Given the RGB-D observation of the environment and language instruction, VoxPoser utilizes large language models to generate code, which interacts with vision-language models to extract a sequence of 3D affordance maps and constraint maps. These maps are composed together to create 3D value maps. The value maps are then utilized as objective functions to guide motion planners to synthesize trajectories for everyday manipulation tasks without requiring any prior training or instruction.

In Mahmoudieh et al. (2022), reward shaping using CLIP is presented. This work considers robot manipulation tasks. The proposed model utilizes CLIP to ground objects in a scene described by the goal text paired with spatial relationship rules to shape the reward by using raw pixels as input. They use developments in building large-scale visuo-lingual models like CLIP to devise a framework that generates the task reward signal from just the goal text description and raw pixel observations. This signal is then used to learn the task policy.

In Mees et al. (2023), Hierarchical Universal Language Conditioned Policies 2.0 (HULC++) is presented. This work considers robot manipulation tasks. A self-supervised visuo-lingual affordance model is used to learn general-purposed language-conditioned robot skills from unstructured offline data in the real world. This method requires annotating as little as 1% of the total data with language. The visuo-lingual affordance model has an encoder-decoder architecture with two decoder heads. Both heads share the same encoder and are conditioned on the input language instruction. One head predicts a distribution over the image, in which each pixel likelihood is an afforded point. The other head predicts a Gaussian distribution from which the corresponding predicted depth is sampled. Given visual observations and language instructions as input, the affordance model outputs a pixel-wise heat map that represents affordance regions and the corresponding depth map.

3.4. Robot task planning using large language models

Large language models (LLMs) can be used to provide high-level task planning for performing complex long-horizon robot tasks.

3.4.1. Language instructions for task specification.
As discussed above, SayCan (Brohan et al., 2023b) uses an LLM for high-level task planning in language, though with a learned value function to ground these instructions in the environment.

Temporal logic is useful for imposing temporal specifications in robotic systems. In Chen et al. (2023e), translation from natural language (NL) to temporal logic (TL) is proposed. A dataset with 28k NL-TL pairs is created and the T5 (Raffel et al., 2020) model is finetuned using the dataset. LLMs are often used to plan task sub-goals. This work considers robot navigation tasks. In Chen et al. (2023d), instead of direct task planning, a few-shot translation from a natural language task description to an intermediary task representation is performed. This representation is used by a Task and Motion Planning (TAMP) algorithm to jointly optimize task and motion plans. Autoregressive re-prompting is used to correct synthetic and semantic errors. This work also considers robot navigation tasks.

3.4.2. Code generation using language models for task planning.
Classical task planning requires extensive domain knowledge and the search space is large (Huang et al., 2021; Sun et al., 2021a). LLMs can be used to generate sequences of tasks required to achieve a high-level task. In ProgPrompt (Singh et al., 2023), the authors introduce a prompting method that uses LLMs to generate sequences of actions directly with no additional domain knowledge. The prompt to the LLM includes specifications of the available actions, objects in the environment, and example programs that can be executed. VirtualHome (Puig et al., 2018) is used as a simulator for demonstration.

Code-as-Policies (Liang et al., 2023) explores the use of code-writing LLMs to generate robot policy code based on natural language commands. This work considers robot manipulation and navigation tasks using a real-world mobile manipulator robot from Everyday Robots. The study demonstrates that LLMs can be repurposed to write policy code by expressing functions or feedback loops that process perception outputs and invoke control primitive APIs. To achieve this, the authors utilize few-shot prompting, where example language commands formatted as comments are provided along with the corresponding policy code. Without any additional training on this data, they enable the models to autonomously compose API calls and generate new policy code when given new commands. The approach leverages classic logic structures and references third-party libraries like NumPy and Shapely to perform arithmetic operations. By chaining these structures and using contextual information (behavioral commonsense), the LLMs can generate robot policies that exhibit spatial-geometric reasoning, generalize to new instructions, and provide precise values (e.g., velocities) for ambiguous descriptions such as "faster." The concept of "code as policies" formalizes the generation of robot policies using language model-generated programs (LMPs). These policies can represent reactive policies like impedance controllers, as well as waypoint-based policies such as vision-based pick and place or trajectory-based control. The effectiveness of this approach is demonstrated on multiple real robot platforms. A crucial aspect of this approach is the hierarchical code generation process, which involves recursively defining undefined functions. This enables the LLMs to generate more complex code structures to fulfill the desired policy requirements.

In Vemprala et al. (2023), the authors provide design principles for using ChatGPT in robotics and demonstrate how LLMs can help robotic capabilities rapidly generalize to different form factors. This work considers robot manipulation and aerial navigation tasks. First, a high-level robot function library that maps to multiple atomic tasks executable by the robot is defined. Then, a prompt is crafted that includes these functions, and the required constraints along the task description. ChatGPT then provides executable code specific to the given robot configuration and task. The generated code can then be evaluated by a user and appropriate feedback with modified prompts to LLMs further help refine and generate programs that are safe and deployable on the physical robot. The study demonstrated that such a methodology can be applied to multiple form factors both in simulation and in the real world.

3.5. In-context learning (ICL) for decision-making

In-context learning (ICL) (Dong et al., 2022) operates without the need for parameter optimization, relying instead on a set of examples included in the prompt (the concept of prompting). This learning approach is intimately linked

with prompt engineering and finds extensive use in natural language processing. The method of Chain-of-Thought (Wei et al., 2022) is a prominent technique within in-context learning. It involves executing a sequence of intermediate steps to arrive at the final solution for complex, multi-step problems. This technique allows models to produce step-by-step explanations that parallel human cognitive processes. However, despite its numerous benefits, ICL also faces certain challenges, including issues related to ambiguity and interpretation, domain-specific knowledge, transparency, and explainability. In-context learning has had a significant impact on the field of LLMs in a broad sense, and many robotics works have used it to apply LLMs to specific domains. Investigating this, Mirchandani et al. (2023) illustrate that Large Language Models (LLMs) possess remarkable pattern recognition abilities. They reveal that, through in-context learning, LLMs can effectively handle general patterns that extend beyond standard language-based prompts. This capability allows for the application of LLMs in scenarios such as offline trajectory optimization and online, in-context reinforcement learning. Additionally, Jia and the team in their work on Chain-of-Thought Predictive Control (Jia et al., 2023) suggest a method to identify specific brief sequences within demonstrations, termed as 'chain-of-thought'. They focus on understanding and representing the hierarchical structure of these sequences, highlighting the achievement of subgoals within tasks. This work considers robot policy learning from demonstrations for contact-rich object manipulation tasks.

3.6. Open-vocabulary robot navigation and manipulation

3.6.1. Open-vocabulary navigation. Open-vocabulary navigation addresses the challenge of navigating through unseen environments. The open-vocabulary capability signifies that the robot possesses the capacity to comprehend and respond to language cues, instructions, or semantic information, without being restricted to a predefined dataset. In this section, we explore papers that examine the integration of LLMs, VLMs, or a combination of both in a plug-and-play manner for robot navigation tasks. Additionally, we discuss papers that take a different approach by constructing foundation models explicitly tailored for robot navigation tasks.

In VLN-BERT (Majumdar et al., 2020), the authors present a visual-linguistic transformer-based model that leverages multi-modal visual and language representations for visual navigation using web data. The model is designed to score the compatibility between an instruction, such as "…stop at the brown sofa," and a sequence of panoramic RGB images captured by the agent. Similarly, LM-Nav (Shah et al., 2023a) considers visual navigation tasks. LM-Nav is a system that utilizes pretrained models of images and language to provide a textual interface to visual navigation. LM-Nav demonstrates visual navigation in a real-world outdoor environment from natural language instructions. LM-Nav utilizes an LLM (GPT-3, Brown et al., 2020), a VLM (CLIP, Radford et al., 2021a), and a VNM (Visual Navigation Model). First, LM-Nav constructs a topological graph of the environment via the VNM estimating the distance between images. The LLM is then used to translate the natural instructions to sequences of intermediate language landmarks. The VLM is used to ground the visual observations in landmark descriptions via a joint probability distribution over landmarks and images. Using the VLM's probability distribution, the LLM instructions, and the VNM's graph connectivity, the optimal path is planned using the search algorithm. Then the plan is executed by the goal-conditioned policy of VNM.

While LM-Nav makes use of LLMs and VLMs as plug-and-play for visual navigation tasks, the authors of ViNT (Shah et al., 2023b) propose to build a foundation model for visual navigation tasks. ViNT is an image goal-conditioned navigation policy trained on diverse training data and can control different robots in zero-shot. It can be fine-tuned to be adapted for different robotic platforms and various downstream tasks. ViNT is trained on various navigation datasets from different robotic platforms. It is trained with goal-reaching objectives and utilizes a Transformer-based architecture to learn navigational affordances. ViNT encodes visual observations and visual goals using an EfficientNet CNN and predicts temporal distance and normalized actions in an embodiment-agnostic manner. Additionally, ViNT can be augmented with diffusion-based sub-goal proposals to help explore environments not encountered during training. An image-to-image diffusion generates sub-goal images, which the ViNT then navigates toward while building a topological map in the background.

Another work that considers zero-shot navigation tasks is Audio Visual Language Maps (AVLMaps) (Huang et al., 2023a). AVLMaps presents a 3D spatial map representation for cross-modal information from audio, visual, and language cues. AVLMaps receives multi-modal prompts and performs zero-shot navigation tasks in the real world. The inputs are depth and RGB images, camera pose, and audio. Visual features are encoded using pretrained foundation models. Visual localization features like NetVLAD (Arandjelovic et al., 2016) and SuperPoint (DeTone et al., 2018), visual-language features like LSeg (Li et al., 2022a), and audio-language features like AudioCLIP (Guzhov et al., 2022) are computed and predictions from different modalities are combined into 3D heatmaps. The pixel-wise joint probability of the heatmap is computed and used for planning. Additionally, navigation policies are generated as executable codes with the help of GPT-3. Finally, 3D heatmaps are predicted indicating the location of multi-modal concepts such as objects, sounds, and images.

Many roboticists may wonder about the comparative strengths of classical modular robot navigation systems versus end-to-end learned systems. Semantic navigation (Gervet et al., 2023) seeks to address this question by presenting an empirical analysis of semantic visual

navigation methods. The study compares representative approaches from classical, modular, and end-to-end learning paradigms across six different homes, without any prior knowledge, maps, or instrumentation. The findings of the study reveal that modular learning methods perform well in real-world scenarios. In contrast, the end-to-end learning approaches face challenges due to a significant domain gap between simulated and real-world images. This domain gap hinders the effectiveness of end-to-end learning methods in real-world navigation tasks. For practitioners, the study emphasizes that modular learning is a reliable approach to object navigation. The modularity and abstraction in policy design enable successful transfer from simulation to reality, making modular learning an effective choice for practical implementations. For researchers, the study also highlights two critical issues that limit the reliability of current simulators as evaluation benchmarks. Firstly, there exists a substantial Sim-to-Real gap in images, which hampers the transferability of learned policies from simulation to the real world. Secondly, there is a disconnect between simulation and real-world error modes, which further complicates the evaluation process.

Another line of work in open-vocabulary navigation is object navigation tasks. In this task, the robot must be able to find the object described by humans and navigate towards the object. The navigation task is decomposed into exploration when the language target is not detected, and exploitation when the target is detected and the robot navigates toward the target. As the robot moves in the environment, it creates a top-down map using RGB-D observations and poses estimates. In Gadre et al. (2023), the authors introduce a zero-shot object navigation setting that uses an open-vocabulary classifier such as CLIP (Radford et al., 2021a) to compute the cosine similarity between an image and a user-specified description.

Common datasets and benchmarks for these types of problems are Matterport3D (Anderson et al., 2018; Chang et al., 2017), Gibson (Li et al., 2021) and Habitat (Szot et al., 2021). L3MVN (Bangguo et al., 2023) enhances visual target navigation by constructing an environment map and selecting long-term goals using the inference capabilities of large language models. The system can determine appropriate long-term goals for navigation by leveraging pretrained language models such as RoBERTa-large (Liu et al., 2019), enabling efficient exploration and searching. Chen et al. (Chen et al., 2023a) presents a training-free and modular system for object goal navigation, which constructs a structured scene representation through active exploration. The system utilizes semantic information in the scene graphs to deduce the location of the target object and integrates semantics with the geometric frontiers to enable the agent to navigate effectively to the most promising areas for object search while avoiding detours in unfamiliar environments. HomeRobot (Yenamandra et al., 2023) introduces a benchmark for the Open-Vocabulary Mobile Manipulation (OVMM) task. OVMM task is the problem of finding an object in any unseen environment, navigating towards the

object, picking it up, and navigating towards a goal location to place the object. HomeRobot provides a benchmark in simulation and the real world for OVMM tasks.

3.6.2. Open-vocabulary manipulation. Open-vocabulary manipulation refers to the problem of manipulating any object in a previously unseen environment. VisuoMotor Attention Agent (VIMA) (Jiang et al., 2023) learns robot manipulation from multi-modal prompts. VIMA is a transformer-based agent that predicts motor commands conditioned on a task prompt and a history of interactions. It introduces a new form of task specifications that combines textual and visual tokens. Multi-modal prompting converts different robot manipulation tasks, such as visual goal-reaching, learning from visual demonstrations, and novel concept grounding into one sequence modeling problem. It offers the training of a unified policy across diverse tasks, potentially allowing for zero-shot generalization to previously unseen ones. VIMA-BENCH is introduced as a benchmark for multi-modal robot learning. The VIMA-BENCH simulator supports collections of objects and textures that can be utilized in multi-modal prompting. RoboCat (Bousmalis et al., 2023) is a self-improving AI agent. It uses a 1.18B-parameter decoder-only transformer. It learns to operate different robotic arms, solves tasks from as few as 100 demonstrations, and improves from self-generated data. RoboCat is based on Gato (Reed et al., 2022) architecture and is trained with a self-improvement cycle.

For robots to operate effectively in the real world, they must be able to manipulate previously unseen objects. Liu et al. present StructDiffusion (Liu et al., 2023d), which seeks to enable robots to use partial viewpoint clouds and natural language instructions to construct a goal configuration for objects that were previously seen or unseen. They accomplish this by first using segmentation to break up the scene into objects. Then they use a multi-model transformer to combine word and point cloud embeddings and output a 6-DoF goal pose prediction. The predictions are iteratively refined via diffusion and a discriminator that is trained to determine if a sampled configuration is feasible. Manipulation of Open-World Objects (MOO) (Stone et al., 2023) leverages a pretrained vision-language model to extract object-centric information from the language command and the image and conditions the robot policy on the current image, the instructions, and the extracted object information in a form of a single-pixel overlaid onto the image. MOO uses Owl-ViT for object detection and RT-1 for language-conditioned policy learning.

Another task in robot manipulation involves autonomous scene rearrangement and in-painting. DALL-E-Bot (Kapelyukh et al., 2023) performs zero-shot autonomous rearrangement in the scene in a human-like way using pretrained image diffusion model DALL-E2 (Ramesh et al., 2022). DALL-E-Bot autonomous object rearrangement does not require any further data collection or training. First, the initial observation image (of the disorganized scene) is converted into a per-object representation, including a segmentation mask using Mask R-CNN (He et al., 2017), an

object caption, and a CLIP visual feature vector. Then a text prompt is generated by describing the object in the scene and is given to DALL-E to create a goal image for the rearrangement task (the objects should be rearranged in a human-like way). Next, the objects in the initial and generated images are matched using their CLIP visual features. Poses are estimated by aligning their segmentation masks. The robot rearranges the scene based on the estimated poses to create the generated arrangement.

