Conference on Robot Learning (CoRL) 2023 Note

Seungchan Kim (CMU)
seungch2@andrew.cmu.edu | https://seungchan-kim.github.io/

November 6-9th 2023

Table of Contents

This note is taken during my attendance at the CoRL 2023 held in Atlanta. Please feel free to distribute. If you have any sugestions for revisions or feedback, please send them to my email at *seungch2@andrew.cmu.edu*. Unfortunately, I wasn't able to participate in all sessions. For the parts I missed, please refer to the CoRL 2023 website

1 Day 1 (Nov 6th): Workshops

On the first day, I attended multiple workshops. I moved to different workshops at different timeslots. I mainly attended Learning Effective Abstractions for Planning (LEAP) Workshop, Pre-Training for Robot Learning (PRL) Workshop, and Language and Robot Learning (LangRob) Workshop.

1.1 <LEAP Workshop>

Interests in learning abstractions for planning, how to learn the abstractions, how to use it for planning, and how to use it effectively?

Paper themes: language models, vision-language models, diffusion models, long-horizon planning, skill discovery, integrated planning and RL, humans in the learning and planning loop

Questions of the day
1. When, where, and from what **should** abstractions be learned?
2. What is the **right objective** for abstraction learning?
3. To what extent should the abstractions be **interpretable**?
4. How can and should we leverage **foundation models**?
5. To what extent is **natural language** a useful representation for abstraction learning?

<Invited Talk 1> Aviv Tamar, **"Explore to Generalize"**

Why do we learn abstractions? Motivation comes from difficult planning problem, where human can identify structures in solution (state abstraction, tempoeral abstrafction, hierarchy), use this structure to solve the problem.
Approach: just *learn* to solve the problem (abstractions will be automatically identified if needed)

Few-shot generalization in RL: train an agent sovle different training problems; agent will learn how to solve a new test problem in few-shot

Generalization is difficult:

- ProcGEN benchmark - still a challenge.
- Theory - PAC bounds exponential in #DoF of MDP distribution $P(M)$
- Overfitting - easy to "memorize" train task solutions

Can we learn to solve mazes with less than 10K examples? What are we missing?

Previous dominant approaches:
- Inductive bias for a planning policy: Value iteration networks, domain specific
- Invariant policies: to appearance (augmentation), to length of task (IDAAC) [1]

Recall

$$L(\pi) = E_{M\sim P} E_{\pi;M}[\sum_{t=0}^{T} C(s_t, a_t)]$$

(Zero-shot T=1, few-shot T=few episodes)

Explore at the test time!

Observation: *Max Entropy exploration generalizes!* Why? Exploration behavior is harder to memorize! [2]

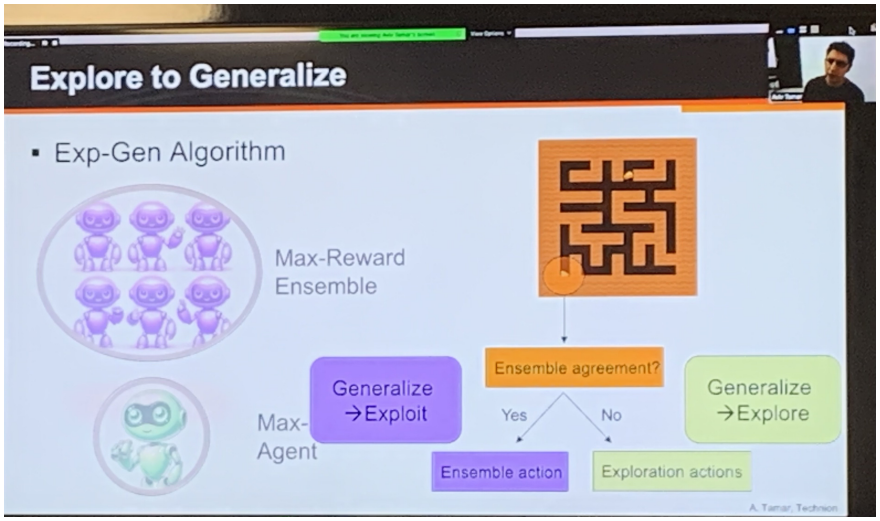

Figure 1: Explore to Generalize [3]

Exp-Gen Algorithm [3]: Max-Reward Ensemble with Max-entropy agent; during the test time: ensemble agreement? Yes → ensemble action. But if no? → exploration actions

Summary [3]

- Exploration at test time → Generalization
- Max-Ent exploration policies generalize well
- Exp-Gen idea: detect uncertainty (ensemble), exploration when uncertain, otherwise exploit
- Abstraction for planning → abstraction for exploration.

