Robotics: An Idiosyncratic Snapshot in the Age of LLMs

Anirudha Majumdar

August 2, 2023

1 Introduction

The goal of this document is to help me think through the state of the art in robotics today, along with the primary research challenges that persist. To this end, I will consider two main sets of questions:

- What are state-of-the-art approaches in different application domains? What are the primary challenges in each area?
- What are approaches that the field is currently excited about? What are the main opportunities and challenges?

Note: This document was primarily written for myself. It is not meant to be an academic survey; it is thoroughly incomplete and biased towards directions I was interested to learn more about. But, I am making it accessible in case others find it useful as a source of further sources!

2 State of the Art

2.1 Autonomous vehicles

Waymo. Waymo uses a modular sense-predict-plan-act pipeline, as outlined here [1, 2]. They fuse information from multiple sensors and a prior map in order to localize the ego-vehicle and estimate the state of objects and agents around the vehicle. The car makes predictions about other (relevant) agents in the scene. Both the perception module [3] and the behavior prediction module [4] heavily leverage transformers. These predictions are used to plan a safe and comfortable path; the planning happens in a receding horizon fashion. An important componenet of their overall framework is to leverage auto-labeling (e.g., labeling multiple timesteps in a run segment based on labels provided for a single timestep) and self-supervision (e.g., enforcing temporal consistency of predictions based on velocity estimation).

Testing is a major challenge with AVs. Waymo does a lot of its validation in simulation [2]. They collect data from real-world environments and populate their simulation environments using these. They use Nerfs for camera image synthesis in simulation (built from images that were collected from actual data). Simulating realistic (closed-loop) behaviors of other agents is a major challenge. Imitation learning plays a key role here [5].

More details on their collision avoidance testing procedure for scenarios that require urgent evasive maneuvers are provided in [6]. They utilize a scenario-based approach, where they manually generate various types of scenarios (e.g., unprotected left turns, vehicle cutting across ego vehicle, etc.) based on the operational design domain (ODD), recreate these on test tracks, augment these with scenarios from human crashes, and further augment these with scenarios collected in the field. They evaluate (in simulation) the performance of the AV in each type of scenario using collision and severity metrics, and also compare the performance with a model based on data from human drivers who are *Non-Impaired and Eyes ON the conflict* (NIEON). The goal is to demonstrate superior performance of the AV in each type of scenario.

Tesla. Tesla also uses a similar pipeline [7]. The perception module produces voxel-based occupancy representations, where each voxel has a corresponding occupancy probability and semantic information. The perception module (based purely on vision) uses a combination of convolutions and attention. They also have an interesting approach to lane detection; they treat the problem as a language modeling problem, where each token ("word") corresponds to a point where the lane branches. This approach provides benefits over simpler approaches given the complex connectivity (and resulting multi-modality) of lanes. Their planner uses tree search, where branches correspond to different maneuvers. The maneuvers are synthesized by sampling different intermediate goal locations and using a neural planner trained from human demonstrations and datasets of offline trajectory optimization results. They evaluate nodes in the tree by performing collision checking, comfort analysis, estimating probability of intervention (using a model trained from customer fleet data), and scoring based on a discriminator of human-like behavior. Like Waymo, they are also trying to leverage Nerfs for scene representation in simulation.

Zoox. Zoox has a similar algorithmic pipeline [8]. However, note that their hardware is custom-designed from the ground up. Thus, hardware-software co-design is an important part of their philosophy. This allows them to ameliorate some technical challenges in interesting ways (e.g., being able to follow waypoints with a much higher precision than would be feasible with a regular vehicle) [9].

Wayve. Wayve is taking a somewhat different approach; see [10] for an overview. Their goal is to use end-to-end learning techniques using only vision, no HD maps, and relatively few hand-crafted representations. They leverage significant amounts of offline data for imitation learning. They use hand-crafted representations as an "inductive bias" by performing end-to-end learning where latent representations can be used to decode hand-crafted features (e.g., locations of agents, semantic categories, etc.). They also leverage world models (i.e., predictive models of other agents) in their imitation learning framework [11] in a similar manner, i.e., ensuring that the latent state of the policy can be decoded to make predictions about the motion of other agents in the scene. This is an interesting approach overall, but time will tell whether this is a winning approach to autonomous driving; currently, Wayve has not deployed their systems at the same scale as some of the other players above.

2.2 Drones

Skydio. At a high level, the overall pipeline is similar to the one used by autonomous vehicles, with separate modules for state estimation, local mapping, and integrated planning/control; see [12] for an overview. State estimation is performed by formulating the visual-interial odometry (VIO) problem as nonlinear least-squares. The drone creates a *local* (i.e., not necessarily globally consistent) map of its environment using multiple fish-eye cameras (that provide 360° coverage). They use an end-to-end deep learning-based approach for stereo vision [13]. Planning and control are performed in an integrated manner by solving a nonlinear optimization problem with a large number of decision variables (including motor rates), and multiple constraints and objectives (including collision avoidance and objectives that encode camera-view considerations). This planning and control loop operates at 500Hz [12]. Similar to autonomous vehicles, simulation plays a critical role by allowing for testing in many different scenarios (which also incorporate data from real-world logs).

Learning high-speed flight in the wild. Arguably the most impressive vision-based drone navigation demos in academic research are presented in [14]. The policy is trained purely in simulation using many different forest-like scenes by learning to imitate an expert with privileged information. Specifically, the expert has access to low-level state information for the vehicle, along with obstacle location/geometry information; the expert performs Monte Carlo generation of (thousands of) collision-free trajectories that make progress towards a global target. These samples are generated from a multi-modal distribution (which is important, e.g., going left vs. right around a tree). The k-means algorithm is used to select three representative trajectories from the thousands that are sampled. The "student" policy learns to imitate this expert ("teacher") policy, but only has access to the raw sensory information from the drone (depth image, IMU measurements, etc.). More specifically, the student policy learns to output three trajectories; one of these is selected for execution based on a few factors (including trying to ensure continuity of the executed trajectory w.r.t. the previous trajectory). Finally, nonlinear model predictive control (NMPC) is used for low-level trajectory

tracking control of the selected trajectory.