3.6.3. Open-vocabulary grasping. Open-vocabulary robot grasping refers to a robot's capability to execute grasping tasks for objects described in natural language even if the objects or descriptions were not encountered during training. Roboticists have started utilizing these language-embedded 3D fields to facilitate robot grasping and manipulation of 3D objects and interaction with 3D scenes. Semantic features encoded by pretrained foundation models such as CLIP or DINO are used to be embedded into 3D field representations such as NeRFs or Gaussian Splatting fields. For example, LERF (Kerr et al., 2023) is a language-embedded NeRF scene representation, and LangSplat (Qin et al., 2023) is a language-embedded 3D Gaussian Splatting environment representation. These language-embedded fields enable open-vocabulary grasping and manipulation. In LERF-TOGO (Rashid et al., 2023), the authors present a zero-shot open-vocabulary grasping framework that generates grasp proposals over an object given a natural language instruction. First, a 3D object mask is created using DINO features, and then LERF is conditionally queried on this mask to obtain a 3D semantic relevancy map. This relevancy map extracts parts of the object suitable for grasping (affordance) based on the given text. The relevancy map is input to an off-the-shelf grasp planner (Fang et al., 2020) to rank grasp proposals. Similarly, F3RM presents few-shot language-guided robot manipulation (Shen et al., 2023b) that leverages a NeRF-based distilled feature field (Zhou et al., 2022) to embed task-specific features into the 3D scene field. These features are learned and extracted from human demonstrations, embedded into the field, and then utilized to generate, rank, and refine grasps. Both LERF-TOGO and F3RM rely on NeRF scene representation. However, NeRF training and rendering are computationally intensive and can be infeasible for real-time robotic applications. Moreover, editing NeRFs is both challenging and resource-demanding. To address these limitations, the authors of Splat-MOVER (Shorinwa et al., 2024) propose using a language-embedded Gaussian Splatting 3D field. Gaussian Splatting (Kerbl et al., 2023) is a recently developed technique that excels in providing real-time performance for rendering and editing. Splat-MOVER capitalizes on the flexibility of Gaussian Splatting to dynamically modify the scene as the robot operates and interacts with its environment. Compared to LERF-TOGO and F3RM, Splat-MOVER supports multi-stage open-vocabulary manipulation, a feature enabled by the real-time scene editability of Gaussian Splatting.

In Table 2, some robotic-specific foundation models are reported along with information about their size and architecture, pretrained task, inference time, and hardware setup.

3.7. Challenges and perspectives

Empowering robots with foundation model capabilities brings significant versatility and adaptability across a range of downstream tasks. These models enable robots to generalize across platforms, environments, tasks, and open-vocabulary settings. Moreover, they introduce emergent capabilities that allow robots to devise novel solutions beyond what is explicitly present in their training data. Although there have been exciting advancements in creating generalizable solutions in robotics with the help of foundation models, several challenges continue to exist. One of the main advantages of foundation models is their ability to generalize across various tasks. However, there are inherent limitations. For instance, a robot transformer trained specifically for manipulators may not be effective when applied to drones or humanoid robots. This limitation is not adequately addressed in the existing literature. While current research highlights impressive cross-embodiment and generalization capabilities, these achievements are often demonstrated within the context of single-arm tabletop manipulation in controlled and predictable environments. In contrast, real-world applications, such as open-world navigation, dealing with dynamic obstacles, and evolving tasks, require more sophisticated adaptability, including the ability to retrain online based on new experiences. In robot policy learning using imitation learning, key challenges are (i) obtaining a sufficient number of demonstrations along with conditioning labels necessary for policy training, and (ii) addressing distribution shifts during closed-loop policy operation, which can cause the robot to enter poorly represented areas of the state space in the training data, thereby degrading its performance. Robot transformers have been recently developed and can provide unprecedented generalization capabilities. However, their reliance on real-world hardware limits their reproducibility. To enhance benchmarking and ensure reproducibility, it is essential to develop open hardware platforms and high-fidelity, photorealistic, physics-based simulators. In robot planning, incorporating multimodal information, including audio cues, has the potential to enhance planning performance. However, there are currently few works that utilize audio for prediction and planning purposes. While current robotic-specific foundation models primarily focus on single-arm pick-and-place tasks and mobile robot navigation, there is a need for more research in areas such as precise grasping and dexterity (that capture a wide range of actions like tossing, pushing, and chopping), flying robots, and collaborative multi-robot systems. According to Table 2, the substantial hardware requirements of some models, as well as their high inference time, may hinder their feasibility for training and deployment in real-time robotics applications. Thus more research should be conducted to improve these aspects.

Table 2. Pretrained Models for Robotics.

Paper	Backbone	Size (Parameters)	Pretrained Task	Inference Speed	Hardware[a]
RoboCat (Bousmalis et al., 2023)	Decoder-only transformer	1.18B	manipulation	10-20 Hz	
Gato (Reed et al., 2022)	Decoder-only transformer	1.2B	Generalist agent	20 Hz	4 days on 16x16 TPU v3 slice
PaLM-E-562B (Driess et al., 2023)	Decoder-only transformer	562B	1 Hz for Language subgoals + 5 Hz low-level control policies	5-6 Hz	Runs on multi-TPU cloud service
ViNT (Shah et al., 2023b)	EfficientNet+decoder transformer	31M	Visual navigation	4 Hz	Variety of GPU configurations is used including 2× 4090, 3× Titan Xp, 4× P100, 8× 1080Ti, 8× V100, and 8× A100
VPT (Baker et al., 2022)	A temporal convolution layer, a ResNet 62 image processing stack, and residual unmasked attention layers,	0.5 B	Embodied agent in Minecraft	20 Hz	9 days on 720 V100 GPUs
RT-1 (Brohan et al., 2023a)	Conditioned EfficientNet + TokenLearner + decoder-only transformer	35M	Real-world robotics tasks	3 Hz	
RT-2 (Zitkovich et al., 2023)	PaLI-X	55B	Real-world robotics tasks	1-3 Hz	Runs on multi-TPU cloud service
RT-2-X (Padalkar et al., 2023)	ViT and Language model UL2 (Yi et al., 2023)	55B	Real-world robotics tasks	1-3 Hz	Runs on multi-TPU cloud service
LIV (Ma et al., 2023a)	CLIP		reward learning	15 Hz	8 NVIDIA V100 GPUs
SMART (Sun et al., 2023f)	Decoder-only transformer	11M	Bidirectional dynamics prediction and masked hindsight control	1 Hz	8 Nvidia V100 GPUs
COMPASS (Ma et al., 2022)	3D-Resnet encoder	20M	Contrastive loss	30 Hz	8 Nvidia V100 GPUs
PACT (Bonatti et al., 2023)	Decoder-only transformer	12M	Forward dynamics and next action prediction	10 Hz (edge)/ 50 Hz	Nvidia Xavier NX (edge)/8 Nvidia V100 GPUs

[a]Empty fields in the table denote no data is reported.

4. Perception

Robots interacting with their surrounding environments receive raw sensory information in different modalities such as images, video, audio, and language. This high-dimensional data is crucial for robots to understand, reason, and interact in their environments. Foundation models, including those that have been developed in the vision and NLP domains, are promising tools for converting these high-dimensional inputs into abstract, structured representations that can be more easily interpreted and manipulated. Particularly, multi-modal foundation models enable robots to integrate different sensory inputs into a unified representation encompassing semantic, spatial, temporal, and affordance information. These multi-modal models reflect cross-modal interactions, often by aligning elements across modalities to ensure coherence and correspondence. For example, text and image data are aligned for image captioning tasks. This section will explore a range of tasks related to robot perception that are improved through aligning modalities using foundation models, with a focus on vision and language. There is an extensive body of literature studying multi-modality in the machine learning community, and an interested reader is referred to the survey paper (Baltrusaitis et al., 2019) that presents a taxonomy of multi-modal learning. We focus on applications of multi-modal models to robotics.

4.1. Open-vocabulary object detection and 3D classification

In this section, we discuss zero-shot open-vocabulary object detection and classification.

4.1.1. Object detection. Zero-shot object detection allows robots to identify and locate objects they have never encountered previously. Grounded Language-Image Pre-

training (GLIP) (Li et al., 2022d) integrates object detection and grounding by redefining object detection as phrase grounding. This reformulation enables the learning of a visual representation that is both language-aware and semantically rich at the object level. In this framework, the input to the detection model comprises not only an image but also a text prompt that describes all the potential categories for the detection task. To train GLIP, a dataset of 27 million grounding instances was compiled, consisting of 3 million human-annotated pairs and 24 million image-text pairs obtained by web crawling. The results of the study demonstrate the remarkable zero-shot and few-shot transferability of GLIP to a wide range of object-level recognition tasks. Recently, PartSLIP (Liu et al., 2023a) demonstrated that GLIP can be used for low-shot part segmentation on 3D objects. PartSLIP renders a 3D point cloud of an object from multiple views and combines 2D bounding boxes in these views to detect object parts. To deal with noisy 2D bounding boxes from different views, PartSLIP runs a voting and grouping method on super points from 3D, assigns multi-view 2D labels to super points, and finally groups super points to obtain a precise part segmentation. To enable few-shot learning of 3D part segmentation, prompt tuning, and multi-view feature aggregation are proposed to improve performance.

OWL-ViT (Minderer et al., 2022) is an open-vocabulary object detector. OWL-ViT uses a vision transformer architecture with contrastive image-text pre-training and detection end-to-end fine-tuning. Unlike GLIP, which frames detection as a phrase grounding problem with a single text query and limits the number of possible object categories, OWL-ViT can handle multiple text-based or image-driven queries. OWL-ViT has been applied to robot learning for example in VoxPoser (Huang et al., 2023c) as the open-vocabulary object detector to find "entities of interest" (e.g., vase or drawer handles) and ultimately define value maps for optimizing manipulation trajectories.

Grounding DINO (Liu et al., 2023c) combines DINO (Caron et al., 2021) with grounded pre-training, extending the closed-set DINO model to open-set detection by fusing vision and language. Grounding DINO outperforms GLIP in open-set object detection. This superior performance is mainly due to the transformer architecture of Grounding DINO, which facilitates multi-modal feature fusion at multiple stages.

4.1.2. 3D classification.
Zero-shot 3D classifiers can enable robots to classify objects in their environments without explicit training data. Foundation models are strong candidates for performing 3D classification. PointCLIP (Zhang et al., 2022a) transfers CLIP's pre-trained knowledge of 2D images to 3D point cloud understanding by aligning point clouds with text. The authors propose to project each point onto a series of pre-defined image planes to generate depth maps. Then, the CLIP visual encoder is used to encode multi-view features of the point cloud and predict labels in natural language for each view. The final prediction for the point cloud is computed via weighted aggregation of the predictions for each view. PointBERT (Yu et al., 2022) uses a transformer-based architecture to extract features from point clouds, generalizing the concept of BERT into 3D point clouds.

Unlike PointCLIP, which converts the task of matching point clouds and text to image-text alignment, ULIP (Xue et al., 2022, 2023) is a Unified representation of Language, Images, and Point clouds for 3D understanding. It achieves this by pre-training with object triplets (image, text, point cloud). The model is trained using a small number of automatically synthesized triplets from ShapeNet (Chang et al., 2015), which is a large-scale 3D model repository. ULIP uses CLIP as the vision-language model. During pretraining, the CLIP model is kept frozen and a 3D encoder is trained by aligning the 3D features of an object with its associated textual and visual features from CLIP using contrastive learning. The pretraining process allows ULIP to learn a joint embedding space where the three modalities are aligned. One of the major advantages of ULIP is that it can substantially improve the recognition ability of 3D backbone models. This is because the pretraining process allows ULIP to learn more robust and discriminative features for each modality, which can then be used to improve the performance of 3D models. Another advantage of ULIP is that it is agnostic to the 3D model architecture, and thus can be easily integrated into the pretraining process of existing 3D pipelines. ULIP adopts masked language modeling from BERT to 3D by tokenizing 3D patches, randomly masking out 3D tokens, and predicting them back during pretraining. ULIP (Xue et al., 2022, 2023) has shown that the performance of recognition capability of models such as PointBERT can be improved by using a unified multimodal representation of ULIP.

4.2. Open-vocabulary semantic segmentation

Semantic segmentation classifies each pixel in an image into semantic classes. This provides fine-grained information about object boundaries and locations within an image and enables embodied agents to understand and interact with the environment at a more granular level. Several works explore how foundation models such as CLIP can enhance the generalizability and flexibility of semantic segmentation tasks.

LSeg is a language-driven semantic segmentation model (Li et al., 2022a) that associates semantically similar labels to similar regions in an embedding space. LSeg uses a text encoder based on the CLIP architecture to compute text embeddings and an image encoder with the underlying architecture of Dense Prediction Transformer (DPT) (Ranftl et al., 2021). Similar to CLIP, LSeg creates a joint embedding space using text and image embeddings. LSeg freezes the text encoder at training time and trains the image encoder to maximize the correlation between the text embedding and the image pixel embedding of the ground-truth pixel class. It allows users to arbitrarily shrink, expand, or

rearrange the label set (with unseen categories) for any image at test time.

Segment Anything Model (SAM) (Kirillov et al., 2023) introduces a framework for promptable segmentation consisting of the task definition for promptable segmentation, a segmentation foundation model (the Segment Anything Model, or SAM), and a data engine. SAM adapts a pretrained Vision Transformer from Masked Auto-Encoder (MAE) (He et al., 2022) as an image encoder while using a text encoder from CLIP (Radford et al., 2021b) for sparse prompts (points, boxes, and text) and a separate dense prompt encoder for masks. In contrast to other foundation models that are trained in an unsupervised manner on web-scale data, SAM is trained using supervised learning with data engines that help scale the number of available annotations. Along with the model, the authors released the Segment Anything 1 Billion (SA-1B) dataset. It consists of 11M images and 1.1 B segmentation masks. In the work, the authors conducted experiments on five zero-shot transfer tasks, including point-valid mask evaluation, edge detection, object proposal, instance segmentation, and text-to-mask. The system's composable design, facilitated by prompt engineering techniques, enables a broader range of applications compared to systems trained specifically for fixed task sets. However, one limitation of this work that is particularly relevant to robotic applications is that SAM cannot run in real time.

FastSAM (Zhao et al., 2023) and MobileSAM (Zhang et al., 2023a) achieve comparable performance to SAM at faster inference speeds. The Track Anything Model (TAM) (Yang et al., 2023a) combines SAM and XMem (Cheng and Schwing, 2022), an advanced video object segmentation (VOS) model, to achieve interactive video object tracking and segmentation. Anything-3D (Shen et al., 2023a) employs a collection of visual-language models and SAMs to elevate objects into the realm of 3D. It uses BLIP (Li et al., 2022c) to generate textual descriptions while using SAM to extract objects of interest from visual input. Then, Anything-3D lifts the extracted objects into a Neural Radiance Field (NeRF) (Mildenhall et al., 2021) representation using a text-to-image diffusion model, enabling their integration into 3D scenes.

Amidst these remarkable advancements, achieving fine-grained detection with real-time performance still remains challenging. For example, LSeg (Li et al., 2022a) reports failure cases related to misclassification, when the test time input labels do not include the true label for the pixel, and the model thus assigns the highest probability to the closest label. Another failure case occurs when multiple labels can be correct for a particular pixel, and the model must classify it as just one of the categories. For example "window" and "house" may both be defined as labels, but during inference, a pixel representing a "window" may be labeled instead as "house." SAM also does not provide precise segmentation for fine structures and often fails to produce crisp boundaries. All models that use SAM as a sub-component may encounter similar limitations. In the future, fine-grained

semantic segmentation models that can assign multiple labels to a pixel when there are multiple correct descriptions should be considered. Additionally, developing models that can run in real time will be critical for robotics applications.