<Invited Talk 2> Shiqi Zhang - **"Symbolic State Space Optimization: Learning Abstractions for Mobile Manipulation"**

Symbolic state optimization work [4]

Task planning - standard way is.. to locate one object here, there is a symbolic location, navigate to this location, grasp it, and move somewhere...etc, etc.. if you do this task planning this way → long sequence of actions!

Example Task: Tidy home
Previous task and motion planing methods assume predefined symbolic locations

The focus is to optimize abstracted locations and their geometric groundings!

Instead of a long sequence of plans, group some of them.
Care both feasibility and efficiency.

Symbolic State Space Optimization (S3O); not only doing optimizing for task planning, but also optimize the task planner. Challenge: how to reduce search complexity? Leverage computer vision to gauge feasibility.

Takeaway message:

- Learning abstraction is important for robots to deal with the complexity of the real world
- Challenge: abstraction effectiveness depends on the robot kills, planning goals, and real-world configurations
- Symbolic State Space Optimization paves the way for learning abstracted locations for mobile manipulation.

<Invited Talk 3> Alex Lagrassa - **"Transition model abstractions for efficient and reliable planning"**

Transition models are imperfect abstractions of the real world. However, the real world is quite complex. We use anlaytical models, simulator, learned models.

The right model abstractions are a key to good plans:

Expressiveness → compute plans to reach a goal, execute plans to goals, satisfy constraints, avoid unwanted effects.
However this model should also be **simple**.
Simplicity → Compute quick plans, within memory limits.
Trade-off between expressiveness and simplicity

How should planners select how and where to apply transition models?

Key concept: **Model Precondition** [5]

subset of $S \times A$ space where a model is sufficiently accurate

1. Selecting between multiple models in planning
Use model preconditions for planning with multiple models
- prioritize faster models, apply models wehre sufficiently accurate
What does 'sufficiently accurate' mean?
Predictive accuracy as a proxy for sufficient; model deviation (distance between predicted next state and true next state)

However, we don't know the ground-truth state; we learn and plan with model preconditions by fitting model deviation estimator (MDE)

Multi-representation A* + Prioritized SElection (PS) → fastest model

Prioritize expansion with faster models
combine searches with different models; select model based on optimality requirements, computation time per model

MRA* with model preconditions improves performance

2. Focusing dynamics adaption with model precondition

What if the current model is insufficient/ Adapt model for more complex dynamics? Adapt a learned model

Limitation: real-world data is expensive. Sequential nature of trajectories also aggravates this.

Data-efficient learning of model preconditions

Informing active learning with deviation estimate
1) Generate task relevant candiate trajectories
2) Compute step-wise utility using estimated model
3) Aggregate
4) Execute highest utility trajectory

Active learning of MDE improves data-efficiency.

3. Planning-guided model generation: generate dynamics models for a task; adaptively focus model fidelity.
what is necessary to select good actions from that state?

<Invited Talk 4> Amy Zhang, **"Value-Based Abstractions for Planning"**

What makes representations amenable to planning?
Build a graph from data; use dijkstra's to construct a global metric and latent representation space. → Downside: restricted to small dataset

Plan2Vec: Unsupervised representation learning by latent plans. [6]

Learning a Universal Value Function

$$V * (Observation, Goal)$$

V*: general notion of goal-directed task progress. Rich visual representation so that V* can be expressed

Key Idea: learning from human videos as a BIG Offline goal-conditioned RL problem
VIP: Towards Universal Visual Reward and Representation via Value-implicit pre-training [7]; Offline dataset of diverse human videos (human videos are rich sources of gobal-directed behavior); offline value learning on human videos

Towards Universal Visual Reward and Representation via Value-implicit pre-training
Downstream tasks: imitation learning, trajecotry optimization, online RL, few-shot real-world offline RL
- VIP robustly minimizes both robot and object pose errors

Extension to language: [8] Solving diverse language-specified tasks in diverse environments. Previously:
- Lacks visaul rewards/feedback grounded in language
- Generalizable vision-language rewards and representations

LIV [8]: simple and effective algorithm for peretraining multimodal value representations from large-scale offline human video data

1.2 < PRL Workshop >

<Invited Talk 1> Chelsea Finn, **"Can robots fine-tune autonomously?"**

Context: the advent of broad robot pre-training data nowadays
→ Bridge V2, RH20T, open X-embodiment

Main Question: How can we make fine-tuning as easy and fast as possible?

1. Fine-tuning to new tasks

How can we make fine-tuning as easy and fast as possible?

Can the robot fine-tune autonomously? e.g. with RL?