2.3 Legged robots

Boston Dynamics. Boston Dynamics utilizes a heavily model-based approach for planning and control (see, e.g., [15, 16]). Some of their demos involve executing a certain behavior (e.g., Atlas performing a backflip or jumping onto a box), where the environment is tightly controlled (e.g., known height from/to which the robot is jumping). In these cases, the reference motion is generated offline via trajectory optimization. Their approach follows the commonly used strategy of decoupling the planning of (i) centroidal motion (by optimizing contact forces), and (ii) whole-body kinematic motion; see [17] and references therein for technical details on this kind of approach. Based on talks [15, 16], there are some important details regarding the specific manner in which the centroidal motion is planned in a way that takes into account changes in the moment of inertia resulting from the motion of arms and legs. The offline trajectory optimization problem is large and takes a while to solve. For real-time planning and control, they use the same overall decoupled planning of centroidal dynamics and full-body kinematics in a receding-horizon manner; this MPC problem is solved for very short time horizons (with warm-starting via a library of offline solutions). The real-time MPC-based planning scheme corrects for deviations from the nominal plan (e.g., reacting to a disturbance by replanning the desired location where a foot will land), or re-planning by taking into account new obstacles or potential contact locations. While their approach to planning and control leads to extremely impressive results, the integration of perception into the loop still seems relatively primitive. For the most part, they seem to rely on localizing objects with known geometry, or identifying instances of objects from a parameterized class (e.g., boxes with variable dimensions), or performing plane detection for finding potential contact locations.

Learning vision-based locomotion. Over the past 2-3 years, there have been a series of impressive demonstrations of vision-based quadruped locomotion in the academic research community [18, 19, 20, 21, 22, 23]. These papers almost all rely on a student-teacher training scheme implemented in simulation (similar to some of the work highlighted above with drones). A "teacher" policy with access to privileged information (e.g., terrain height and properties) is trained first; a "student" policy with access only to onboard sensor information is then trained to imitate the teacher policy. An important variation on this theme was introduced in [19], which proposes rapid motor adaptation (RMA). Here, a base policy trained using RL takes as input different pieces of privileged information (e.g., terrain height and friction properties); then, an adaptation module is trained to map sensor information available to the robot to estimates of the privileged information (or encoded versions of the privileged information). But, recent work [23] has shown that a "monolithic" student architecture (one that just learns to imitate the teacher without directly estimating intermediate quantities) can also perform well. Beyond the teacher-student policy training scheme, keys to all of these results are: (i) fast simulation (e.g., with Isaac Gym [24]), (ii) massive amounts of domain randomization in simulation, (iii) a training curriculum from easy-to-hard environments, and (iv) relatively careful choice of reward functions. A few other points to note: (i) these demonstrations are exclusively on quadrupeds (not bipeds), which makes things a bit easier; (ii) depth is used instead of RGB (RGB is more challenging, but could allow one to leverage additional semantic information, e.g., slipperiness of a surface); (iii) the work highlighted above focuses on locomotion rather than navigation (i.e., not focusing on generating high-level commands of which direction to move in; an exception is the work from the Cerberus team in the DARPA SubT challenge [25], where they used high-level planners for exploration and local traversability mapping); (iv) these demonstrations are in environments without humans.

2.4 Manipulators

Amazon. Amazon has some of the most advanced manipulation capabilities in deployment today. Several elements of their approach to manipulation are described in [26]. There are two kinds of manipulation scenarios that Amazon's warehouses require: (i) manipulating boxes (e.g., during the process of sortation), and (ii) manipulating individual items (e.g., during fulfillment). For manipulating boxes, Amazon uses suction-based grippers. There are three key elements to this: (i) perception, (ii) OOD detection, and (iii) planning.

The perception module is responsible for performing instance segmentation, and then regressing to a 6DoF pose estimate. As expected, they use a learning-based approach to perception with data augmentation and domain randomization with photorealistic simulators. The main challenge in perception is dealing with the open-world nature of packaging and distribution shifts that arise (due to packaging from third-party entities, e.g., diaper packages). OOD detection is thus a key component of Amazon's approach. OOD detection allows the robots to raise an alarm, which then triggers a re-training phase for the perception module. This involves humans annotating OOD data (with auto-labeling as well), and then re-training the perception module via importance sampling. A significant challenge with re-training is to prevent catastrophic forgetting (and, more subtly, ensuring that the new perception module still correctly handles packages that the old perception module did; if this isn't the case, the human operators have a hard time trusting the system). The planning module takes a 6DoF pose estimate from the perception module and plans forces for a subset of the suction cups in the gripper. The suction cups have relatively complex fluid mechanics one needs to account for; they use classical system identification techniques to obtain accurate models of the suction grippers. They use trajectory optimization techniques to plan a trajectory for the arm (and also plan a trajectory of suction forces) to ensure that the box does not fly off as the arm is moving it, and to place the box down gently.

In order to manipulate individual objects (e.g., in a fulfillment center), they use grippers in addition to suction (since suction can tear apart objects such as books). They use multi-view depth fusion, compute features on this fused signed-distance function-based representation, generate grasps (using a learned model), and then use trajectory optimization to move the arm to the desired grasp pose.

Covariant. The approach taken by Covariant [27] is also similar in spirit to Amazon. They take a modular approach, with learning playing an important role in each module. Their perception module is responsible for instance-level segmentation of objects. They have invested a lot of effort to train models that can handle very cluttered scenes. In addition, accurate depth estimation plays a major role in their manipulation pipeline. Again, they have found that state-of-the-art depth estimation techniques (even using LIDAR or structured light) did not meet their needs (e.g., for objects such as plastic water bottles with non-Lambertian reflectance properties); as a result, they have invested a lot of effort in learning-based approaches for accurate depth estimation. Self-supervised learning is also a key element of their approach. As an example, they use measurements of whether or not a suction cup has properly sealed in order to provide labels for grasping. Similarly, they monitor whether an object was dropped while moving it in order to provide a label for trajectory speed. As expected, photorealistic simulation plays a crucial role in all aspects of Covariant's approach.

2.5 Collaborative robots

Robots in factories have typically been isolated away from humans. This can take the form of hard fencing around the robot, or virtual fencing where sensors detect if a human gets close enough to the robot in order to trigger the robot to stop. More recently, collaborative robots (CoBots) are becoming more sophisticated (see [28] for a nice overview of the different levels of CoBot sophistication). Many start-up companies (e.g., Robust AI, Diligent, and Veo Robotics) and more traditional robot manufacturers (e.g., Kuka and Fanuc) are developing CoBots with the goal of seamless human-robot interaction in shared spaces. This includes social navigation (e.g., in hospital settings) and collaborative manipulation (e.g., for manufacturing).