4.3. Open-vocabulary 3D scene and object representations

Scene representations allow robots to understand their surroundings, facilitate spatial reasoning, and provide contextual awareness. Language-driven scene representations align textual descriptions with visual scenes, enabling robots to associate words with objects, locations, and relationships. In this section, we study recent works that use foundation models to enhance scene representations.

4.3.1. Language grounding in 3D scene. Language grounding refers to combining geometric and semantic representations of an environment. One type of representation that can provide an agent with a strong geometric prior is an implicit representation. One example of an implicit representation is a Neural Radiance Field (NeRF, Mildenhall et al., 2021; Sun et al., 2023d,c). NeRF creates high-quality 3D reconstructions of scenes and objects from a set of 2D images captured from different viewpoints (without the need for explicit depth information). The NeRF neural network takes camera poses as input and predicts the 3D geometry of the scene as well as color and intensity. Most NeRF-based models memorize the light field in a single environment and are not pre-trained on a large data set, hence they are not foundation models. However, foundation models such as CLIP can be combined with NeRFs to extract semantic information from an agent's environment.

Kerr et al. (2023) propose language-embedded radiance fields (LERFs) that ground CLIP embeddings into a dense multi-scale 3D field. This results in a 3D representation of the environment that can be queried to produce semantic relevancy maps. The LERF model takes 3D position (x, y, z), viewing direction (ϕ, θ), and a scaling factor as input and outputs an RGB value, density (σ), as well as DINO (Caron et al., 2021) and CLIP features. The LERF is optimized in two stages: initially, a multi-scale feature pyramid of CLIP embeddings over training views is computed; then, the pyramid is interpolated using the image scale and pixel location to obtain the CLIP embedding; and finally, the CLIP embeddings are supervised through cosine similarity, and the RGB and density are supervised using the standard mean squared-error.

Models such as LERF inherit the shortcomings of CLIP and NeRF. For example, CLIP exhibits difficulty in capturing spatial relationships between objects. In addition, language queries from CLIP can highlight a significant issue similar to the bag-of-words model, which struggles to distinguish terms with opposite sentiments. Also, NeRF relies on known camera poses associated with pre-captured multi-view images.

In CLIP-Fields (Shafiullah et al., 2023), an implicit scene representation $g(x, y, z) : \mathcal{R}^3 \rightarrow \mathcal{R}^d$ is trained by decoding a d-dimensional latent vector to different modality-specific outputs. The model distills information from pretrained image models by back-projecting the pixel labels to 3D space and training the output heads to predict semantic labels from an open-vocab object detector called Detic, the CLIP visual representation, and one-hot instance labels using a contrastive loss. The scene representation can then be used as a spatial database for segmentation, instance identification, semantic search over space, and 3D view localization from images.

Another related work is VLMaps (Huang et al., 2023b), which projects pixel embeddings from LSeg to grid cells in a top-down grid map. This method does not require training and instead directly backprojects pixel embeddings to grid cells and averages the values in overlapping regions. By combining a VLMap with a code-writing LLM, the authors demonstrate spatial goal navigation using landmarks (e.g., move to the plant) or spatial references with respect to landmarks (between the keyboard and the bowl). Semantic Abstraction (SemAbs) (Ha and Song, 2022) presents another approach for 3D scene understanding by decoupling visual-semantic reasoning and 3D reasoning. In SemAbs, given an RGB-D image of a scene, a semantic-aware 2D VLM extracts 2D relevancy maps for each queried object, while semantic-abstracted 3D modules predict the 3D occupancy of each object using the relevancy maps. Because the 3D modules are trained irrespective of the specific object labels, the system demonstrates strong generalization capabilities, including generalization to new object categories and from simulation to the real world.

Current VLMs can reason about 2D images; however, they are not grounded in the 3D world. The main challenge for building 3D VLM foundation models is the scarcity of 3D data. Particularly, 3D data paired with language description is scarce. One strategy to circumvent this issue is to take advantage of 2D models trained on large-scale data to supervise 3D models. For instance, the authors of FeatureNeRF (Ye et al., 2023a) propose to learn 3D semantic representations by distilling 2D vision foundation models (i.e., DINO or Latent Diffusion) into 3D space via neural rendering. FeatureNeRF predicts a continuous 3D semantic feature volume from a single or few images, which can be used for downstream tasks such as key-point transfer or object part co-segmentation.

In 3D-LLM (Hong et al., 2023), the authors propose to use 2D VLMs as backbones to train a 3D-LLM that can take 3D representations (i.e., 3D point clouds with their features) as inputs and accomplish a series of diverse 3D-related tasks. The 3D features are extracted from 2D multi-view images and mapped to the feature space of 2D pretrained VLMs. To overcome 3D data scarcity, the authors propose an efficient prompting procedure for ChatGPT to generate 3D-language data encompassing a diverse set of tasks. These tasks include 3D captioning, dense captioning, 3D

question answering, 3D task decomposition, 3D grounding, 3D-assisted dialog, and navigation. Also, to capture 3D spatial information, the authors propose a 3D localization mechanism by (1) augmenting 3D features with position embedding and (2) augmenting LLM vocabularies with 3D location tokens. In the first part, the position embeddings of the three dimensions are generated and concatenated with 3D features. In the second part, the coordinates of the bounding box representing the grounded region are discretized to voxel integers as location tokens $<x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max}>$. It is important to highlight that, typically, creating 3D representations necessitates the use of 2D multi-view images and camera matrices. These resources are not as readily available as the vast amounts of internet-scale text and image data that current foundation models are trained on.

In general, integrating these high-fidelity scene representations such as LERF (Kerr et al., 2023) into robotics is challenging due to prolonged training and rendering times, as well as a lack of capability to accurately update changes in dynamic scenes. Unlike NeRF, which requires sampling-based neural network evaluations, Gaussian Splatting (GS) (Kerbl et al., 2023) provides efficient real-time rendering. Gaussian Splatting is a recently developed technique for 3D scene representation and rendering using 3D Gaussian functions and fast rasterization for image rendering. Besides real-time rendering performance, GS provides real-time scene editing, a beneficial capability for robotics applications with visual feedback. Recent studies (Hu et al., 2024; Liao et al., 2024; Qin et al., 2023; Shorinwa et al., 2024; Zhou et al., 2023; Zuo et al., 2024) have investigated the integration of semantic features into 3D Gaussian Splatting. Roboticists have started using these language-embedded fields for open-vocabulary robot planning, as discussed in Section 3.6.3.

4.3.2. Scene editing. When an embodied agent relies on an implicit representation of the world, the capability to edit and update this representation enhances the robot's adaptability. For instance, consider a scenario where a robot utilizes a pretrained NeRF model of an environment for navigation and manipulation. If a portion of the environment changes, being able to adjust the NeRF without retraining the model from scratch saves time and resources.

In the case of NeRFs, Wang et al. (2022) propose a text and image-driven method for manipulating NeRFs called CLIP-NeRF. This approach uses CLIP to disentangle the dependence between shape and appearance in conditional neural radiance fields. CLIP-NeRF facilitates the editing of the shape and appearance of NeRFs using either image or text prompts. It is composed of two modules: the disentangled conditional NeRF and CLIP-driven manipulation. The former takes the positional encoding $\gamma(x, y, z)$, a shape code z_s, viewing direction $v(\phi, \theta)$, and appearance code z_a as an input and outputs color and density. The disentanglement is achieved using a deformation network that is appended as

input to the traditional NeRF MLP that produces density, and by taking the output from this MLP and concatenating it with an appearance code to attain the color value. The CLIP-driven manipulation module takes an image example or text prompt as an input and outputs a shape deformation Δz_s and an appearance deformation Δz_a, from shape mapping and appearance mapping MLPs, respectively. These deformation values aim to perturb the shape code and appearance code in the disentangled conditional NeRF module to produce the desired output.

A key limitation of the CLIP-NeRF approach is that prompting can impact the entire scene rather than a selected region. For example, prompting to change the color of a flower's petals might also impact the shape and color of its leaves. To address this limitation, Kobayashi et al. propose to train distilled feature fields (DFFs) (Kobayashi et al., 2022) and then manipulate DFFs through query-based scene decomposition and editing. Pre-trained 2D VLMs such as LSeg (Li et al., 2022a) and DINO (Caron et al., 2021) are employed as teacher networks and distilled into 3D distilled feature fields via volume rendering. Editing is achieved by alpha compositing the density and color values of the two NeRF scenes. When combined with CLIP-NeRF, this method enables CLIP-NeRF to selectively edit specific regions of multi-object scenes. Tschernezki et al. (2022) explore a similar approach, showing that enforcing the 3D consistency of features in the NeRF embedding improved segmentation performance compared to using features from the original 2D images.

Another approach to more controlled 3D scene editing is to use structured 3D scene representations. Nerflets (Zhang et al., 2023c) represent a 3D scene as a combination of local neural radiance fields where each maintains its own spatial position, orientation, and dimension. Instead of employing a single large MLP to predict colors and densities as standard NeRF, individual Nerflets are combined to predict these values, modulated by their weights. After optimizing posed 2D images and segmentations, Nerflets reflect the decomposed scene and support more controlled editing.

One application of image editing in robotics is for data augmentation during policy learning. ROSIE (Yu et al., 2023) uses the Imagen editor (Wang et al., 2023c) to modify training images to add additional distractors and unseen objects and backgrounds to train robust imitation learning policies. GenAug (Chen et al., 2023f) similarly generates images with in-category and cross-category object substitutions, visual distractors, and diverse backgrounds. The CACTI (Mandi et al., 2022) pipeline includes a step inpainting different plausible objects via Stable-Diffusion (Rombach et al., 2022) onto training images. These approaches generate photorealistic images for training robust policies; however, generating images with sufficient diversity while also maintaining physical realism, for example, for object contacts, remains a challenge. Existing approaches use learned or provided masks to specify areas of the image to keep, or heuristics based on the particular robotic task.

Another direction is to use generative models to define goal images for planning. DALL-E-Bot (Kapelyukh et al., 2023) uses DALL-E 2 to define a goal image of human-like arrangements from observations.

4.3.3. Object representations. Learning correspondences between objects can facilitate manipulation by enabling skill transfer from trained objects to novel object instances in known categories or novel object categories at test time. Traditionally, object correspondences have been learned using strong supervision, such as keypoints and keyframes. Neural descriptor fields (NDFs) (Simeonov et al., 2022) remove the need for dense annotation by leveraging layer-wise activations from an occupancy network; however, this approach still requires many training shapes for each target object category. Additional works have started to build object representations directly from image features of pretrained vision models. Feature Fields for Robotic Manipulation (F3RM) (Shen et al., 2023c) builds on DFF to develop scene representations that support finding corresponding object regions. F3RM uses a similar feature representation for 6-DoF poses relative to objects (e.g., a grasp on the handle of the mug) to NDF. Besides allowing corresponding 6-DoF poses to be found from a few demonstrations, the pose embeddings can also be directly compared to text embeddings from CLIP to leverage language guidance (e.g., pick up the bowl). Correspondences between objects have also been directly extracted from DINO features (Goodwin et al., 2022a) without training. This method first extracts dense ViT feature maps of two objects using multiple views. Similar regions on the two objects are found by computing the cyclical distance metric (Goodwin et al., 2022b) on the feature maps. With the 2D patch correspondences, a 7-D rigid body transform (i.e., a SO(3) pose, a translation, and a scaling scalar) between the objects can be solved together with RANSAC and Umeyama's method (Umeyama, 1991).

4.4. Learned affordances

Affordances refer to the potential of objects, environments, or entities to offer specific functions or interactions to an agent. They can include actions such as pushing, pulling, sitting, or grasping. Detecting affordances bridges the gap between perception and action.

Affordance Diffusion (Ye et al., 2023b) synthesizes complex interactions of, for example, an articulated hand with a given object. Given an RGB image, Affordance Diffusion aims to generate images of human hands for hand-object interaction (HOI). The authors propose a two-step generative approach based on large-scale pretrained diffusion models based on where to interact (layout) and how to interact (content). The layout network generates a 2D spatial arrangement of hand and object. The content network then synthesizes images of a hand grasping the object conditioned on the given object and the sampled HOI layout. Affordance Diffusion outputs both hand articulation and approach orientation.

Vision-Robotic Bridge (VRB) (Bahl et al., 2023) trains a visual affordance model on internet videos of human behavior. Particularly, it estimates the likely location and manner in which a human interacts within a scene. This model captures the structural information of these behavioral affordances. The authors seamlessly integrate the affordance model with four different robot learning paradigms. Firstly, they apply offline imitation learning, where the robot learns by imitating the observed human interactions from the videos. Secondly, they use exploration techniques to enable the robot to actively discover and learn new affordances in its environment. Thirdly, the authors incorporate goal-conditioned learning, allowing the robot to learn how to achieve specific objectives by leveraging the estimated affordances. Finally, they integrate action parameterization for reinforcement learning, enabling the robot to learn complex behaviors by optimizing its actions based on the estimated affordances.

4.5. Predictive models

Predictive dynamics models, or world models, predict how the state of the world changes given particular agent actions, that is, they attempt to model the state transition function of the world (Sun et al., 2021b). When applied to visual observations, dynamics modeling can be formulated as a video prediction problem (Sun et al., 2022b; 2023b). While video generation and prediction, particularly over long horizons, is a longstanding challenge with many prior efforts, recent models based on vision transformers and diffusion models have demonstrated improvements (Janner et al., 2022; Sun et al., 2023a). For instance, the Phenaki model (Villegas et al., 2023) generates variable length video up to minutes in length conditioned on text prompts.

Several approaches apply these models to robotics in the literature. Note that while learned dynamics or world models in robotics have been explored in constrained or smaller-data regimes, we focus in this section on works that train on a diversity or volume of data that is characteristic of foundation models. One strategy is to learn an action-conditioned model that may be used directly for downstream planning by optimizing an action sequence (Dasari et al., 2019), for example, performing model-predictive control, or for policy learning via training on simulated rollouts. One example is the GAIA-1 model, which generates predictions of driving video conditioned on arbitrary combinations of video, action, and text (Hu et al., 2023a). It was trained on 4700 hours of proprietary driving data.

Another approach is to use a video prediction model to generate a plan of future states, and then learn a separate goal-conditioned policy or inverse dynamics model to infer control actions based on the current and target state. One line of work instantiates this by combining text-conditioned video diffusion models with image-goal-conditioned policies to solve manipulation tasks in simulated and real tabletop settings (Du et al., 2023c). This approach has been extended to longer-horizon object manipulation tasks by

using the PaLM-E VLM to break down a high-level language goal into smaller substeps, leveraging feedback between the VLM and video generation models (Du et al., 2023d). Another example is COMPASS (Ma et al., 2022), which first constructs a comprehensive multimodal graph to capture crucial relational information across diverse modalities. The graph is then used to construct a rich spatio-temporal and semantic representation. Pretrained on the TartanAir multimodal dataset, COMPASS was demonstrated to address multiple robotic tasks, including drone navigation, vehicle racing, and visual odometry.

4.6. Challenges and perspectives

While significant advances have been made toward generalizable methods for perception via foundation models, many challenges remain. For open-vocabulary object detection and 3D classification, the granularity at which objects should be detected and whether object parts should be considered separate or combined can often lead to incompatibilities with the rest of the system. Additionally, semantic alignment (e.g., prompting) is often still unintuitive and practically challenging when performing segmentation for arbitrary in-the-wild objects. Efficiently and consistently updating 3D scene representations, including implicit scene representations, is critical for dynamic robotic tasks while existing works primarily focus on static scenes and grasping tasks. Existing affordance models trained on large-scale data, largely due to data quality and quantity limitations and embodied gaps, mostly provide information on how a robot should make initial contact with an object rather than how it should be manipulated. In the realm of predictive models, there are few to no open-sourced predictive models trained on large-scale robotic datasets, due to data scarcity and computational cost. Training video models on diverse datasets remains generally challenging, although models such as OpenAI's Sora (Brooks et al., 2024) provide a glimpse of future possibilities.