Q: What prevents us from using RL for easy fine-tuning?
A: - need feedback from reward function

- need to autonomously retry the task
- should incorporate prior experience

Q: Can we just run RL algorithms without resets?
A: ability of RL severely decreases with less reset [9] → we cannot apply vanilla RL without reset

Alternative paradigm for auotnomous robot learning:

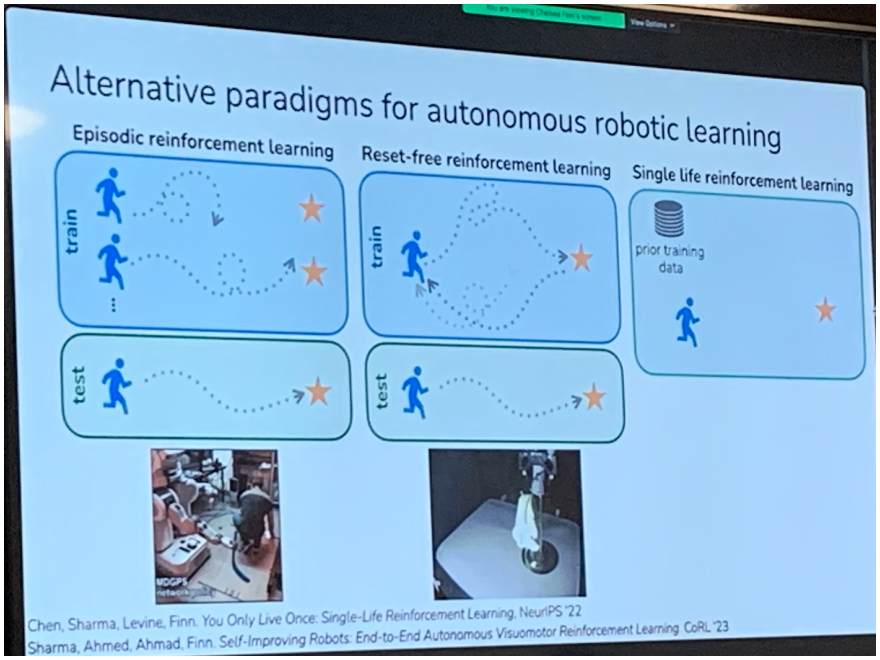

Figure 2: episodic RL vs reset-free RL vs single-life RL

Self-improving robot [10]

Can robots pretrain with broad off-the-shelf data, and fine tune with minimal human supervision?

RoboFUME: making robot fine-tuning easy by improving autonomoy. [11]

- diverse off-the-shelf demo data → small amount of target task demos → pretrained policy
- pretrained VLM → finetuned VLM reward model
- Pretrained policy + finedtuned VLM reward model → online finetuning with minimal human inverventons → fine-tuned policy

RoboFume: key design choices [11]

- learn language-conditioned policy and critic on all data
- fine-tune mini-GPT4 to predict binary success/failure using prior and new demo data (including small number of failed demos)
- cal-QL algorithm for RL pretraining and fine-tuning
- behavior cloning regularization

Analysis: are prior data, language-condition and online data important?
- Without online fine-tuning: performance drops significantly
- offline only (without language conditioning): performance drops more
- offline only without prior data: performacne drops a lot

2. Fine-tuning to new conditions

Can the robot adapt during deployment?

Alternative paradigms for autonomous robot learning: single-life RL

Self-improving robots [10]

single-life RL: given prior data in train env, agent has one life to autonomously complete task in novel, OOD scenario.
can we fine-tune with RL in a single episode at test time?
can we fine-tune with vanilla RL, but at test time?[12]

Idea: can robots adapt both in the space of actions and in the space of behaviors? not only the low-level action, but also the high-level behaviors?

Robust autonomous modulation (ROAM)
Key Idea: use the value functions of the behaviors to identify an appropriate behavior at every timestep.
(1) Further train each behavior's value function to encourage identifiability of familiar states; Bellmen error + additional cross-entropy error term.
(2) At test time, sample behavior, sample actions, optionally fine-tune

Related work: Adapt-on-the-go: Behavior modulation for single-life deployment (Chen et al 2023)

+ doesn't require a separate high-level controller
+ agnostic to how pretrained policies and value functions are obtianed
+ simple mechanism for adapting within a single episode to a variety of situations

Takeaway: with important modifications, RL enables autonomous fine-tuning.
1. fine-tuning with reset-free RL:

→ *with careful design choices, offline-to-online RL enables robots to learn new tasks with off-the-shelf prior datasets*
2. adaption with single-life RL:
→ *autonomous adaptation at high and low levels enables robot to handle OOD conditions*

<Invited Talk 2> Kristen Grauman, **"From Goals to Grasps: Learning About Action from People in Video"**

Q: What could be learned by watching people?
A: First-person video reveals how people move in the environment, direct attention, interact with people, interact with objects

Learning about action from video → towards intelligent embodied agents that learn by watching people interact with the environment and objects:

- hierarchical video-language embeddings
- ego-exo alignment from unparied data
- environment context from ego-video
- grasping with human-like poses

Ego4D: everyday activity around the world [13] → 3670 hours of in-the-wild daily life activity; 931 participants from 74 worldwide locations; benchmark tasks; multimodal (audio, 3d scans, imu, stereo, multi-camera, text narrations).