I found it hard to find details on the technical approach that different companies are taking. There is a little bit of technical detail here [29, 30] on Diligent, which is deploying robots to help nurses in hospitals. It seems like they are using a fairly traditional pipeline where they first build a map of the environment and specify different landmark locations that the robot can then be instructed to navigate to. They also place QR/APRIL tags in different locations (e.g., elevators, automatic door buttons, shelves, etc.) to ease the burden on the perception system. For some of their manipulation tasks, they use learning from demonstrations (albeit with their staff heavily involved in this initial on-boarding phase). I wasn't able to find details on the hardest features of this application, which in my mind have to do with social navigation and scene understanding.

3 Research challenges

Based on the survey above, I highlight research challenges that cut across the different application domains, and also ones that are specific to the application domains.

3.1 Challenges that cut across application domains

System-level challenges:

- Generalization to a broad range of new scenarios and edge cases.
- Continuous improvement of performance via data collection and hard-instance generation (iterating in both simulation and hardware). Ensuring that updates do not hurt performance in unexpected ways.
- Choosing the right system architecture: a modular approach with separate modules for perception, planning, etc. can be easier to architect, debug, and update as compared to an end-to-end blackbox policy; but, one needs to be careful with the interfaces between the modules (e.g., preventing or addressing distribution shifts in sensory inputs when outputs of the perception system are used to choose actions, which affect future sensory inputs).
- Understanding error rates of different modules (e.g., perception, planning, etc.) and composing these together to obtain estimates of end-to-end error rates (e.g., collision rates and severe injury rates for autonomous vehicles).
- Addressing issues relating to fairness, privacy, and economic impact during development (instead of after-the-fact band-aids).
- Effective regulation, certification, and communication with the public on the opportunities and risks of the systems we create.

Perception:

- Extremely reliable prediction of occupancy and semantics, in many edge cases. How can we continuously improve performance as we collect more data?
- Representing uncertainty in a well-calibrated manner. For interpretability, these estimates of uncertainty should be Frequentist in nature.
- Choosing good task-driven representations for perception (e.g., pixelated top down view of scene vs. a more compact representation of features and their locations). Deciding what is important to perceive and predict well (e.g., current and future locations of agents closer to us vs. farther away) for end-to-end performance.
- Scene understanding. A great example of limitations of the current paradigm for perception and prediction is shown here. A car is parked in an awkward position; the Tesla car infers that the car is simply stopped for traffic (and may soon move). However, the car actually has no human occupant! This is an example that calls out for high-level scene understanding: the Tesla car should understand that a car without a human driver is not going to start driving. Maybe it should infer that the car is stopped, look to see if someone is inside, and then reason that the car will not move. Can data alone solve these kinds of edge cases as Tesla and others are betting? (Note that this is *not* a rhetorical question; one would be foolish to bet against this given the success of deep learning.) Or do we need our perception systems to be grounded in causal models of the world? If the latter, how can we do that? Or can the high-level reasoning capabilities of large language models provide a solution to these kinds of challenges? High-level reasoning in perception (and other parts of the stack) could be one of the main use cases of LLMs in robotics (beyond the obvious use cases of human-robot interaction via language).

Planning and control:

- Better modeling and simulation of nonlinear dynamics, and especially contact. This is a challenge regardless of whether one is utilizing model-based or RL approaches (with training happening in simulation).
- Planning in a way that takes into account uncertainty in perception and predictions of other agents' motions.

Learning:

- Instance-level failure prediction and uncertainty quantification [31, 32, 33]: how can we predict whether our learned policy may fail in a particular setting? This can trigger a call for help from a human, or aborting the mission. This remains extremely challenging for settings with neural network-based policies processing rich sensory inputs, especially in OOD settings. This is also an important challenge in the context of language-instructed robots faced with potentially ambigious instructions [33].
- Few-shot detection of task-relevant distribution shifts [34, 35]. Detecting distribution shifts quickly can allow one to re-train on the new distribution.
- Re-training in OOD environments without catastrophic forgetting.
- Few-shot sim-to-real adaptation (e.g., [36, 37, 38]) in a task-driven manner [39].

Human-robot interaction:

- *Closed-loop* motion prediction (i.e., predictions that take into account the ego-agent's actions). Do we need causal models to do this extremely well (e.g., understand that two cars came to a stop because the car in front braked, which caused the car behind to brake as well)?
- Realistic behavior simulation: capturing the diversity (and realism) of behaviors.
- Multi-modal probabilistic predictions that cover different realistic possibilities.

Testing:

- At-scale testing in simulation by leveraging real-world data.
- Generating good test cases for validation. Exploring the space of "unknown-unknowns" is a major challenge.
- Building an overall safety case for regulators.

Scaling:

- Data pipeline (auto-labeling and self-supervision).
- Software infrastructure.
- Computation (training and inference).

Challenges that are more specific to different application domains are discussed below.

3.2 Autonomous vehicles

System-level challenges:

- Perhaps the biggest source of challenges in AVs is the extremely low levels of error that can be tolerated. This has an impact on every aspect of designing and testing an AV.

Human-robot interaction:

- The challenges of human-robot interaction are particularly exacerbated in AV applications. Good motion prediction, realistic simulation, and game-theoretic planning are all significant challenges (especially with all the edge cases that arise).

Planning:

- Planning in a way that takes into account the rules of the road and breaks them when necessary (e.g., veering into opposite lane when a car is double parked).

3.3 Drones

Dynamics:

- Complex aerodynamics, especially at high speeds.
- Uncertain wind disturbances.

Perception:

- GPS is denied in indoor environments.
- LIDAR is challenging to place on drones required to perform agile flight.
- Vision can be noisier, e.g., due to motion blur when flying very fast.
- Lack of high-fidelity wind sensing.

Scaling:

- Significantly more constrained processing and memory onboard.