5. Embodied AI

Recently, researchers have shown that the success of LLMs can be extended to embodied AI domains (Brohan et al., 2023b; Huang et al., 2022b; Liang et al., 2023; Zeng et al., 2022), where "embodied" typically refers to a virtual embodiment in a world simulator, not a physical robot embodiment. Statler (Yoneda et al., 2023) is a framework that endows LLMs with an explicit representation of the world state as a form of "memory" that is maintained over time. Statler uses two instances of general LLMs: a world-model reader and a world-model writer, that interface with and maintain the world state. Statler improves the ability of existing LLMs to reason over longer time horizons without the constraint of context length.

Large Scale Language Models (LSLMs) have exhibited strong reasoning ability and the ability to adapt to new tasks through in-context learning. Dasgupta et al. (2022) combine

these complementary abilities in a single system consisting of three parts: a Planner, an Actor, and a Reporter. The Planner is a pretrained language model that can issue commands to a simple embodied agent (the Actor), while the Reporter communicates with the Planner to inform its next command. Mu et al. (2023) build EgoCOT, a dataset consisting of carefully selected videos from the Ego4D dataset, along with corresponding high-quality language instructions. EmbodiedGPT (Mu et al., 2023) utilizes prefix adapters to augment the 7B language model's capacity to generate high-quality planning, training it on the EgoCOT dataset to avoid overly divergent language model responses. Comprehensive experiments were conducted, demonstrating that the model effectively enhances the performance of embodied tasks such as Embodied Planning, Embodied Control, Visual Captioning, and Visual Q&A. Embodied agents should autonomously and endlessly explore the environment. They should actively seek new experiences, acquire new skills, and improve themselves.

The game of Minecraft (Engelbrecht and Schiele, 2014) provides a platform for designing intelligent agents capable of operating in the open world. MineDojo (Fan et al., 2022) is a framework for developing generalist agents in the game of Minecraft. MineDojo offers thousands of open-ended and language-prompted tasks, where the agent can navigate in a progressively generated 3D environment to mine, craft tools, and build structures. As part of this work, the authors introduce MiniCLIP, a video-language model that learns to capture the correlations between a video clip and its time-aligned text that describes the video. The MineCLIP model, trained on YouTube videos, can be used as a reward function to train the agent with reinforcement learning. By maximizing this reward function, it incentivizes the agent to make progress toward solving tasks specified in natural language.

Voyager (Wang et al., 2023b) introduces an LLM-powered embodied lifelong learning agent in the realm of Minecraft. Voyager uses GPT-4 to continuously explore the environment. It interacts with GPT-4 through in-context prompting and does not require model parameter fine-tuning. Exploration is maximized by querying GPT-4 to provide a stream of new tasks and challenges based on the agent's interaction history and current situations. Also, the iterative prompting mechanism generates code as the action space to control the Minecraft agent. Iterative prompting incorporates environment feedback provided by Minecraft, execution errors, and a self-verification scheme. For self-verification, GPT-4 acts as a critic by checking task success and providing suggestions for task completion in the case of failure. The GPT-4 critic can be replaced by a human critic to provide on-the-fly human feedback during task execution. Ghost in the Minecraft (GITM) (Zhu et al., 2023) leverages LLM to break down goals into sub-goals and map them to structured actions for generating control signals. GITM consists of three components: an LLM Decomposer, an LLM Planner, and an LLM Interface. The LLM Decomposer is responsible for dividing the given Minecraft

goal into a sub-goal tree. The LLM Planner then plans an action sequence for each sub-goal. Finally, the LLM Interface executes each action in the environment using keyboard and mouse operations.

Reinforcement learning in embodied AI virtual environments has the potential to improve the capabilities of real-world robotics by providing efficient training and optimizing control policies in a safe and controlled setting. Reward design is a crucial aspect of RL that influences the robot's learning process. Rewards should be aligned with the task's objective and guide the robot to achieve the desired task. Foundation models can be leveraged to design rewards. Kwon et al. (2023) investigate the simplification of reward design by utilizing a large language model (LLM), such as GPT-3, as a proxy reward function. In this approach, users provide a textual prompt that contains a few examples (few-shots) or a description (zero-shot) of the desired behavior. The proposed method incorporates this proxy reward function within a reinforcement learning framework. Users specify a prompt at the start of the training process. During training, the LLM evaluates the RL agent's behavior against the desired behavior outlined in the prompt, resulting in a corresponding reward signal generated by the LLM. Subsequently, the RL agent employs this reward to update its behavior through the learning process.

In Du et al. (2023b), the authors propose a method called Exploring with LLMs (ELLM) that rewards an agent for achieving goals suggested by a language model. The language model is prompted with a description of the agent's current state. Therefore, without having a human in the loop, ELMM guides agents toward meaningful behavior.

Zhang et al. (2022c) explore the potential relationship between offline reinforcement learning and language modeling. They hypothesize that RL and LM share similarities in predicting future states based on current and past states, considering both local and long-range dependencies across states. To validate this assumption, the authors pretrain Transformer models on different offline RL tasks and assess their performance on various language-related tasks. Tarasov et al. (2022) present an approach to harness pretrained language models in deep offline reinforcement learning scenarios that are not inherently compatible with textual representations. The authors suggest a method that involves transforming the RL states into human-readable text and performing fine-tuning of the pretrained language model during training with deep offline RL algorithms.

Advances in model architecture (e.g., transformer) for foundation models allow the model to effectively model and predict sequences. To harness the power of these models, some recent studies investigate exploiting these architectures for sequence modeling in RL problems. Reid et al. (2022) explore the potential of leveraging the sequence modeling formulation of reinforcement learning and examine the transferability of pretrained sequence models across different domains, such as vision and language. They specifically focus on the effectiveness of fine-tuning these pretrained models on offline RL tasks, including control and

games. In addition to investigating the transferability of pretrained sequence models, the authors propose techniques to enhance the transfer of knowledge between these domains. These techniques aim to improve the adaptability and performance of the pretrained models when applied to new tasks or domains.

High-level task planning using LLMs is demonstrated in embodied AI environments. Huang et al. (2022a) propose employing pretrained Language Models (LMs) as zero-shot planners. The approach is evaluated in the VirtualHome (Puig et al., 2018) environment. In this work, first, an autoregressive LLM such as GPT-3 (Brown et al., 2020) or Codex (Chen et al., 2021) is quarried to generate action plans for high-level tasks. Some of these action plans might not be executable by the agent due to ambiguity in language or referring to objects that are not present or grounded in the environment. So, to select the admissible action plans, admissible environment actions, and generated actions by the causal LLM are embedded using a BERT-style LM. Then for each admissible environment action, its semantic distance to the generated action is computed using cousin similarity.

Chain of thought reasoning and action generation are proposed for embodied agents as well. ReAct (Yao et al., 2023) combines reasoning (e.g., chain of thought) and acting (e.g., sequence of action generation) within LLM. Reasoning traces enhance the model's ability to deduce, monitor, and revise action plans, along with managing exceptions effectively. Actions facilitate interaction with external resources, like knowledge bases or environments, enabling it to acquire supplementary information. ReAct showcases its proficiency across a wide array of language and decision-making tasks, including question-answering and fact verification. It enhances interpretability and trust for users by transparently illustrating the process through which it searches for evidence and formulates conclusions. Unlike prior methods that depend on a singular chain of thought, ReAct engages with a Wikipedia API for pertinent information retrieval and belief updating. This strategy effectively mitigates the issues commonly associated with chain-of-thought reasoning, such as hallucination and error propagation.

VPT (Baker et al., 2022) presents video pretraining in which the agent learns to act by watching unlabeled online videos. It is shown that an inverse dynamic model can be trained with a small labeled dataset and the model can be used to label a huge unlabeled data of the internet. Videos of people who have played Minecraft are used to train an embodied AI agent to play Minecraft. The model exhibits zero-shot performance and can be fine-tuned for more complex skills using imitation learning or reinforcement learning. The VPT model is trained with a standard behavioral cloning loss (9) (negative log-likelihood) while the actions are drawn from the inverse dynamic model. Lin et al. (2023a) presents an agent, Dynalang, that learns a multi-modal world model to predict the future text and image representations. The world model is trained using a replay buffer of the agent's past interactions with the environment. Actions are selected by training an actor-critic model on a sequence of representations generated by the world model.

5.1. Generalist AI

A long-standing challenge in robotics research is deploying robots or embodied AI agents in various non-factory real-world applications to perform a range of tasks. To make generalist robots that can operate in diverse environments with diverse tasks, some researchers have proposed generative simulators for robot learning. For example, Generative Agents (Park et al., 2023a) discusses how generative agents can produce realistic imitations of human behavior for interactive applications, creating a miniature community of agents similar to those found in games like The Sims. The authors connect their architecture with the ChatGPT large language model to create a game environment with 25 agents. The study includes two evaluations, a controlled evaluation and an end-to-end evaluation, which demonstrate the causal effects of the various components of their architecture. Xian et al. (2023) propose a fully automated generative pipeline, known as a generative simulation for robot learning, which utilizes models to generate diverse tasks, scenes, and training guidance on a large scale. This approach can facilitate the scaling up of low-level skill learning, ultimately leading to a foundational model for robotics that empowers generalist robots.

An alternative method for developing generalist AI involves using generalizable multi-modal representations. Gato (Reed et al., 2022) is a generalist agent that works as a multi-modal, multi-task, multi-embodiment generalist policy. Using the same neural network with the same set of weights, Gato can sense and act with different embodiments in various environments across different tasks. Gato can play Atari, chat, caption images, stack blocks with a real robot arm, navigate in a 3D simulated environment, and more. Gato is trained on 604 different tasks with various modalities, observations, and actions. In this setting, language acts as a common grounding across different embodiments. Gato has 1.2B parameters and is trained offline in a supervised way. Positioned at the confluence of representation learning and reinforcement learning (RL), RRL (Shah and Kumar, 2021) learns behaviors directly from proprioceptive inputs. By harnessing pre-trained visual representations, RRL is able to learn from visual inputs, which typically pose challenges in conventional RL settings.

5.2. Simulators

High-quality simulators or benchmarks are crucial for robotics development. Hence, we put the "simulator" section here to highlight its essential role. To facilitate generalization from simulation to the real world, Gibson (Xia et al., 2018) emphasizes real-world perception for embodied agents. To bridge the gap between simulation and real-world,

iGibson (Li et al., 2021) and BEHAVIOR-1K (Li et al., 2022b) further support the simulation of a more diverse set of household tasks and reach high levels of simulation realism. As a simulation platform for research in Embodied AI, Habitat (Savva et al., 2019) consists of Habitat-Sim and Habitat-API. Habitat-Sim can achieve several thousand frames per second (fps) running single-threaded. Rather than modeling into low-level physics, Habitat-Lab (Szot et al., 2021), is a high-level library for embodied AI, giving a modular framework for end-to-end development. It facilitates the definition of embodied AI tasks, such as navigation, interaction, instruction following, and question answering. Additionally, it enables the configuration of embodied agents, encompassing their physical form, sensors, and capabilities. The library supports various training methodologies for these agents, including imitation learning, reinforcement learning, and traditional non-learning approaches like the SensePlanAct pipelines. Furthermore, it provides standard metrics for evaluating agent performance across these tasks. In line with this, the recent release of Habitat 3.0 (Puig et al., 2023) further expands these capabilities.

Similarly, RoboTHOR (Deitke et al., 2020) serves as a platform for the development and evaluation of embodied AI agents, offering environments in both simulated and physical settings. Currently, RoboTHOR includes a training and validation set comprising 75 simulated scenes. Additionally, there are 14 scenes each for test-dev and test-standard in the simulation, with corresponding physical counterparts. Key features of RoboTHOR include its reconfigurability and benchmarking capabilities. The physical environments are constructed using modular, movable components, enabling the creation of diverse scene layouts and furniture configurations in a single physical area. Another simulator, VirtualHome (Puig et al., 2018), models complex activities that occur in a typical household. It supports program descriptions for a variety of activities that happen in people's homes. Huang et al. (2022b) use VirtualHome to evaluate the robot planning ability with language models. These simulators have the potential to be applied for evaluating LLMs on robotics tasks.

5.3. Challenges and perspectives

As robot learning in the real world can lead to unsafe scenarios, embodied AI facilitates adaptable robot learning through interactions with objects and scenes in virtual environments. Developing high-fidelity, physics-based 3D simulators that are highly photorealistic and accurately represent real-world counterparts can accelerate robot learning and lead to generalized robotic solutions capable of interacting with diverse objects and scenarios. Thus, a natural extension of embodied AI research in robotics applications involves replacing game environments with high-fidelity physics-based simulators. Current embodied AI simulators, as discussed in 5.2, are not physics-based. Therefore, efforts should be directed towards developing

these 3D simulators to include physics-based modeling. Additionally, embodied AI enables long-horizon planning and lifelong learning for robots. For example, analogous to the Voyager agent that learns to play in the open-world environment of Minecraft game, a robot can learn long-horizon planning in an open-world environment.

6. Challenges and future directions

In this section, we examine challenges related to integrating foundation models into robotics settings. We also explore potential future avenues to address some of these challenges.

6.1. Overcoming data scarcity in training foundation models for robotics

One main challenge is that compared to the internet-scale text and image data that large models are trained on, robotic-specific data is scarce. We discuss various techniques to overcome data scarcity. For example, to scale up robot learning, some recent works suggest the use of play data instead of expert data for imitation learning. Another technique is data augmentation using in-painting techniques.

6.1.1. Scaling robot learning using unstructured play data and unlabeled videos of humans. Language-conditioned learning, such as language-conditioned behavioral cloning or language-conditioned affordance learning, requires having access to large annotated datasets. To scale up learning, in Play-LMP (Lynch et al., 2020), the authors suggest using teleoperated human-provided play data instead of fully annotated expert demonstrations. Play data is unstructured, unlabeled, cheap to collect, but rich. Collecting play data does not require scene staging, task segmenting, or resetting to an initial state. Also, in MimicPlay (Wang et al., 2023a), a goal-conditioned trajectory generation model is trained based on human-play data. The play data includes unlabeled video sequences of humans interacting with the environment with their hands. Recently works such as Mees et al. (2023) have shown a very small percentage (as little as 1%) of language-annotated data is needed to train a visuo-lingual affordance model for robot manipulation tasks.

6.1.2. Data augmentation using inpainting. Collecting robotics data requires the robot to interact with the real physical world. This data collection process can be associated with significant costs and potential safety concerns. One way to tackle this challenge is to use generative AI, such as text-to-image diffusion models, for data augmentation. For example, ROSIE (Scaling Robot Learning with Semantically Imagined Experience) (Yu et al., 2023) presents a diffusion-based data augmentation. Given a robot manipulation dataset, they use inpainting to create various unseen objects, backgrounds, and distractors with textual guidance. One important challenge for these methods is

developing inpainting strategies that can generate sufficient semantically and visually diverse data, while at the same time ensuring that this data is physically feasible and accurate. For instance, using inpainting to modify an image of an object within a robot's gripper may result in an image with a physically unrealistic grasp, leading to poor downstream training performance. Additional investigation into generative foundation models that are evaluated not only for visual quality but also for physical realism may improve the generality of these methods.

6.1.3. Overcoming 3D data scarcity for training 3D foundation models. Currently, multi-modal Vision-and-Language Models (VLMs) can analyze 2D images, but they lack a connection to the 3D world, which encompasses 3D spatial relationships, 3D planning, 3D affordances, and more. The primary obstacle in developing foundational 3D VLM models lies in the scarcity of 3D data, especially data that is paired with language descriptions. As discussed, language-driven perception tasks such as language-driven 3D scene representation, language-driven 3D scene editing, language-driven 3D scene or shape generation, language-driven 3D classification, and affordance prediction require access to 3D data or multi-view images with camera matrices which are not readily available data types. New datasets or data generation methods need to be created in the future to overcome data scarcity in the 3D domain.

6.1.4. Synthetic data generation via high-fidelity simulation. High-fidelity simulation via gaming engines can provide an efficient means to collect data, especially to solve multimodal and 3D perception tasks on robots. For example, TartanAir (Wang et al., 2020), a dataset for robot navigation tasks, was collected in Shah et al. (2017) with the presence of moving objects, changing light, and various weather conditions. By collecting data in simulations, it was possible to obtain multi-modal sensor data and precise ground truth labels such as the stereo RGB image, depth image, segmentation, optical flow, camera poses, and Li-DAR point cloud. A large number of environments were set up with various styles and scenes, covering challenging viewpoints and diverse motion patterns that are difficult to achieve by using physical data collection platforms. An extension TartanAir-V2 (https://tartanair.org) furthers the dataset by incorporating additional environments and modalities, such as fisheye, panoramas, and pinholes, with arbitrary camera intrinsic and rotations.

6.1.5. Sim-to-real transfer. Robotics policies trained in simulated environments can be transferred to and deployed on robots in the real world. While sim-to-real transfer holds promise for addressing data scarcity in robotics, the sim-to-real gap poses a significant challenge for foundation models trained using simulated data. There are examples before the appearance of foundation models that sim-to-real has been studied in Margolis and Agrawal (2023) propose the multiplicity of behavior (MoB) to promote policy generalization by conditioning on a behavior parameter. These policies are trained with tuned auxiliary reward components that promote sim-to-real transfer. Kumar et al. (2021) focuses on real-time policy adaptation by training an adaptation module to estimate an extrinsic vector that encodes the environment the robot is operating in. To reduce the sim-to-real gap, Kumar et al. (2021) introduces an additional component of the reward function in addition to altering the terrain of the physics simulator used for training. Chen et al. (2023b) seek to train policies for in-hand reorientation of novel objects by training a teacher policy with reinforcement learning in a physics simulator and a student policy through imitation learning. To reduce the sim-to-real gap the student policies take as input a point cloud from a partial rendering from the physics simulator. Methodologies from these works can be used to guide sim-to-real from the world of foundation models.

6.1.6. Data augmentation using VLMs. Data augmentation can be provided using Visual-Language Models (VLMs). In DIAL (Xiao et al., 2023a), Data-driven Instruction Augmentation for Language-conditioned control is introduced. DIAL uses VLM to label offline datasets for language-conditioned policy learning. DIAL performs instruction augmentation using VLMs to weakly relabel offline control datasets. DIAL consists of three steps: (1) Contrastive fine-tuning of a VLM such as CLIP (Radford et al., 2021a) on a small robot manipulation dataset of trajectories with crowd-sourced annotation, (2) producing new instruction labels by using the fine-tuned VLM to score relevancy of crowd-sourced annotations against a larger dataset of trajectories, and (3) training a language-conditioned policy using behavior cloning on both, the original and re-annotated dataset.

6.1.7. Robot physical skills are limited to distribution of skills. One key limitation of the existing robot transformers and other related works in robotics is that robot physical skills are limited to the distribution of skills observed within the robot data. Using these transformers, the robot lacks the capability to generate new movements. To address this constraint, an approach involves using motion data from videos that humans performing various tasks. The inherent motion information within these videos can then be employed to facilitate the acquisition of physical skills in robotics.

6.2. Real time performance (high inference time of foundation models)

Another bottleneck for deploying foundation models on robots is the inference time of these models. In Table 2, the inference time for some of these models is reported. As seen, the inference time for some of the models still needs to be improved for reliable real-time deployment of the robotic systems. As real-time capability is an essential requirement for any robotic system, more research needs to be performed

to improve the computational efficiency of foundation models.

Furthermore, foundation models are most often stored and run in remote data centers, and accessed through APIs that require network connectivity. Many foundation models (e.g., the GPT models, the Dall-E models) can only be accessed this way, while others are usually accessed this way, but can also be downloaded and run locally with sufficient local computing power, such as SAM (Kirillov et al., 2023), LLaMA (Touvron et al., 2023b), and DINOv2 (Oquab et al., 2023). Given this cloud-service paradigm, the latencies and service times in response to an API call for a foundation model depend on the underlying network over which the data is routed and the data center where the computation takes place—factors that are beyond the control of a robot. So network reliability should be taken into account before integrating a foundation model into a robot's autonomy stack.

For some robotics domains, reliance on the network and 3rd party computing may not be a safe or realistic operating paradigm. In autonomous driving, autonomous aircraft, search and rescue or emergency response applications, and defense applications the robot cannot rely on network connectivity for time-critical perception or control computations. One option is to have a safe fall-back mode that relies on classical autonomy tools using only local computation, which can take over if access to the cloud is interrupted for some reason. Another potential longer-term solution for network-free autonomy is the distillation of large foundation models into smaller-sized specialized models that run on onboard robot hardware. Some recent work has attempted this approach (though without an explicit link to robotics) (Lin et al., 2023b). Such distilled models would likely give up some aspect of the full model, for example, restricting operation to a certain limited context, in exchange for smaller size and faster computation. This could be an interesting future direction for bringing the power of foundation models to safety-critical robotics systems.

6.3. Limitations in multimodal representation

Multimodal interaction implicitly assumes that the modality is tokenizable and can be standardized into input sequences without losing information. The Multimodal models provide information sharing between multiple modalities and are some variation of multimodal transformers with cross-modal attention between every pair of inputs. In multimodal representation learning, it is assumed that cross-modal interactions and the dimension of heterogeneity between different modalities can all be captured by simple embeddings. In other words, a simple embedding is assumed to be sufficient to identify the modality or, for example, how different language is from vision. In the realm of multimodal representation learning, the question of whether a single multimodal model can accommodate all modalities remains an open challenge.

Additionally, when paired data between a modality and text is available one can embed that modality into text directly. In robotics applications, there are some modalities for which sufficient data is not available, and to be able to align them with other modalities, they need to be first converted to other modalities and then used. For example, 3D point cloud data have various applications in robotics, but training a foundation model using this type of data is challenging since data is scarce and is not aligned with text. So, one way to overcome this challenge is first converting this 3D point cloud data to other modalities such as images and, subsequently, images to text as the secondary step of alignment. Then they can be used in foundation model training. As another example, in Socratic models (Zeng et al., 2022), each modality, whether visual or auditory, is initially translated into language, after which language models attempt to respond to these modalities.

6.4. Uncertainty quantification

How can we provide assurances on the reliability of foundation models when they are deployed in potentially safety-critical robotics applications (Sun et al., 2023a)? Current foundation models such as LLMs often *hallucinate*, for example, produce outputs that are factually incorrect, logically inconsistent, or physically infeasible. While such failures may be acceptable in applications where the outputs from the model can be checked by a human in real-time (e.g., as is often the case for LLM-based conversational agents), they are not acceptable when deploying autonomous robots that use the outputs of foundation models in order to act in human-centered environments. Rigorous uncertainty quantification is a key step toward addressing this challenge and safely integrating foundation models into robotic systems. Below, we highlight challenges and recent progress in uncertainty quantification for foundation models in robotics.

6.4.1. Instance-level uncertainty quantification. How can we quantify the uncertainty in the output of a foundation model for a *particular* input? As an example, consider the problem of image classification; given a particular image, one may quantify uncertainty in the output by producing a *set* of object labels that the model is uncertain among or a *distribution* over object labels. Instance-level uncertainty quantification can inform the robot's decisions at runtime. For example, if an image classification model running on an autonomous vehicle produces a prediction set {Pedestrian, Bicyclist} representing that it is uncertain whether a particular agent is a pedestrian or a bicyclist, the autonomous vehicle can take actions that consider both possibilities.

6.4.2. Distribution-level uncertainty quantification. How can we quantify the uncertainty in the correctness of a foundation model that will be deployed on a *distribution* of possible future inputs? For the problem of image

classification, one may want to compute or limit the probability of errors over the distribution of inputs that a robot may encounter when deployed. Distribution-level uncertainty quantification allows us to decide whether a given model is sufficiently reliable to deploy in our target distribution of scenarios. For example, we may want to collect additional data or fine-tune the model if the computed probability of error is too high.

6.4.3. Calibration. In order to be useful, estimates of uncertainty (both at the instance-level and distribution level) should be *calibrated*. If we perform instance-level uncertainty quantification using prediction sets, calibration asks for the prediction set to contain the true label with a user-specified probability (e.g., 95%) over future inputs. If instance-level uncertainty is quantified using a distribution over outputs, it should be the case that outputs that are assigned confidence p are, in fact, correct with probability p over future inputs. Similarly, distribution-level uncertainty estimates should limit the true probability of errors when encountering inputs from the target distribution.

We highlight a subtle but important point that is often overlooked when performing uncertainty quantification in robotics: it can be crucial to pay attention to the distinction between Frequentist and Bayesian interpretations of probabilities. In many robotics contexts—particularly safety-critical ones—the desired interpretation is often Frequentist in nature. For example, if we produce a bound ϵ for the probability of collision of an autonomous vehicle, this should limit the actual observed rate of collisions when the vehicle is deployed. Bayesian techniques (e.g., Gaussian processes or Bayesian ensembles) do not necessarily produce estimates of uncertainty that are calibrated in this Frequentist sense (since the estimates depend on the specific prior that is used to produce the estimates). Trusting the resulting uncertainty estimates may lead one astray if the goal is to provide statistical guarantees on the safety or performance of the robotic system when it is deployed.

6.4.4. Distribution shift. An important challenge in performing calibrated uncertainty quantification is *distribution shift*. A foundation model trained on a particular distribution of inputs may not produce calibrated estimates of uncertainty when deployed on a different distribution for a downstream task. A more subtle cause of distribution shift in robotics arises from *closed-loop* deployment of a model. For example, imagine an autonomous vehicle that chooses actions using the output of a perception system that relies on a pretrained foundation model; since the robot's actions influence future states and observations, the distribution of inputs the perception system receives can be potentially very different from the one it was trained on.

6.4.5. Case study: uncertainty quantification for language-instructed robots. Recently, there has been exciting progress in performing rigorous uncertainty quantification for language-instructed robots (Ren et al., 2023). This work proposes an approach called KNOWNO for endowing language-instructed robots with the ability to *know when they don't know* and to ask for help or clarification from humans in order to resolve uncertainty. KNOWNO performs both instance-level and distribution-level uncertainty quantification in a calibrated manner using the theory of *conformal prediction*. In particular, given a language instruction (and a description of the robot's environment generated using its sensors), conformal prediction is used to generate a prediction set of candidate actions. If this set is a singleton, the robot executes the corresponding action; otherwise, the robot seeks help from a human by asking them to choose an action from the generated set. Using conformal prediction, KNOWNO ensures that asking for help in this manner results in a statistically guaranteed level of task success (i.e., distribution-level uncertainty quantification). KNOWNO tackles potential challenges with distribution shift by collecting a small amount of calibration data from the target distribution of environments, tasks, and language instructions, and using this as part of the conformal prediction calibration procedure. While KNOWNO serves as an example of calibrated instance-level and distribution-level uncertainty quantification for LLMs, future research should also explore assessing and ensuring the reliability of various other foundation models, such as vision-language models, vision-navigation models, and vision-language-action models, commonly employed in robotics. In addition, exploring how Bayesian uncertainty quantification techniques, such as ensembling (Park et al., 2023b; Sun et al., 2022c), can be combined with approaches such as conformal prediction to produce calibrated estimates of instance-level and distribution-level uncertainty is a promising direction.

6.5. Safety evaluation

The problem of *safety evaluation* is closely related to uncertainty quantification. How can we rigorously *test* for the safety of a foundation model-based robotic system (i) before deployment, (ii) as the model is updated during its lifecycle, and (iii) as the robot operates in its target environments? We highlight challenges and research opportunities related to these problems below.

6.5.1. Pre-deployment safety tests. Rigorous pre-deployment testing is crucial for ensuring the safety of any robotic system. However, this can be particularly challenging for robots that incorporate foundation models. First, foundation models are trained on vast amounts of data; thus, a rigorous testing procedure should ensure that the model does not see test scenarios during training. Second, foundation models often commit errors in ways that are hard to predict a priori; thus, tests need to cover a diverse enough range of scenarios to uncover flaws. Third, foundation models such as LLMs are often used to produce open-ended outputs (e.g., a plan for a robot described in natural language). The correctness of such outputs can be challenging

to evaluate in an automated manner if these outputs are evaluated in isolation from the entire system.

The deployment cycle of current foundation models (in non-robotics applications) involves thorough *red-teaming* by human evaluators (Ganguli et al., 2022; OpenAI, 2023). Recent work has also considered partially automating this process by using foundation models themselves to perform red-teaming (Perez et al., 2022; Tong et al., 2023). Developing ways to perform red-teaming (both by humans and in a partially automated way) for foundation models in robotics is an exciting direction for future research.

In addition to evaluating the foundation model in isolation, it is also critical to assess the safety of the end-to-end robotic system. Simulation can play a critical role here and already does so for current field-deployed systems such as autonomous vehicles (Kusano et al., 2022; Webb et al., 2020). The primary challenges are to ensure that (i) the simulator has high enough fidelity for results to meaningfully transfer to the real world, and (ii) test scenarios (manually specified, replicated from real-world scenarios, or automatically generated via adversarial methods (Ding et al., 2023)) are representative of real-world scenarios and are diverse enough to expose flaws in the underlying foundation models. In addition, finding ways to augment large-scale simulation-based testing with smaller-scale real-world testing is an important direction for future work. We emphasize the need for performing such testing throughout the lifecycle of a field-deployed robotic system, especially as updates are made to different components (which may interact in unpredictable ways with foundation models).

6.5.2. Runtime monitoring and out-of-distribution detection.
In addition to performing rigorous testing *offline*, robots with foundation model-based components should also perform *runtime monitoring*. This can take the form of *failure prediction* in a given scenario, which can allow the robot to deploy a safety-preserving fallback policy (Luo et al., 2022; Farid et al., 2022a, 2022b; Hsu et al., 2023b, 2023a). Alternately, the robot can perform *out-of-distribution (OOD) detection* using experiences collected from a small batch of scenarios in a novel distribution (Cai and Koutsoukos, 2020; Farid et al., 2021; Greenberg and Mannor, 2021; Sinha et al., 2022); this can potentially trigger the robot to cease its operations and collect additional training data in the novel distribution in order to retrain its policy. Developing techniques that perform runtime monitoring and OOD detection with statistical guarantees on false positive/negative error rates in a data-efficient manner remains an important research direction.

6.5.3. Performance evaluation.
Given the high complexity of the robot policies themselves and in general foundation models, we anticipate a greater reliance on statistical performance evaluation methods that remain independent of the policy's complexity. For example, some early works in this area include Vincent et al. (2023, 2024).

6.6. Using existing foundation models "plug-and-play" vs. building new foundation models for robotics

To incorporate foundation models into robotics, either existing pretrained large models can be employed as plug-and-play, or new foundation models can be built using robotics data. Using foundation models "plug-and-play" refers to integrating foundation models into various applications without extensive customization. A large body of the existing literature on foundation models in robotics is centered around the use of foundation models from other domains, such as language or vision. The "plug-and-play" approach simplifies and facilitates the integration of recent AI advances into the robotics domain. However, these models are not always customized to specific applications. When specific domain expertise is needed, it is necessary to build a foundation model from scratch or fine-tune the existing models. Building a foundation model from scratch is resource-intensive and demands significant computational power. However, it provides fine-grained control over the architecture, training parameters, and overall behavior.