3.4 Legged robots

Dynamics:

- Specifying accurate models of contact and friction (e.g., point contact/fricton vs. patch contact/friction); this is a challenge regardless of whether one is utilizing model-based or RL approaches (in simulation).
- System identification for the robot itself can be challenging if you have actuator limits and compliance in joints.

Planning and control:

- Model-based planning and control work quite well when the hybrid mode sequence is specified *a priori*; however, planning contact-mode sequences in real time is a major challenge (Boston Dynamics' approach assumes known hybrid mode sequence when planning). See [40] for some recent progress (albeit still with relatively simple systems).
- For RL-based approaches, using vision to perform high-level planning (in addition to locomotion control) is still relatively challenging; this is a place where using more traditional planners in combination with RL-based controllers for locomotion probably makes the most sense.

Perception and state estimation:

- Many of the same challenges with respect to robust perception carry over from AVs and drones to legged robots. However, there are even more pronounced challenges with legged robots since the perception pipeline is not just responsible for estimating obstacle locations/geometry, but also other physical properties and affordances of the environment (e.g., friction properites of a surface, or the likelihood of a surface supporting the weight of the robot if it jumps onto it).
- Using RGB observations robustly (e.g., for estimating slipperiness of terrain) is still a challenge (due to the larger sim-to-real gap as compared to depth sensors).
- Good contact detection is essential for model-based approaches to planning and control.

Human-robot interaction:

- In settings that require operating around humans, the same challenges with AVs apply here. These challenges are exacerbated if the humans are expected to make contact with the robot.

3.5 Manipulators

System-level challenges:

- The open-world nature of manipulation poses a major challenge in terms of generalization.
- While a lot of academic research has historically focused on grasping (and significant progress has been made on this problem), the broader aspects of manipulation remain significant challenges.

Sensing:

- Tactile sensing is an active area of research, and powerful tactile sensors are becoming more broadly available [41, 42]. Finding good ways to utilize tactile sensors for perception and control (especially real-time feedback control) in order to complement RGB-D sensors is an important challenge.
- Depth sensing with specular objects (e.g., glass or plastic) is challenging. Traditional depth sensors provide noisy measurements in such cases. There is some recent work that uses Nerfs to obtain depth maps with such objects [43].

Perception:

- What is the right representation to connect perception and action in manipulation: traditional object-centric representations (representing shape, pose, etc. such as the approaches highlighted above), features computed directly from images (e.g., using self-supervised learning [44, 45]), dense descriptors that assign features to every point/pixel/voxel (e.g., [46, 47, 48]), key-point-based representations (e.g., [49, 50, 51, 52]), Nerfs or related representations like deep signed distance functions (e.g., [43, 53, 54, 55])?
- Coupling geometric information about the scene with semantic information in order to perform open-ended tasks specified via language. There is some recent work on this [56], but generalization to unseen semantic categories (e.g., bring me the "furry pink toy") is an open challenge.
- Active perception: reducing task-driven notions of uncertainty by taking actions.

Planning and control:

- Task and motion planning [57] over long horizons.

- For model-based approaches to planning and feedback control, the challenges are similar to those of legged robots (but with some important differences). For hybrid MPC, it is important to plan the sequence of hybrid mode switches (e.g., for in-hand manipulation); this remains a challenging problem to solve at real-time rates (see [58] for recent progress). In addition, deformable objects pose a significant challenge to model-based approaches.
- Model-based approaches to planning/control seem to inherently rely on object-centric representations. How can we take into account uncertainty in the environment (e.g., object properties such as friction, etc.), or perceptual uncertainty (e.g., uncertainty in the shape, pose, etc.)? Coupling perception and action in a semi-robust way is where learning-based techniques currently shine (at least in comparison).

Human-robot interaction:

- Predicting the human's actions in co-operative manipulation tasks.
- Maintaining safety *with contact* (e.g., in assistive feeding tasks). In these settings, the notion of safety goes beyond collision avoidance.

3.6 Collaborative robots

System-level challenges:

- Safe operation around humans. This could mean avoiding contact with the humans altogether, or limiting the amount of force exerted on a human.
- Dealing with all the weird edge cases that show up when you're operating with humans (e.g., people trying to mess with your robot for fun).

Perception:

- Scene understanding. Operating with and around humans can require sophisticated scene understanding. For example, consider a hospital robot that encounters a nurse rushing somewhere with a patient. Should it stop? Probably not! It should probably get out of the way. But, stopping may be desirable in other instances. Similarly, consider a robot that encounters two people who are facing each other. Should it try to go in between them? Not if they are talking! If they are facing away from each other, then going in between them may be the right action.

Planning and control:

- Game-theoretic planning in real-time: ensuring safety without being overly conservative.

Human-robot interaction:

- Specifying a new task to a CoBot quickly and easily. This could be via demonstrations, a no-code programming interface, or a natural language description.
- Effective verbal and non-verbal human-robot communication during a task (e.g., a human telling the robot to do something differently).
- Understanding social norms.

Learning:

- Constructing good models of humans. This is beyond just trajectory prediction (e.g., for AVs). If a robot and a human are collaborating on a task, each agents needs to have a good model of what the other can and will do. This is particularly challenging in richly interactive settings where there is a tight feedback loop between each agent's actions (e.g., collaborative manipulation).
- Few-shot generalization to a new human.
- Learning-based approaches are challenging to deploy directly since it is hard to simulate humans. Thus, offline learning techniques are needed.

4 Current Trends

4.1 Language models

There is a massive amount of excitement around large language models (LLMs), with a number of groups thinking about how to leverage their potential for robotics. Here, I try to summarize strengths and potential research opportunities.

Strengths:

- Excellent ability to generate syntactically correct and plausible text. The length of coherent text that can be generated has grown in leaps and bounds over the past few years.
- Code generation. This is one of the most powerful use cases of LLMs; the ability to generate code allows LLMs to access interfaces beyond text and actually affect things in the world. LLMs' strong ability to write code is something I was personally extremely surprised by, and find quite remarkable.
- Few-shot learning via prompting or fine-tuning.
- Multi-modal models (e.g., VLMs) for performing tasks such as VQA (visual question answering) and open-set classification of images.
- Brainstorming ideas (e.g., "find a counter-argument to argument X"). Broadly, LLMs have a lot of potential in settings where generating ideas is challenging, but there is still a human in the loop to verify and select solutions.
- Ability to enable robots to communicate with humans more easily. Task specification has been a major challenge in robotics (i.e., how do we specify a task to a robot? via a reward function? via code? some domain-specific language?); language provides a natural and flexible interface for humans to communicate tasks and context to robots.
- Some capacity to perform high-level planning (e.g., give me a recipe for X dish with Y constraints). The main power here comes from the fact that the generated plan uses the flexibility and familiarity (to humans) of language-based abstractions (since the output is in the form of natural language); this is different from the kinds of abstractions folks in the task and motion planning (TAMP) literature have focused on, which are more limited in scope.