Below, we speculate on potential foundation models that could be built specifically for robotics applications in the future.

6.6.1. Incorporating tactile and audio sensing.
Tactile and audio sensing are less commonly utilized today but are critical for human manipulation. Because the data captured in these modalities is often very sensor-specific, it is challenging to deploy pretrained models and leverage prior datasets for a new hardware setup. Early efforts toward developing unified and grounded representations for touch and audio in the context of robotics (Clarke, 2019; Fu et al., 2024; Gao et al., 2022; Yang et al., 2024) may eventually lead to foundational models that can learn from and be deployed on heterogeneous hardware setups, making these modalities easy to integrate and unlocking their potential.

6.6.2. Foundation models for high-level reasoning and task planning.
In addition to building foundational models for low-level skills, another significant direction is the development of robot-specific models for planning. Planning problems in robotics often require reasoning over both discrete decisions and continuous actions, also known as the task and motion planning problem (Garrett et al., 2021; Zhao et al., 2024). Current solutions typically depend on known object models, computationally intensive search procedures, and domain-specific planning definitions created by experts. Recent methods have demonstrated that foundational models, fine-tuned on domain-specific data, can reason about the physical properties of objects (Gao et al., 2023), understand spatial relations in 3D (Chen et al., 2024), and address planning-related questions from videos (Sermanet et al., 2023). These reasoning abilities are all important for robot planning and can collectively contribute

to establishing a foundation model for long-horizon robot planning.

6.7. End-to-end vs. modular systems

It is an open question of how much structure should be imposed on a robotic system based on learning from large-scale data. Similarly, there is no definitive answer to whether a robotic system should be built on a single foundation model or composed of multiple modular models. On one side of the spectrum are end-to-end systems such as Robotics Transformers (Brohan et al., 2022; Zitkovich et al., 2023), which directly provide raw observations and task specifications to a neural network that outputs low-level actions such as changes in end-effector pose. Toward the other side are methods that build explicit modules with structured inputs and outputs. For instance, such a method may have a module specifically for perception that performs object detection, or action planning within a scene graph. Many learning-based robotic systems do not fall clearly into one of these categories; rather, they are hybrid approaches.

End-to-end approaches may yield stronger performance in the long term, particularly if the quantity of real-world robotics data grows rapidly. Because they avoid imposing structure or making assumptions about how a particular task or physical system works and instead learn this directly from the data, end-to-end methods tend to be more *general*; that is, they can be deployed to a range of tasks without much modification. For example, a system that includes a perception module that performs object pose detection may need to be modified to allow a robot to solve a task involving fluid manipulation, where it is unclear how to represent the pose of the fluid. In contrast, an end-to-end model that directly produces actions based on pixels or point clouds may require fewer changes. When trained on real-world robotics data, end-to-end approaches have shown strong deployment performance with minimal error due to domain gaps.

On the other hand, more modular approaches could be the key to improved generalization capabilities, better sample efficiency, and more easily inheriting improvements from computer vision and natural language processing. Structure in intermediate representations can provide powerful priors for improved generalization capabilities. For instance, an object-centric representation of a scene may provide positional or permutation invariance to a downstream planning module, while end-to-end methods must learn such rules from additional data. This has the potential to improve the sample efficiency of these models, which is critical while real robot data remains scarce. Intermediate structured representations such as language or feature fields offer an interface between these models that have been trained on large-scale *robot-free* data and modules specifically trained for robotic settings. Jointly optimizing all components based on downstream task performance could be a promising approach to overcoming compatibility issues when combining individually trained models.

6.8. High variability in real-world robotic settings

Another challenge is the high variability in robotic settings. Robot platforms are inherently diverse, with different physical characteristics, configurations, and capabilities. Real-world environments that robots operate in are also diverse and uncertain, with a wide range of variations. Due to all these variabilities, robotic solutions are usually tailored to specific robot platforms with specific layouts, environments, and objects for specific tasks. These solutions are not generalizable across various embodiments, environments, or tasks. Hence, to build general-purpose pretrained robotic foundation models, a key factor is to pretrain large models that are task-agnostic, cross-embodiment, and open-ended and capture diverse robotic data. In ROSIE (Yu et al., 2023), a diverse dataset is generated for robot learning by performing inpainting of various unseen objects, backgrounds, and distractors with semantic textual guidance. To overcome variability in robotic settings and improve generalization, another solution as ViNT (Shah et al., 2023b) presents is to train foundation models on diverse robotic data across various embodiments. RT-X (Padalkar et al., 2023) also investigates the possibility of training large cross-embodied robotic models in the domain of robotic manipulation. RT-X is trained using a multi-embodiment dataset called Open X-Embodiment dataset, which is created by collecting data from different robot platforms collected through a collaboration between 21 institutions, demonstrating 160266 tasks. RT-X demonstrates transfer across embodiment improves robot capabilities by employing experience from diverse robotic platforms.

6.9. Benchmarking and reproducibility in robotics settings

Another significant obstacle in incorporating foundation models into robotics research is the necessity of real-world hardware experiments. This creates challenges for reproducibility, as replicating results obtained from hardware experiments may require access to the exact equipment employed in prior work. Conversely, many recent works have relied on non-physics-based simulators (e.g., ignoring or greatly simplifying contact physics in gasping) that instead focus on high-level, long-term tasks and visual environment models. Examples of this class of simulators are common and include many of the simulators described above in Section 5. For example the Gibson family of simulators (Li et al., 2021; Xia et al., 2018), the Habitat family (Puig et al., 2023; Savva et al., 2019; Szot et al., 2021), RobotTHOR (Deitke et al., 2020), and VirtualHome (Puig et al., 2018) all neglect low-level physics in favor of simulating higher level tasks with high visual fidelity. This leads to a large sim-to-real gap and introduces variability in real-world performance based on how low-level planning and control modules handle the true physics of the scenario.

Even when physics-based simulators are used (e.g., Py-Bullet or MuJoCo), the absence of standardized simulation settings, computing environments, and a persistent sim-to-real gap impede efforts to benchmark and compare performance across various research endeavors.

A combination of open hardware, benchmarking in physics-based simulators, and promoting transparency in experimental and simulation setups can significantly alleviate the challenges associated with reproducibility in the integration of foundation models into robotics research. Generally, defining evaluation metrics in robotics research is challenging because each embodiment (robot) and research context is unique. This diversity makes it difficult to establish metrics that are universally applicable across different research groups. This challenge is not new and is not specific to foundation models; it is a well-recognized issue in benchmarking progress within the field of robotics.

With a standardized uniform hardware platform, it becomes easier to conceptualize competition-based evaluation metrics. For instance, task-based evaluation metrics could be established where the challenge is to complete a specific task with a given robot and dataset while allowing researchers to choose their own architectures and approaches. Such competitions hold the potential to provide objective measures of progress in robotics. They can serve as benchmarks to assess whether the field is advancing. By recording the outcomes of these competitions, we can gain valuable insights into the trajectory of innovation and development in robotics. These practices contribute to the development of a more robust and collaborative research ecosystem within the field.

6.10. Envisioning the impact of foundation models in robotics

In this section, we examine the potential answers to the following questions: What is the ultimate power of foundation models in robotics, and what could be achieved in the best-case scenario? The ultimate goal is to develop foundation models that enable robots to safely and efficiently perform a wide range of everyday tasks with a high success rate. These models could operate through simple interfaces, such as natural language text input, similar to how GPT-3 functions. As example applications, household robots could safely navigate indoor spaces, accurately grasp and manipulate objects, and perform various chores. Autonomous vehicles might achieve human-level contextual reasoning, especially in response to safety threats or anomalous situations. They would accurately predict the motion of other agents, facilitating safe real-time planning. Furthermore, robots with open-world navigation capabilities could be used for exploration purposes, such as space rovers or deep-sea explorers. Humanoid robots could be deployed in open-world environments to provide human-level navigation, manipulation, and dexterity. These robots could work alongside humans and other robots to accomplish diverse tasks in warehouses and industrial manufacturing.

7. Conclusion

Through examination of the recent literature, we have surveyed the diverse and promising applications of foundation models in robotics. We have delved into how these models have enhanced the capabilities of robots in areas such as decision-making, planning and control, and perception. We also discussed the literature on embodied AI and generalist AI, with an eye toward opportunities for roboticists to extend the concepts in that research field to real-world robotic applications. Generalization, zero-shot capabilities, multimodal capabilities, and scalability of foundation models have the potential to transform robotics. However, as we navigate through this paradigm shift in incorporating foundation models in robotics applications, it is imperative to recognize the challenges and potential risks that must be addressed in future research. Data scarcity in robotics applications, high variability in robotics settings, uncertainty quantification, safety evaluation, and real-time performance remain significant concerns that demand future research. We have delved into some of these challenges and have discussed potential avenues for improvement.

References

Anderson P, Wu Q, Teney D, et al. (2018) Vision-and-language navigation: interpreting visually-grounded navigation instructions in real environments. In: CVPR, pp. 3674–3683. IEEE.

Arandjelovic R, Gronat P, Torii A, et al. (2016) NetVLAD: CNN architecture for weakly supervised place recognition. In: CVPR, pp. 5297–5307. IEEE.

Bahl S, Mendonca R, Chen L, et al. (2023) Affordances from human videos as a versatile representation for robotics. In: CVPR. IEEE.

Baker B, Akkaya I, Zhokov P, et al. (2022) Video PreTraining (VPT): Learning to act by watching unlabeled online videos. In: NeurIPS. IEEE.

Baltrusaitis T, Ahuja C and Morency LP (2019) Multimodal machine learning: a survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(2): 423–443. DOI: 10.1109/TPAMI.2018.2798607.

Bangguo Y, Hamidreza K and Ming C (2023) LL3MVN: leveraging large language models for visual target navigation. *arXiv preprint arXiv:2304.05501*.

Bommasani R, Hudson DA, Adeli E, et al. (2021) On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Bonatti R, Vemprala S, Ma S, et al. (2023) PACT: perception-action causal transformer for autoregressive robotics pre-training. In: IROS. IEEE.

Bousmalis K, Vezzani G, Rao D, et al. (2023) RoboCat: a self-improving foundation agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*.

Box GE, Jenkins GM, Reinsel GC, et al. (2015) *Time Series Analysis: Forecasting and Control*. Hoboken, New Jersey: John Wiley & Sons.

Brohan A, Brown N, Carbajal J, et al. (2022) Rt-1: robotics transformer for real-world control at scale *arXiv Preprint arXiv:2212.06817*.

Brohan A, Brown N, Carbajal J, et al. (2023a) RT-1: robotics transformer for real-world control at scale. In: RSS.

Brohan A, Chebotar Y, Finn C, et al. (2023b) Do as I can, not as I say: grounding language in robotic affordances. In: CoRL, pp. 287–318. PMLR.

Brooks T, Peebles B, Holmes C, et al. (2024) Video generation models as world simulators. URL: https://Openai.Com/Research/Video-Generation-Models-As-World-Simulators

Brown T, Mann B, Ryder N, et al. (2020) Language models are few-shot learners. *NeurIPS* 33: 1877–1901.

Bucker A, Figueredo L, Haddadin S, et al. (2023) LATTE: LAnguage trajectory TransformEr. In: ICRA, pp. 7287–7294. IEEE.

Cai F and Koutsoukos X (2020) Real-time out-of-distribution detection in learning-enabled cyber-physical systems. In: ICCPS. IEEE.

Caron M, Touvron H, Misra I, et al. (2021) Emerging properties in self-supervised vision transformers. In: ICCV. IEEE.

Chang AX, Funkhouser T, Guibas L, et al. (2015) ShapeNet: an information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*.

Chang A, Dai A, Funkhouser T, et al. (2017) Matterport3D: learning from RGB-D data in indoor environments. In: 3DV, pp. 667–676. DOI: 10.1109/3DV.2017.00081.

Chen T, Kornblith S, Norouzi M, et al. (2020) A simple framework for contrastive learning of visual representations. In: ICML.

Chen M, Tworek J, Jun H, et al. (2021) Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Chen W, Hu S, Talak R, et al. (2022a) Leveraging large language models for robot 3D scene understanding. *arXiv preprint arXiv:2209.05629*.

Chen X, Wang X, Changpinyo S, et al. (2022b) PaLI: a jointly-scaled multilingual language-image model. In NeurIPS. IEEE.

Chen J, Li G, Kumar S, et al. (2023a) How to not train your dragon: training-free embodied object goal navigation with semantic frontiers. *arXiv preprint arXiv:2305.16925*.

Chen T, Tippur M, Wu S, et al. (2023b) Visual dexterity: in-hand reorientation of novel and complex object shapes. *Science Robotics* 8(84): eadc9244.

Chen X, Djolonga J, Padlewski P, et al. (2023c) PaLI-X: on scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*.

Chen Y, Arkin J, Zhang Y, et al. (2023d) AutoTAMP: autoregressive task and motion planning with llms as translators and checkers. *arXiv preprint arXiv:2306.06531*.

Chen Y, Gandhi R, Zhang Y, et al. (2023e) NL2TL: transforming natural languages to temporal logics using large language models. *arXiv preprint arXiv:2305.07766*.

Chen Z, Kiami S, Gupta A, et al. (2023f) GenAug: retargeting behaviors to unseen situations via generative augmentation. In: RSS.

Chen B, Xu Z, Kirmani S, et al. (2024) Spatialvlm: endowing vision-language models with spatial reasoning capabilities. In CVPR. IEEE.

Cheng HK and Schwing AG (2022) XMem: long-term video object segmentation with an atkinson-shiffrin memory model. In: ECCV, pp. 640–658. Springer.

Chowdhery A, Narang S, Devlin J, et al. (2022) PaLM: scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Clarke S (2019) *Robot Learning for Manipulation of Granular Materials Using Vision and Sound*. Master's Thesis. Carnegie Mellon University.

Dai Z, Yang Z, Yang Y, et al. (2019) Transformer-XL: attentive language models beyond a fixed-length context. In: ACL.

Damen D, Doughty H, Farinella GM, et al. (2018) Scaling ego-centric vision: the EPIC-KITCHENS dataset. In: ECCV. Springer.

Dasari S, Ebert F, Tian S, et al. (2019) RoboNet: large-scale multi-robot learning. In: CoRL.

Dasgupta I, Kaeser-Chen C, Marino K, et al. (2022) Collaborating with language models for embodied reasoning. In: Second Workshop on Language and Reinforcement Learning.

Dehghani M, Djolonga J, Mustafa B, et al. (2023) Scaling vision transformers to 22 billion parameters. In: ICML.

Deitke M, Han W, Herrasti A, et al. (2020) RoboTHOR: an open simulation-to-real embodied AI platform. In: CVPR. IEEE.

DeTone D, Malisiewicz T and Rabinovich A (2018) Superpoint: self-supervised interest point detection and description. In: CVPR deep learning for visual SLAM workshop. IEEE.

Devlin J, Chang MW, Lee K, et al. (2018) BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding W, Xu C, Arief M, et al. (2023) A survey on safety-critical driving scenario generation—a methodological perspective. *IEEE ITSC* 99: 1–19.

Dong Q, Li L, Dai D, et al. (2022) A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Dosovitskiy A, Beyer L, Kolesnikov A, et al. (2021) An image is worth 16x16 words: transformers for image recognition at scale. In: ICLR.

Driess D, Xia F, Sajjadi MSM, et al. (2023) PaLM-E: an embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Du Y, Li J, Tang T, et al. (2023a) Zero-shot visual question answering with language model feedback. *arXiv preprint arXiv:2305.17006*.