Current trends:

- Aligning models via instruction fine-tuning and reinforcement learning from human feedback (RLHF).
- Retrieval-based models for knowledge-intensive tasks; more broadly, using external tools such as calculators, search, Python APIs, etc. (see, e.g., [59, 60, 61, 62]).
- Finding mechanistic explanations for what LLMs are doing (e.g., induction heads [63], automata shortcuts [64]).
- Robotics
  - Instruction following (e.g., [56, 65]).
  - High-level planning (e.g., [66, 67]).
  - Co-finetuning vision-language models with robot data (behavior cloning) [68].
  - Code generation (e.g., [69, 70]).
  - Language-driven representations (e.g., [71, 72]).

- Robot learning via language feedback (e.g., [73, 74]).
- Language-conditioned meta-learning (e.g., [75]).
- Dialogue and persuasion (e.g., [76]).
- VQA (e.g., [77, 78]).

Challenges:

- Hallucinations. This is particularly challenging in the context of robotics, where such hallucinations can have physical impact (e.g., a planner based on LLMs could make the robot perform unsafe actions).
- Limited ability to perform many reasoning tasks, especially those that involve physical reasoning (see, e.g., [79]).
- Inability to perform arithmetic (at least out of the box without using external tools). This includes questions that are directly about arithmetic (e.g., what is X times Y?), questions that take the form of word problems that ask the model to count something (see, e.g., [80]), or questions such as "what is the 13th letter of the word 'supralapsarian'?" [81].
- Lack of grounding to the external world. LLMs understand how words relate to one another, but don't have any connection to signals in the real world. Given how lossy language is in describing things, this poses a major challenge to purely LLM-based approaches (see [82] for a longer argument to this effect). But note that this could be alleviated with multi-modal models, especially those trained on embodied data [83].
- Limited understanding of space and time. In its dataset, a model will see statements like "X is true in context C now", and "Y is true in context C now", and will thus not know what is actually true in 2023.
- Lack of knowledge of their own uncertainty. LLMs currently don't have calibrated uncertainty estimates on their outputs (post RLHF) [84], let alone end-to-end performance guarantees.
- Bias (both in terms of the outputs of LLM models, and also bias in terms of underrepresented languages).
- Context-dependent knowledge. This is a subtle one; see end of Section 4.1.2 in [81] for a great example. Whether or not the model seems to "know" something is context-dependent; it seems to get something wrong as part of a longer argument/output, but gets something right when you ask a pointed question. More generally, the prompt and specific dialogue context has a big impact on what it seems to "know". To me, this suggests that we are missing some strong inductive biases; humans know that $(a+b)^2 = a^2 + b^2 + 2ab$, no matter what a and b are, or that $a + b = c \implies 2(a + b) = 2c$, no matter what a, b, c are. For LLMs, these rules seem to be context-dependent. This context dependence is a major strength when generating syntactically correct outputs (since the usage of a particular word is so context dependent), but not so much when it comes to reasoning. This hints at a major missing piece with LLMs: they do not have a consistent and persistent (explicit) representation or causal model of the world; all knowledge is contextual.
- Lack of learning over time (beyond few-shot learning via prompting). The models are not easily updated and thus do not capture the most up-to-date information (without retrieval or other forms of tool use).
- Limited memory: even though the context is long by the standards of previous language models, the effective context (i.e., what the model actually pays attention to) could be much shorter. We see this with recency effects where LLMs have a bias towards outputting option '(e)' in multiple choices questions (with options (a)-(e)). Moreover, the overall context is still limited and thus prevents truly long-horizon interactions (e.g., over days, months, or years).

- Our current paradigm is extremely data hungry. At some point in the near future, we will start running out of internet data to train on.
- Humans have bad intuitions about the strengths and weaknesses of these models. This is partly because of how different the capabilities and limitations of these models are from humans. A human with dazzling ability to produce code is not likely to be unable to count. This makes it particularly hard to deploy these models in use cases with untrained users.
- In robotics, LLMs are enabling unprecedented levels of language understanding, and also seem to have great potential for high-level planning and semantic understanding. However, the core technical challenges of robust manipulation and navigation in novel environments remain significant. As an example, consider the success rates of RT-2 [68] in novel environments (roughly 50%). There is still a lot of work to be done on the core technical challenges highlighted in Section 3.

4.2 Foundation models

Beyond language models, there is an ever-growing list of foundation models. These are models that (i) are pre-trained (usually using a self-supervised objective) on a vast and diverse dataset, and (ii) can be easily fine-tuned for a variety of downstream tasks [85]. The power of these models comes from these two features, which enable excellent out-of-the-box performance on a variety of tasks with minimal tuning (e.g., via prompting or a small amount of fine tuning).

Current progress:

- Language models.
- Code generation.
- Multi-modal models (e.g., vision-language, audio-language, touch-language, etc.).
- Image generation (e.g., DALL-E, Midjourney, etc.).
- Image segmentation [86].

Current trends: It seems quite likely that we will have strong foundation models for the following within the next ~ 5 years:

- Video prediction (at least, short-horizon video prediction, or object-level predictions).
- Human trajectory prediction (e.g., pedestrians in outdoor settings, or perhaps even indoor environments such as buildings).
- Robot task specification (e.g., via natural language, or other modalities such as a visual goal).
- Semantic segmentation.
- Using pretrained foundation models directly or indirectly (e.g., as visual representations) for robotics.

Challenges: A number of the challenges highlighted with LLMs carry over to foundation models more broadly.

Overall, the availability of foundation models will likely alleviate a number of the pain points in the robotics stack and result in improvements across the board. But, I don't think pretrained foundation models alone will solve all the problems highlighted in Section 3. There will inevitably be a distribution shift between the data these models are trained on and the data that the robot observes. This can be due to the nature of the specific environments the robot is deployed in, or due to more subtle issues having to do with feedback (using the outputs of a perception model or human motion prediction model to choose actions for the robot results in a distribution shift).

4.3 Neurosymbolic approaches

While LLMs demonstrate fantastic zero-shot capabilities, they are still limited in many domains such as arithmetic, factual question answering, and planning [87]. There is a lot of recent work on neurosymbolic approaches that seek to merge pretrained language models with structured representations, domain-specific algorithms, and external tools.

Current trends:

- LLMs that can call APIs for external tools such as calculators, search engines, and code interpreters (see, e.g., [60, 61, 88]).
- A special case of this is language-to-code (see, e.g., [62, 69, 89]).
- In robotics contexts, there has been work on combining LLMs with classical planning based approaches, e.g., generating PDDL descriptions from natural language descriptions of planning problems [90, 91, 92, 93], using LLMs as heuristics for classical planners [94], and using LLMs to generate code for a bespoke planning algorithm [95].
- There is also recent work on using LLMs as probabilistic priors for classical inference with graphical models [96].

4.4 Neural scene representations

Neural networks offer compact-yet-rich representations for objects and scenes, e.g., NeRFs [97] and DeepSDFs [55]. There has been a lot of work on improving the basic capabilities of such representations, and also leveraging them for robotics-related applications.

Current trends:

- Faster NeRF training (e.g., [98, 99]).
- NeRFs with fewer images or even one image (e.g., [100, 101, 102, 103]).
- Larger scenes (e.g., [104, 105]).
- Probabilistic scene completion (e.g., [106]; see also with diffusion models [107]).
- Robotics (see [108] for a comprehensive list)
  - Motion planning (e.g., [109]).
  - Pose estimation and reconstruction of unknown objects (e.g., [110]).
  - Mapping (e.g., [111, 112]).
  - Language-queryable neural maps (e.g., [113]), and maps that capture semantic features (e.g., [114]).
  - Active perception (e.g., [115]).
  - Manipulation (e.g., for transparent or shiny objects [54, 53], or from a single RGB image [116]).
  - Simulation (e.g., Scene-to-NeRF for sim-to-real [117]).

5 Conclusions

One interesting observation from looking at the state of the art in autonomous vehicles, drones, legged robots, manipulators, and collaborative robots is the prevalence of relatively "classical" modular architectures (perception-prediction-planning-control) in actual field-deployed systems, in contrast to fully monolithic architectures (pixels to torques via a large neural network). At least as of now, it seems like there is still an important place for classical planning and control techniques (but, of course, augmented with learning-based components). It is interesting to compare this with state of the art systems for playing games such as Go, Poker, and Diplomacy; while the learning-based components of these systems are usually highlighted, they still employ good-old-fashioned search techniques (which, according to some estimates [118], result in the equivalent of a $100,000\times$ scaling up of the underlying learned model). Finding ways to more deeply integrate learning-based components with "classical" planning and control remains an exciting direction towards truly robust and general embodied intelligence.

It is a very exciting time in robotics. There is a frenzy of activity towards incorporating LLMs, VLMs, foundation models, and Nerfs into different parts of the robotics stack. This will unquestionably lead to progress: significantly improved capabilities for natural language understanding and communication, high-level reasoning about semantic concepts, code generation and tool use, high-level planning, improved perception modules, and improved simulation capabilities and scene completion using Nerfs. However, it is instructive to go through each challenge highlighted in Section 3 and see which of these will persist even with recent trends. My feeling is that many of the challenges in "lower-level" parts of the robotics stack (robust perception, control, planning with dynamic constraints, human-robot interaction) continue to remain challenging, especially in applications where one desires low levels of failure in novel scenarios.

I'm excited to see the progress the next few years will bring!

References

[1] Waymo. Sense, solve, and go: The magic of the Waymo driver. URL: `https://youtu.be/hA_-MkUONfw`.

[2] Waymo. Machine learning for autonomous driving. URL: `https://youtu.be/aEWyr3HAO7M`.

[3] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. SWFormer: Sparse window transformer for 3D object detection in point clouds. In *European Conference on Computer Vision*, pages 426–442. Springer, 2022.

[4] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. *arXiv preprint arXiv:2207.05844*, 2022.

[5] Maximilian Igl, Daewoo Kim, Alex Kuefler, Paul Mougin, Punit Shah, Kyriacos Shiarlis, Dragomir Anguelov, Mark Palatucci, Brandyn White, and Shimon Whiteson. Symphony: Learning realistic and diverse agents for autonomous driving simulation. *arXiv preprint arXiv:2205.03195*, 2022.

[6] Kristofer D Kusano, Kurt Beatty, Scott Schnelle, Francesca Favaro, Cam Crary, and Trent Victor. Collision avoidance testing of the waymo automated driving system. *arXiv preprint arXiv:2212.08148*, 2022.

[7] Tesla. Tesla AI Day 2022. URL: `https://youtu.be/ODSJsviD_SU`.

[8] Zoox. Grasp industry talk: Zoox. URL: `https://youtu.be/blzi2d_Y-OQ`.

[9] Robot Brains Podcast. Jesse Levinson: Reinventing personal transportation from the ground up. URL: `https://youtu.be/3_g14RAyOzE`.

[10] Wayve. ICRA 2022 workshop: Fresh perspectives on the future of autonomous driving. URL: https://youtu.be/qzo61V7G1EM?t=759.

[11] Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zak Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. *arXiv preprint arXiv:2210.07729*, 2022.

[12] Skydio. Skydio autonomy engine: Enabling the next generation of autonomous flight. URL: https://youtu.be/5nWnCwO8jIk.

[13] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017.

[14] Antonio Loquercio, Elia Kaufmann, René Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Learning high-speed flight in the wild. *Science Robotics*, 6(59), 2021.

[15] Boston Dynamics. Recent progress on Atlas, the world's most dynamic humanoid robot - Scott Kuindersma. URL: https://youtu.be/EGABAx52GKI.

[16] Boston Dynamics. Do you love MPC? Robot dancing via optimal control. URL: https://youtu.be/mlTLxpKdHfA.

[17] Patrick M Wensing, Michael Posa, Yue Hu, Adrien Escande, Nicolas Mansard, and Andrea Del Prete. Optimization-based control for dynamic legged robots. *arXiv preprint arXiv:2211.11644*, 2022.