Du Y, Watkins O, Wang Z, et al. (2023b) Guiding pretraining in reinforcement learning with large language models. In: ICML, pp. 8657–8677. PMLR.

Du Y, Yang M, Dai B, et al. (2023c) Learning universal policies via text-guided video generation. In: NeurIPS.

Du Y, Yang M, Florence P, et al. (2023d) Video language planning. *arXiv preprint arXiv:2310.10625*.

Duan Y, Schulman J, Chen X, et al. (2016) RL'2: fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.

Dugas D (2023) The gpt-3 architecture, on a napkin. URL: https://dugas.ch/artificial_curiosity/GPT_architecture.html ([Online; accessed 28-November-2023]).

Ehsani K, Gupta T, Hendrix R, et al. (2024) Spoc: imitating shortest paths in simulation enables effective navigation and manipulation in the real world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16238–16250. IEEE.

Engelbrecht HA and Schiele G (2014) Transforming minecraft into a research platform. In: IEEE CCNC. IEEE.

Fan L, Wang G, Jiang Y, et al. (2022) MineDojo: building open-ended embodied agents with internet-scale knowledge. In: NeurIPS Datasets and Benchmarks Track.

Fang HS, Wang C, Gou M, et al. (2020) Graspnet-1billion: a large-scale benchmark for general object grasping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11444–11453. IEEE.

Farid A, Veer S and Majumdar A (2021) Task-driven out-of-distribution detection with statistical guarantees for robot learning. In: CoRL.

Farid A, Snyder D, Ren AZ, et al. (2022a) Failure prediction with statistical guarantees for vision-based robot control. *arXiv preprint arXiv:2202.05894*.

Farid A, Veer S, Ivanovic B, et al. (2022b) Task-relevant failure detection for trajectory predictors in autonomous vehicles. In: CoRL.

Feng X, Luo Y, Wang Z, et al. (2023) ChessGPT: bridging policy learning and language modeling. *arXiv preprint arXiv:2306.09200*.

Fu L, Datta G, Huang H, et al. (2024) A touch, vision, and language dataset for multimodal alignment. *arXiv preprint arXiv:2402.13232*.

Gadre SY, Wortsman M, Ilharco G, et al. (2023) CoWs on pasture: baselines and benchmarks for language-driven zero-shot object navigation. In: CVPR, pp. 23171–23181. IEEE.

Ganguli D, Lovitt L, Kernion J, et al. (2022) Red teaming language models to reduce harms: methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

Gao R, Si Z, Chang YY, et al. (2022) Objectfolder 2.0: a multi-sensory object dataset for sim2real transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10598–10608. IEEE.

Gao J, Sarkar B, Xia F, et al. (2023) Physically grounded vision-language models for robotic manipulation. In: CoRL.

Garrett CR, Chitnis R, Holladay R, et al. (2021) Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems* 4: 265–293.

Gervet T, Chintala S, Batra D, et al. (2023) Navigating to objects in the real world. *arXiv preprint arXiv:2212.00922*.

Goodwin W, Havoutis I and Posner I (2022a) You only look at one: category-level object representations for pose estimation from a single example. In: CoRL.

Goodwin W, Vaze S, Havoutis I, et al. (2022b) Zero-shot category-level object pose estimation. In: ECCV.

Greenberg I and Mannor S (2021) Detecting rewards deterioration in episodic reinforcement learning. In: ICML.

Guzhov A, Raue F, Hees J, et al. (2022) AudioCLIP: extending clip to image, text and audio. In: ICASSP, pp. 976–980. IEEE.

Ha H and Song S (2022) Semantic abstraction: open-world 3D scene understanding from 2D vision-language models. In: CoRL.

Han K, Wang Y, Chen H, et al. (2022) A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 99, 1.

He K, Zhang X, Ren S, et al. (2015) Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

He K, Gkioxari G, Dollár P, et al. (2017) Mask R-CNN. In: ICCV, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

He K, Chen X, Xie S, et al. (2022) Masked autoencoders are scalable vision learners. In: CVPR, pp. 16000–16009.

He X, Lin Z, Gong Y, et al. (2023) AnnoLLM: making large language models to be better crowdsourced annotators. *arXiv preprint arXiv:2303.16854*.

Ho J, Jain A and Abbeel P (2020) Denoising diffusion probabilistic models. In: NeurIPS.

Hong Y, Zhen H, Chen P, et al. (2023) 3D-LLM: injecting the 3D world into large language models. *arXiv preprint arXiv:2307.12981*.

Hsu KC, Hu H and Fisac JF (2023a) The safety filter: a unified view of safety-critical control in autonomous systems. *arXiv preprint arXiv:2309.05837*.

Hsu KC, Ren AZ, Nguyen DP, et al. (2023b) Sim-to-Lab-to-Real: safe reinforcement learning with shielding and generalization guarantees. *Artificial Intelligence* 314: 103811.

Hu A, Russell L, Yeo H, et al. (2023a) GAIA-1: a generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*.

Hu Y, Xie Q, Jain V, et al. (2023b) Toward general-purpose robots via foundation models: a survey and meta-analysis. *arXiv preprint arXiv:2312.08782*.

Hu X, Wang Y, Fan L, et al. (2024) Semantic anything in 3d Gaussians. *arXiv preprint arXiv:2401.17857*.

Huang J, Xie S, Sun J, et al. (2021) Learning a decision module by imitating driver's control behaviors. In: CoRL, pp. 1–10. PMLR.

Huang W, Abbeel P, Pathak D, et al. (2022a) Language models as zero-shot planners: extracting actionable knowledge for embodied agents. In: ICML.

Huang W, Xia F, Xiao T, et al. (2022b) Inner monologue: embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*.

Huang C, Mees O, Zeng A, et al. (2023a) Audio visual language maps for robot navigation. *arXiv preprint arXiv:2303.07522*.

Huang C, Mees O, Zeng A, et al. (2023b) Visual language maps for robot navigation. In: 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 10608–10615. IEEE.

Huang W, Wang C, Zhang R, et al. (2023c) VoxPoser: composable 3D value maps for robotic manipulation with language models. In: CoRL.

Janner M, Du Y, Tenenbaum J, et al. (2022) Planning with diffusion for flexible behavior synthesis. In: ICML.

Jia Z, Liu F, Thumuluri V, et al. (2023) Chain-of-thought predictive control. *arXiv preprint arXiv:2304.00776*.

Jiang Y, Gupta A, Zhang Z, et al. (2023) VIMA: general robot manipulation with multimodal prompts. In: ICML.

Kapelyukh I, Vosylius V and Johns E (2023) DALL-E-Bot: introducing web-scale diffusion models to robotics. *IEEE Robotics and Automation Letters* 8(7): 3956–3963. DOI: 10. 1109/LRA.2023.3272516.

Karamcheti S, Nair S, Chen AS, et al. (2023) Language-driven representation learning for robotics. In: RSS.

Kerbl B, Kopanas G, Leimkuehler T, et al. (2023) 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* 42(4): 1–14.

Kerr J, Kim CM, Goldberg K, et al. (2023) LERF: language embedded radiance fields. In: ICCV, pp. 19729–19739.

Khan S, Naseer M, Hayat M, et al. (2022) Transformers in vision: a survey. *ACM Computing Surveys* 54(10s): 1–41.

Kirillov A, Mintun E, Ravi N, et al. (2023) Segment anything. In: ICCV, pp. 4015–4026.

Kobayashi S, Matsumoto E and Sitzmann V (2022) Decomposing NeRF for editing via feature field distillation. In: NeurIPS.

Kumar A, Fu Z, Pathak D, et al. (2021) Rma: rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*.

Kusano KD, Beatty K, Schnelle S, et al. (2022) Collision avoidance testing of the waymo automated driving system. *arXiv preprint arXiv:2212.08148*.

Kwon M, Xie SM, Bullard K, et al. (2023) Reward design with language models. In: ICLR.

Levesque H, Davis E and Morgenstern L (2012) The Winograd schema challenge. In: KR.

Li C, Xia F, Martín-Martín R, et al. (2021) iGibson 2.0: object-centric simulation for robot learning of everyday household tasks. In: CoRL.

Li B, Weinberger KQ, Belongie S, et al. (2022a) Language-driven semantic segmentation. In: ICLR.

Li C, Zhang R, Wong J, et al. (2022b) BEHAVIOR-1K: a benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In: CoRL.

Li J, Li D, Xiong C, et al. (2022c) BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML.

Li LH, Zhang P, Zhang H, et al. (2022d) Grounded language-image pre-training. In: CVPR,pp. 10965–10975.

Li Y, Fan H, Hu R, et al. (2023) Scaling language-image pre-training via masking. In: CVPR.

Liang J, Huang W, Xia F, et al. (2023) Code as Policies: language model programs for embodied control. In: ICRA, pp. 9493–9500. IEEE.

Liao G, Li J, Bao Z, et al. (2024) Clip-gs: clip-informed gaussian splatting for real-time and view-consistent 3d semantic understanding. *arXiv preprint arXiv:2404.14249*.

Lin J, Du Y, Watkins O, et al. (2023a) Learning to model the world with language. *arXiv preprint arXiv:2308.01399*.

Lin J, Tang J, Tang H, et al. (2023b) AWQ: activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*.

Lin K, Agia C, Migimatsu T, et al. (2023c) Text2Motion: from natural language instructions to feasible plans. In: Special Issue: Large Language Models in Robotics. Autonomous Robots.

Liu PJ, Saleh M, Pot E, et al. (2018) Generating Wikipedia by summarizing long sequences. In: ICLR.

Liu Y, Ott M, Goyal N, et al. (2019) RoBERTa: a robustly optimized BERT pretraining approach. *arXiv preprint arXiv: 1907.11692*.

Liu M, Zhu Y, Cai H, et al. (2023a) Partslip: low-shot part segmentation for 3d point clouds via pretrained image-language models. *CVPR*: 21736–21746.

Liu P, Yuan W, Fu J, et al. (2023b) Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. *ACM Computing Surveys* 55(9): 1–35.

Liu S, Zeng Z, Ren T, et al. (2023c) Grounding DINO: marrying DINO with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*.

Liu W, Du Y, Hermans T, et al. (2023d) StructDiffusion: language-guided creation of physically-valid structures using unseen objects. In: RSS.

Luo R, Zhao S, Kuck J, et al. (2022) Sample-efficient safety assurances using conformal prediction. In: WAFR.

Lynch C and Sermanet P (2021) Language conditioned imitation learning over unstructured data. In: Robotics: Science and Systems.

Lynch C, Khansari M, Xiao T, et al. (2020) Learning latent plans from play. In: CoRL, pp. 1113–1132. PMLR.

Ma S, Vemprala S, Wang W, et al. (2022) COMPASS: contrastive multimodal pretraining for autonomous systems. In: IROS, pp. 1000–1007. IEEE.

Ma YJ, Kumar V, Zhang A, et al. (2023a) LIV: language-image representations and rewards for robotic control. In: ICML.

Ma YJ, Sodhani S, Jayaraman D, et al. (2023b) VIP: towards universal visual reward and representation via value-implicit pre-training. In:ICLR.

Mahmoudieh P, Pathak D and Darrell T (2022) Zero-shot reward specification via grounded natural language. In ICML, pp. 14743–14752. PMLR.

Majumdar A, Shrivastava A, Lee S, et al. (2020) Improving vision-and-language navigation with image-text pairs from the web. In: ECCV, pp. 259–274. Springer.

Mandi Z, Bharadhwaj H, Moens V, et al. (2022) CACTI: a framework for scalable multi-task multi-scene visual imitation learning. *arXiv preprint arXiv:2212.05711*.

Margolis GB and Agrawal P (2023) Walk these ways: tuning robot control for generalization with multiplicity of behavior. In: Conference on Robot Learning, pp. 22–31. PMLR.

Mees O, Borja-Diaz J and Burgard W (2023) Grounding language with visual affordances over unstructured data. In: ICRA, pp. 11576–11582. IEEE.

Mildenhall B, Srinivasan PP, Tancik M, et al. (2021) Nerf: representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* 65(1): 99–106.

Minderer M, Gritsenko A, Stone A, et al. (2022) Simple open-vocabulary object detection with vision transformers. In: ECCV, pp. 728–755. Springer.

Mirchandani S, Xia F, Florence P, et al. (2023) Large language models as general pattern machines. *arXiv preprint arXiv: 2307.04721*.

Mu Y, Zhang Q, Hu M, et al. (2023) EmbodiedGPT: vision-language pre-training via embodied chain of thought. *arXiv preprint arXiv:2305.15021*.

Nair S, Mitchell E, Chen K, et al. (2022a) Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In: CoRL, pp. 1303–1315. PMLR.

Nair S, Rajeswaran A, Kumar V, et al. (2022b) R3M: a universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*.

Oord A, Li Y and Vinyals O (2018) Representation learning with contrastive predictive coding. *arXiv preprint arXiv: 1807.03748*.

OpenAI (2023) GPT-4 technical report. *arXiv preprint arXiv: 2303.08774*.

Oquab M, Darcet T, Moutakanni T, et al. (2023) DINOv2: learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Padalkar A, Pooley A, Jain A, et al. (2023) Open X-Embodiment: robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*.

Palo ND, Byravan A, Hasenclever L, et al. (2023) Towards a unified agent with foundation models. In: Workshop on Reincarnating Reinforcement Learning at ICLR 2023.

Park JS, O'Brien JC, Cai CJ, et al. (2023a) Generative Agents: interactive simulacra of human behavior. In: ACM Symposium on User Interface Software and Technology. ACM.

Park YJ, Wang H, Ardeshir S, et al. (2023b) Representation reliability and its impact on downstream tasks. *arXiv preprint arXiv:2306.00206*.

Perez E, Huang S, Song F, et al. (2022) Red teaming language models with language models. In: EMNLP.

Puig X, Ra K, Boben M, et al. (2018) VirtualHome: simulating household activities via programs. In: CVPR, pp. 8494–8502.

Puig X, Undersander E, Szot A, et al. (2023) Habitat 3.0: a co-habitat for humans, avatars and robots. *arXiv preprint arXiv: 2310.13724*.

Qin M, Li W, Zhou J, et al. (2023) Langsplat: 3d language Gaussian splatting. *arXiv preprint arXiv:2312.16084*.

Qiu J, Li L, Sun J, et al. (2023) Large ai models in health informatics: applications, challenges, and the future. *IEEE Journal of Biomedical and Health Informatics* 27(12): 6074–6087. DOI: 10.1109/JBHI.2023.3316750.

Radford A, Narasimhan K, Salimans T et al. (2018) Improving language understanding by generative pre-training. https://openai.com/research/language-unsupervised

Radford A, Wu J, Child R, et al. (2019) *Language Models Are Unsupervised Multitask Learners*. OpenAI Blog.

Radford A, Kim JW, Hallacy C, et al. (2021a) Learning transferable visual models from natural language supervision. In: M Meila and T Zhang (eds) ICML, Proceedings of Machine Learning Research, Vol. 139, 8748–8763. PMLR.

Radford A, Kim JW, Hallacy C, et al. (2021b) Learning transferable visual models from natural language supervision. In: ICML, pp. 8748–8763.

Radosavovic I, Xiao T, James S, et al. (2023) Real-world robot learning with masked visual pre-training. In: CoRL, pp. 416–426. PMLR.

Raffel C, Shazeer N, Roberts A, et al. (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21(1): 5485–5551.

Ramesh A, Pavlov M, Goh G, et al. (2021a) Zero-shot text-to-image generation. In: ICML, pp. 8821–8831. PMLR.

Ramesh A, Pavlov M, Goh G, et al. (2021b) Zero-shot text-to-image generation. In: M Meila and T Zhang (eds) Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research, Vol. 139, pp. 8821–8831. PMLR.