[18] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47), 2020.

[19] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

[20] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[21] Laura Smith, J Chase Kew, Xue Bin Peng, Sehoon Ha, Jie Tan, and Sergey Levine. Legged robots that keep on learning: Fine-tuning locomotion policies in the real world. *arXiv preprint arXiv:2110.05457*, 2021.

[22] Gabriel B Margolis, Tao Chen, Kartik Paigwar, Xiang Fu, Donghyun Kim, Sangbae Kim, and Pulkit Agrawal. Learning to jump from pixels. *arXiv preprint arXiv:2110.15344*, 2021.

[23] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. *arXiv preprint arXiv:2211.07638*, 2022.

[24] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[25] Marco Tranzatto, Frank Mascarich, Lukas Bernreiter, Carolina Godinho, Marco Camurri, Shehryar Khattak, Tung Dang, Victor Reijgwart, Johannes Loeje, David Wisth, et al. Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the DARPA subterranean challenge. *arXiv preprint arXiv:2201.07067*, 2022.

[26] Amazon. Robotics and AI in fulfillment at Amazon. URL: https://scs.hosted.panopto.com/Panopto/Pages/Embed.aspx?id=169b346b-ec69-46ef-a713-af2b01695783.

[27] Covariant. Delivering AI robotics at scale: A behind-the-scenes look. URL: https://www.youtube.com/watch?v=1QPr59OUZ5I.

[28] KUKA. Webinar collaborative robots. URL: https://youtu.be/Fx0bE3sDcaE.

[29] Diligent Robotics. Andrea Thomaz, Diligent Robotics: The Robot Brains Podcast. URL: https://youtu.be/nUtwOUNoZw0.

[30] Diligent Robotics Andrea Thomaz. Andrea Thomaz, CEO of Diligent Robotics. URL: https://youtu.be/T3ctJux_7P4.

[31] Alec Farid, David Snyder, Allen Z Ren, and Anirudha Majumdar. Failure prediction with statistical guarantees for vision-based robot control. *arXiv preprint arXiv:2202.05894*, 2022.

[32] Cem Gokmen, Daniel Ho, and Mohi Khansari. Asking for help: Failure prediction in behavioral cloning through value approximation. *arXiv preprint arXiv:2302.04334*, 2023.

[33] Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. *Under Review*, 2023.

[34] Alec Farid, Sushant Veer, and Anirudha Majumdar. Task-driven out-of-distribution detection with statistical guarantees for robot learning. In *Conference on Robot Learning*, pages 970–980. PMLR, 2022.

[35] Rohan Sinha, Apoorva Sharma, Somrita Banerjee, Thomas Lew, Rachel Luo, Spencer M Richards, Yixiao Sun, Edward Schmerling, and Marco Pavone. A system-level view on out-of-distribution data in robotics. *arXiv preprint arXiv:2212.14020*, 2022.

[36] Fabio Ramos, Rafael Carvalhaes Possas, and Dieter Fox. BayesSim: adaptive domain randomization via probabilistic inference for robotics simulators. *arXiv preprint arXiv:1906.01728*, 2019.

[37] Fabio Muratore, Christian Eilers, Michael Gienger, and Jan Peters. Data-efficient domain randomization with bayesian optimization. *IEEE Robotics and Automation Letters*, 6(2):911–918, 2021.

[38] Jacky Liang, Saumya Saxena, and Oliver Kroemer. Learning active task-oriented exploration policies for bridging the sim-to-real gap. *arXiv preprint arXiv:2006.01952*, 2020.

[39] Allen Z Ren, Hongkai Dai, Benjamin Burchfiel, and Anirudha Majumdar. AdaptSim: task-driven simulation adaptation for sim-to-real transfer. *arXiv preprint arXiv:2302.04903*, 2023.

[40] Michael Posa. Multi-contact learning and real-time control. URL: https://youtu.be/2uq3u6oSH6A.

[41] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020.

[42] Digit. https://digit.ml/. Accessed: 2023-02-18.

[43] Yen-Chen Lin, Pete Florence, Andy Zeng, Jonathan T Barron, Yilun Du, Wei-Chiu Ma, Anthony Simeonov, Alberto Rodriguez Garcia, and Phillip Isola. MIRA: Mental imagery for robotic affordances. In *6th Annual Conference on Robot Learning*, 2022.

[44] Michelle A Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Transactions on Robotics*, 36(3):582–596, 2020.

[45] Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021.

[46] Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756*, 2018.

[47] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[48] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. *arXiv preprint arXiv:2209.05451*, 2022.

[49] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. KPAM: keypoint affordances for category-level robotic manipulation. In *Robotics Research: The 19th International Symposium ISRR*, pages 132–157. Springer, 2022.

[50] Tejas D Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. *Advances in neural information processing systems*, 32, 2019.

[51] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR, 2020.

[52] Xingyu Liu, Rico Jonschkowski, Anelia Angelova, and Kurt Konolige. Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11602–11610, 2020.

[53] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. Nerf-supervision: Learning dense object descriptors from neural radiance fields. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6496–6503. IEEE, 2022.

[54] Jeffrey Ichnowski, Yahav Avigal, Justin Kerr, and Ken Goldberg. Dex-nerf: Using a neural radiance field to grasp transparent objects. *arXiv preprint arXiv:2110.14217*, 2021.

[55] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[56] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[57] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4:265–293, 2021.

[58] Wanxin Jin and Michael Posa. Task-driven hybrid model reduction for dexterous manipulation. *arXiv preprint arXiv:2211.16657*, 2022.

[59] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv*, 2208, 2022.

[60] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[61] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

[62] Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128*, 2023.

[63] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[64] Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749*, 2022.

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

[66] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. Do as I can, not as I say: grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[67] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[68] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. *Preprint*, 2023. URL: `https://robotics-transformer2.github.io/assets/rt2.pdf`.

[69] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022.

[70] Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. ChatGPT for robotics: Design principles and model abilities. 2023.

[71] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. *arXiv preprint arXiv:2209.09874*, 2022.

[72] Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023.

[73] Pratyusha Sharma, Balakumar Sundaralingam, Valts Blukis, Chris Paxton, Tucker Hermans, Antonio Torralba, Jacob Andreas, and Dieter Fox. Correcting robot plans with natural language feedback. *arXiv preprint arXiv:2204.05186*, 2022.

[74] Yuchen Cui, Siddharth Karamcheti, Raj Palleti, Nidhya Shivakumar, Percy Liang, and Dorsa Sadigh. "No, to the right"–online language corrections for robotic manipulation via shared autonomy. *arXiv preprint arXiv:2301.02555*, 2023.

[75] Allen Z Ren, Bharat Govil, Tsung-Yen Yang, Karthik R Narasimhan, and Anirudha Majumdar. Leveraging language for accelerated learning of tool manipulation. In *Conference on Robot Learning*, pages 1531–1541. PMLR, 2023.

[76] Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

[77] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.

[78] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[79] Gary Smith. Large language models, though impressive, are not the solution. They may well be the catalyst for calamity. URL: `https://mindmatters.ai/2023/02/lets-take-the-i-out-of-ai/`.

[80] Gary Marcus and Ernest Davis. Large language models like ChatGPT say the darnedest things. URL: `https://cacm.acm.org/blogs/blog-cacm/268575-large-language-models-like-chatgpt-say-the-darnedest-things/fulltext`.

[81] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023.

[82] Jacob Browning and Yann LeCun. AI and the limits of language. URL: `https://www.noemamag.com/ai-and-the-limits-of-language/`.

[83] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[84] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[85] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[86] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[87] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jian, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D Hwang, et al. Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.18654*, 2023.

[88] Ehud Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hofit Bata, Yoav Levine, Kevin Leyton-Brown, et al. MRKL systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *arXiv preprint arXiv:2205.00445*, 2022.

[89] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023.

[90] Katherine M Collins, Catherine Wong, Jiahai Feng, Megan Wei, and Joshua B Tenenbaum. Structured, flexible, and robust: benchmarking and improving large language models towards more human-like behavior in out-of-distribution reasoning tasks. *arXiv preprint arXiv:2205.05718*, 2022.

[91] Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. Leveraging pretrained large language models to construct and utilize world models for model-based task planning. *arXiv preprint arXiv:2305.14909*, 2023.

[92] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023.

[93] Yaqi Xie, Chen Yu, Tongyao Zhu, Jinbin Bai, Ze Gong, and Harold Soh. Translating natural language to planning goals with large-language models. *arXiv preprint arXiv:2302.05128*, 2023.

[94] Tom Silver, Varun Hariprasad, Reece S Shuttleworth, Nishanth Kumar, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Pddl planning with pretrained large language models. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[95] Tom Silver, Soham Dan, Kavitha Srinivas, Joshua B Tenenbaum, Leslie Pack Kaelbling, and Michael Katz. Generalized planning in pddl domains with pretrained large language models. *arXiv preprint arXiv:2305.11014*, 2023.

[96] Belinda Z Li, William Chen, Pratyusha Sharma, and Jacob Andreas. Lampp: Language models as probabilistic priors for perception and action. *arXiv e-prints*, pages arXiv–2302, 2023.

[97] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[98] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022.

[99] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*, 2023.

[100] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. Sinnerf: Training neural radiance fields on complex scenes from a single image. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 736–753. Springer, 2022.

[101] Zuoyue Li, Tianxing Fan, Zhenqiang Li, Zhaopeng Cui, Yoichi Sato, Marc Pollefeys, and Martin R Oswald. Compnvs: Novel view synthesis with scene completion. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, pages 447–463. Springer, 2022.

[102] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. *arXiv preprint arXiv:2303.07418*, 2023.

[103] Hanwen Jiang, Zhenyu Jiang, Kristen Grauman, and Yuke Zhu. Few-view object reconstruction with unknown categories and camera poses. *arXiv preprint arXiv:2212.04492*, 2022.

[104] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022.

[105] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12922–12931, 2022.

[106] Dongsu Zhang, Changwoon Choi, Inbum Park, and Young Min Kim. Probabilistic implicit scene completion. *arXiv preprint arXiv:2204.01264*, 2022.

[107] Jumin Lee, Woobin Im, Sebin Lee, and Sung-Eui Yoon. Diffusion probabilistic models for scene-scale 3d categorical data. *arXiv preprint arXiv:2301.00527*, 2023.

[108] Awesome-implicit-nerf-robotics. `https://github.com/zubair-irshad/Awesome-Implicit-NeRF-Robotics`. Accessed: 2023-04-22.

[109] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2):4606–4613, 2022.

[110] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Muller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. *arXiv preprint arXiv:2303.14158*, 2023.

[111] Jiahui Huang, Shi-Sheng Huang, Haoxuan Song, and Shi-Min Hu. Di-fusion: Online implicit 3d reconstruction with deep priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8932–8941, 2021.

[112] Joseph Ortiz, Alexander Clegg, Jing Dong, Edgar Sucar, David Novotny, Michael Zollhoefer, and Mustafa Mukadam. isdf: Real-time neural signed distance fields for robot perception. *arXiv preprint arXiv:2204.02296*, 2022.

[113] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*, 2022.

[114] Kirill Mazur, Edgar Sucar, and Andrew J Davison. Feature-realistic neural fusion for real-time, open set scene understanding. *arXiv preprint arXiv:2210.03043*, 2022.

[115] Soomin Lee, Le Chen, Jiahao Wang, Alexander Liniger, Suryansh Kumar, and Fisher Yu. Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields. *IEEE Robotics and Automation Letters*, 7(4):12070–12077, 2022.

[116] Valts Blukis, Taeyeop Lee, Jonathan Tremblay, Bowen Wen, In So Kweon, Kuk-Jin Yoon, Dieter Fox, and Stan Birchfield. Neural fields for robotic object manipulation from a single image. *arXiv preprint arXiv:2210.12126*, 2022.

[117] Arunkumar Byravan, Jan Humplik, Leonard Hasenclever, Arthur Brussee, Francesco Nori, Tuomas Haarnoja, Ben Moran, Steven Bohez, Fereshteh Sadeghi, Bojan Vujatovic, et al. Nerf2real: Sim2real transfer of vision-guided bipedal motion skills using neural radiance fields. *arXiv preprint arXiv:2210.04932*, 2022.

[118] Robot Brains Podcast. Noam Brown. URL: `https://www.therobotbrains.ai/who-is-noam-brown`.