Ramesh A, Dhariwal P, Nichol A, et al. (2022) Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*.

Ranftl R, Bochkovskiy A and Koltun V (2021) Vision transformers for dense prediction. In: ICCV, pp. 12179–12188. PMLR.

Rashid A, Sharma S, Kim CM, et al. (2023) Language embedded radiance fields for zero-shot task-oriented grasping. In: 7th Annual Conference on Robot Learning (CoRL), pp. 178–200. PMLR.

Reed S, Zolna K, Parisotto E, et al. (2022) A generalist agent. *arXiv preprint arXiv:2205.06175*.

Reid M, Yamada Y and Gu SS (2022) Can Wikipedia help offline reinforcement learning? *arXiv preprint arXiv:2201.12122*.

Ren AZ, Dixit A, Bodrova A, et al. (2023) Robots that ask for help: uncertainty alignment for large language model planners. In: CoRL.

Rombach R, Blattmann A, Lorenz D, et al. (2022) High-resolution image synthesis with latent diffusion models. In: CVPR.

Savva M, Kadian A, Maksymets O, et al. (2019) Habitat: a platform for embodied AI research. In: ICCV.

Sennrich R, Haddow B and Birch A (2016) Neural machine translation of rare words with subword units. In: ACL.

Sermanet P, Ding T, Zhao J, et al. (2023) Robovqa: multimodal long-horizon reasoning for robotics. *arXiv:2311.00899*.

Shafiullah NMM, Paxton C, Pinto L, et al. (2023) CLIP-fields: weakly supervised semantic fields for robotic memory. In: RSS.

Shah R and Kumar V (2021) RRL: resnet as representation for reinforcement learning. In: ICML.

Shah S, Dey D, Lovett C, et al. (2017) AirSim: high-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics.

Shah D, Osiński B, Levine S, et al. (2023a) LM-Nav: robotic navigation with large pre-trained models of language, vision, and action. In: CoRL, pp. 492–504. PMLR.

Shah D, Sridhar A, Dashora N, et al. (2023b) ViNT: a foundation model for visual navigation. In: CoRL.

Shah R, Martín-Martín R and Zhu Y (2023c) MUTEX: learning unified policies from multimodal task specifications. In: CoRL.

Shen Q, Yang X and Wang X (2023a) Anything-3D: towards single-view anything reconstruction in the wild. *arXiv preprint arXiv:2304.10261*.

Shen W, Yang G, Yu A, et al. (2023b) Distilled feature fields enable few-shot language-guided manipulation. In: 7th Annual Conference on Robot Learning (CoRL).

Shen W, Yang G, Yu A, et al. (2023c) Distilled feature fields enable few-shot manipulation. In: CoRL.

Shorinwa O, Tucker J, Smith A, et al. (2024) Splat-mover: multi-stage, open-vocabulary robotic manipulation via editable Gaussian splatting.

Shridhar M, Manuelli L and Fox D (2022) CLIPort: what and where pathways for robotic manipulation. In: CoRL, pp. 894–906. PMLR.

Shridhar M, Manuelli L and Fox D (2023) Perceiver-Actor: a multi-task transformer for robotic manipulation. In: CoRL, pp. 785–799. PMLR.

Simeonov A, Du Y, Tagliasacchi A, et al. (2022) Neural Descriptor Fields: SE(3)-equivariant object representations for manipulation. In: ICRA.

Singh I, Blukis V, Mousavian A, et al. (2023) ProgPrompt: generating situated robot task plans using large language models. In: ICRA, pp. 11523–11530. IEEE.

Sinha R, Sharma A, Banerjee S, et al. (2022) A system-level view on out-of-distribution data in robotics. *arXiv preprint arXiv: 2212.14020*.

Sohl-Dickstein J, Weiss E, Maheswaranathan N, et al. (2015) Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML.

Sohn K (2016) Improved deep metric learning with multi-class n-pair loss objective. In: NeurIPS.

Song Y and Ermon S (2019) Generative modeling by estimating gradients of the data distribution. In: NeurIPS.

Stone A, Xiao T, Lu Y, et al. (2023) Open-world object manipulation using pre-trained vision-language model. *arXiv preprint arXiv:2303.00905*.

Sun J, Sun H, Han T, et al. (2021a) Neuro-symbolic program search for autonomous driving decision module design. In: CoRL, pp. 21–30. PMLR.

Sun J, Yu L, Dong P, et al. (2021b) Adversarial inverse reinforcement learning with self-attention dynamics model. *IEEE Robotics and Automation Letters* 6(2): 1880–1886.

Sun J, Huang DA, Lu B, et al. (2022a) PlaTe: visually-grounded planning with transformers in procedural tasks. *IEEE Robotics and Automation Letters* 7(2): 4924–4930.

Sun J, Kousik S, Fridovich-Keil D, et al. (2022b) Self-supervised traffic advisors: distributed, multi-view traffic prediction for smart cities. In: IEEE ITSC.

Sun M, Yan W, Abbeel P, et al. (2022c) Quantifying uncertainty in foundation models via ensembles. In: NeurIPS Workshop on Robustness in Sequence Modeling.

Sun J, Jiang Y, Qiu J, et al. (2023a) Conformal prediction for uncertainty-aware planning with diffusion dynamics model. In: NeurIPS.

Sun J, Kousik S, Fridovich-Keil D, et al. (2023b) Connected autonomous vehicle motion planning with video predictions from smart, self-supervised infrastructure. *arXiv preprint arXiv:2309.07504*.

Sun J, Qiu J, Zheng C, et al. (2023c) Aria-NeRF: multimodal egocentric view synthesis. *arXiv preprint arXiv:2311.06455*.

Sun J, Xu Y, Ding M, et al. (2023d) NeRF-Loc: transformer-based object localization within neural radiance fields. *IEEE Robotics and Automation Letters* 8(8): 5244–5250. DOI: 10. 1109/LRA.2023.3293308.

Sun J, Zheng C, Xie E, et al. (2023e) Reasoning with foundation models: concepts, methodologies, and outlook. *Zenodo Preprint*. DOI: 10.5281/zenodo.10298866.

Sun Y, Ma S, Madaan R, et al. (2023f) SMART: self-supervised multi-task pretraining with control transformers. In: ICLR.

Szot A, Clegg A, Undersander E, et al. (2021) Habitat 2.0: training home assistants to rearrange their habitat. In: NeurIPS.

Tarasov D, Kurenkov V and Kolesnikov S (2022) Prompts and pre-trained language models for offline reinforcement learning. In: ICLR Workshop on Generalizable Policy Learning in Physical World.

Team OEL, Stooke A, Mahajan A, et al. (2021) Open-ended learning leads to generally capable agents. *arXiv preprint arXiv:2107.12808*.

Team AA, Bauer J, Baumli K, et al. (2023) Human-timescale adaptation in an open-ended task space. In: ICML.

Thickstun J (2023) The transformer model in equations. URL: https://johnthickstun.com/docs/transformers.pdf. [Online; accessed 28-November-2023].

Tong S, Jones E and Steinhardt J (2023) Mass-producing failures of multimodal systems with language models. *arXiv preprint arXiv:2306.12105*.

Touvron H, Lavril T, Izacard G, et al. (2023a) LLaMA: open and efficient foundation language models. *arXiv preprint arXiv: 2302.13971*.

Touvron H, Lavril T, Izacard G, et al. (2023b) LLaMA: open and efficient foundation language models. *arXiv preprint arXiv: 2302.13971*.

Tschernezki V, Laina I, Larlus D, et al. (2022) Neural Feature Fusion Fields: 3D distillation of self-supervised 2D image representations. In: 3DV, pp. 443–453. IEEE.

Umeyama S (1991) Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(04): 376–380.

Vaswani A, Shazeer N, Parmar N, et al. (2017) Attention is all you need. In: NeurIPS.

Vemprala S, Bonatti R, Bucker A, et al. (2023) ChatGPT for robotics: design principles and model abilities. Technical Report MSR-TR-2023-8, Microsoft.

Villegas R, Babaeizadeh M, Kindermans PJ, et al. (2023) Phenaki: variable length video generation from open domain textual description. In: ICLR.

Vincent JA, Feldman AO and Schwager M (2023) Guarantees on robot system performance using stochastic simulation rollouts. *arXiv preprint arXiv:2309.10874*.

Vincent JA, Nishimura H, Itkina M, et al. (2024) How generalizable is my behavior cloning policy? a statistical approach to trustworthy performance evaluation. *arXiv preprint arXiv: 2405.05439*.

Wang A, Singh A, Michael J, et al. (2018) GLUE: a multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Wang W, Zhu D, Wang X, et al. (2020) TartanAir: a dataset to push the limits of visual SLAM. In: IROS.

Wang C, Chai M, He M, et al. (2022) CLIP-NeRF: text-and-image driven manipulation of neural radiance fields. In: CVPR, pp. 3835–3844.

Wang C, Fan L, Sun J, et al. (2023a) MimicPlay: long-horizon imitation learning by watching human play. In: CoRL.

Wang G, Xie Y, Jiang Y, et al. (2023b) Voyager: an open-ended embodied agent with large language models. *arXiv preprint arXiv: Arxiv-2305.16291*.

Wang S, Saharia C, Montgomery C, et al. (2023c) Imagen editor and editbench: advancing and evaluating text-guided image inpainting. In: CVPR, pp. 18359–18369.

Webb N, Smith D, Ludwick C, et al. (2020) Waymo's safety methodologies and safety readiness determinations. *arXiv preprint arXiv:2011.00054*.

Wei J, Wang X, Schuurmans D, et al. (2022) Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS* 35: 24824–24837.

Wen M, Lin R, Wang H, et al. (2023) Large sequence models for sequential decision-making: a survey. *Frontiers of Computer Science* 17(6): 176349.

Wikipedia (2023) GPT-3. URL: https://en.wikipedia.org/wiki/GPT-3. [Online; accessed 28-November-2023].

Wu Z, Xiong Y, Yu SX, et al. (2018) Unsupervised feature learning via non-parametric instance discrimination. In: CVPR.

Xia F, R Zamir A, He ZY, et al. (2018) Gibson Env: real-world perception for embodied agents. In: CVPR.

Xian Z, Gervet T, Xu Z, et al. (2023) Towards generalist robots: a promising paradigm via generative simulation. *arXiv preprint arXiv:2305.10455*.

Xiao T, Radosavovic I, Darrell T, et al. (2022) Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*.

Xiao T, Chan H, Sermanet P, et al. (2023a) Robotic skill acquisition via instruction augmentation with vision-language models. In: RSS.

Xiao X, Liu J, Wang Z, et al. (2023b) Robot learning in the era of foundation models: a survey. *arXiv preprint arXiv:2311.14379*.

Xue L, Gao M, Xing C, et al. (2022) ULIP: learning unified representation of language, image and point cloud for 3D understanding. *arXiv preprint arXiv:2212.05171*.

Xue L, Yu N, Zhang S, et al. (2023) ULIP-2: towards scalable multimodal pre-training for 3d understanding. *arXiv preprint arXiv:2305.08275*.

Yang J, Gao M, Li Z, et al. (2023a) Track anything: segment anything meets videos. *arXiv preprint arXiv:2304.11968*.

Yang S, Nachum O, Du Y, et al. (2023b) Foundation models for decision making: problems, methods, and opportunities. *arXiv preprint arXiv:2303.04129*.

Yang F, Feng C, Chen Z, et al. (2024) Binding touch to everything: learning unified multimodal tactile representations. In: Computer Vision and Pattern Recognition (CVPR). Springer.

Yao L, Huang R, Hou L, et al. (2022) FILIP: fine-grained interactive language-image pre-training. In: ICLR.

Yao S, Zhao J, Yu D, et al. (2023) ReAct: synergizing reasoning and acting in language models. In: ICLR.

Ye J, Wang N and Wang X (2023a) FeatureNeRF: learning generalizable nerfs by distilling pre-trained vision foundation models. *arXiv preprint arXiv:2303.12786*.

Ye Y, Li X, Gupta A, et al. (2023b) Affordance diffusion: synthesizing hand-object interactions. In: CVPR.

Yenamandra S, Ramachandran A, Yadav K, et al. (2023) Homerobot: open vocabulary mobile manipulation. *arXiv preprint arXiv:2306.11565*.

Yi T, Mostafa D, Vinh QT, et al. (2023) UL2: unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*.

Yoneda T, Fang J, Li P, et al. (2023) Statler: state-maintaining language models for embodied reasoning. *arXiv preprint arXiv:2306.17840*.

Yu X, Tang L, Rao Y, et al. (2022) Point-BERT: pre-training 3D point cloud transformers with masked point modeling. In: CVPR, pp. 19313–19322. Springer.

Yu T, Xiao T, Stone A, et al. (2023) Scaling robot learning with semantically imagined experience. *arXiv Preprint arXiv: 2302.11550*.

Yuying G, Annabella M, Li EL, et al. (2023) Policy adaptation from foundation model feedback. In: CVPR.

Zeng A, Florence P, Tompson J, et al. (2020) Transporter networks: rearranging the visual world for robotic manipulation. In: CoRL.

Zeng A, Attarian M, Ichter B, et al. (2022) Socratic Models: composing zero-shot multimodal reasoning with language. *arXiv*.

Zeng A, Liu X, Du Z, et al. (2023a) GLM-130B: an open bilingual pre-trained model. In: ICLR.

Zeng Y, Jiang C, Mao J, et al. (2023b) CLIP2: contrastive language-image-point pretraining from real-world point cloud data. In: CVPR.

Zhai X, Kolesnikov A, Houlsby N, et al. (2022) Scaling vision transformers. In: CVPR.

Zhang R, Guo Z, Zhang W, et al. (2022a) PointCLIP: point cloud understanding by CLIP. In: CVPR, pp. 8552–8562.

Zhang Y, Jiang H, Miura Y, et al. (2022b) Contrastive learning of medical visual representations from paired images and text. In: MLHC.

Zhang Z, Wang Y, Zhang Y, et al. (2022c) Can offline reinforcement learning help natural language understanding? *arXiv preprint arXiv:2212.03864*.

Zhang C, Han D, Qiao Y, et al. (2023a) Faster segment anything: towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*.

Zhang D, Liang D, Yang H, et al. (2023b) SAM3D: zero-shot 3D object detection via segment anything model. *arXiv preprint arXiv:2306.02245*.

Zhang X, Kundu A, Funkhouser T, et al. (2023c) Nerflets: local radiance fields for efficient structure-aware 3d scene representation from 2d supervision. In: CVPR, pp. 8274–8284.

Zhao X, Ding W, An Y, et al. (2023) Fast segment anything. *arXiv preprint arXiv:2306.12156*.

Zhao Z, Chen S, Ding Y, et al. (2024) A survey of optimization-based task and motion planning: from classical to learning approaches. *arXiv preprint arXiv:2404.02817*.

Zhou C, Loy CC and Dai B (2022) Extract free dense labels from CLIP. In: European Conference on Computer Vision (ECCV), pp. 696–712. Springer.

Zhou S, Chang H, Jiang S, et al. (2023) Feature 3dgs: supercharging 3d Gaussian splatting to enable distilled feature fields. *arXiv preprint arXiv:2312.03203*.

Zhu X, Chen Y, Tian H, et al. (2023) Ghost in the Minecraft: generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*.

Zitkovich B, Yu T, Xu S, et al. (2023) RT-2: vision-language-action models transfer web knowledge to robotic control. In: CoRL.

Zuo X, Samangouei P, Zhou Y, et al. (2024) Fmgs: foundation model embedded 3d Gaussian splatting for holistic 3d scene understanding. *arXiv preprint arXiv: 2401.01970*.