Idea: ***robots need to understand the intent of actions!*** Robots need to understand why the action is happening.

Ego4D data has text narrations: dense descriptions of camera wearer activity **and video-level summaries.** 13 sentences per minute; 4 million sentences.

1. Hierarchical video-language learning
Existing methods: match short clips to corresponding narrations.
Instead, the idea is to **summarize!** instead of long sequences of short actions (fetch the water into the pot, cuts the onions with the knife, drops chopsticks on the board..), just *summarize* sentence! (make salad dressing with some oil and sauce..) [14]

Idea: ***video-language embedding learning, representing the hierarchical relationship between action descriptions (what) and higher-level semantics (why)***

Child-level what; parent-level why. HierVL [14] achieves parent-level summary and child-level narrations. Pretrained HierVL features provide strong performance on multiple downstream video tasks.

2. Ego-exo alignment from unpaired data:
Robots need fine-grained video understanding

Fine-grained activity recognition
Challenges:

- subtle differencees between action phases
- major viewpoint variation

→ Importance of exocentric and egocentric relationship; want to relate the actions of the demonstrator (exo) to the actor (ego)

Ego-exo alignment (AE2) for feature learning
Idea: ***learn fine-grained view-invariant representations by temporally aligning egocentric and exocentric videos of the same action*** [15]
Unlike prior work, it operates with unparied, unsynchronized data while allowing dramatic viewpoint differences.
→ AE2 consists of (1) object-centric encoder (→ extracts object-centric representations to bridge the ego-exo gap)
(2) contrastive regularizer (→ avoid collapse and more robust to noisy videos)

3. Environment context from ego-video
Robots need environment-aware video representations

Instead of 'Videos as fixed-size chunks of frames'
→ Videos in spatial and activity context
→ Representations that are predictive of the camera-wearer's local surroundings
: Learn using videos from simulated agents in 3D environments → Transfer learned environment representations to enhance ego-video understanding

EgoEnv: Human-centric environment representations from egocentric video. Nagarajan et al. 2023
→ Downstream task 1: room prediction (which room is this? bathroom, living room, kitchen, staircase?)
→ Downstream task 2: Episodic memory natural language query (e.g. when did I visit bathroom?)

4. Grasping with human-like poses
Idea: **Let agent prefer grasping at human affordance regions and with human-like hand poses**
In-the wild Youtube videos → dexterous robotic grasping

Summary

- Ego4D as a resource for learning human behavior and interaction
- Hierarchical video-language learning for video features [14]
- Ego-exo alignment for fine-grained activity understanding [15]

- Environment-aware ego-video representation (Nagarajan et al. 2023)
- Dexterous grasping with affordance points from video

1.3 < LangRob Workshop>

< Panel Discussion >
Panelists: Zsolt Kira (ZK), Jeannette Bohg (JB), Ed Johns (EJ), Fei Xia (FX), Sergey Levine (SL)

Q: Opinions on the term foundation models (in the context of robotics)

FX: large enough model that is trained on large, diverse datasets. Can come up with emergent capability, strong generalization beyond what's in the training. For robotic foundation model, it should be a model that can do reasoning, manipulation in embodied setting, etc.

JB: Need to be trained a large massive data. (whatever domain). Often trained using self-supervised method. Being applied zero-shot to downstream tasks.

EJ: As an academic in a small lab, it is something we can take from big tech company. (laugh)

ZK: ...(sorry I missed his answer)...

SL: ... really great model with a lot of positive features that people want to use; great model in the context of robotics: we're pretty close there.

Q: What Large Language Model cannot do soon?

FX: real, fine-grained manipulation, or whatever that requires some know-how (e.g. cycling); Can vision-language model learn such fine-grained details?

JB: In principle, if you can sample enough data; it's possible. The right question is, for what problem can you not feasibly collect data? Will we be able to get the data, the trillion data, in robotics? Just million trajectories is not the kind of diversity we're looking for. More than that we need.

EJ: LLMs have superhuman or same ability as human experts in certain field; I think we can expect LLMs to be at the point where they are good as best roboticists in the world. But even best roboticists don't know how to solve some tasks. some vision-motor control, etc. Fine-grained manipulation tasks where we don't have sufficient representations. some areas even the best roboticists can't design the features.

ZK: Long-horizon planning is a challenge. Long-horizon planning is something

that LLMs can't solve soon? Another interesting one is, creativity, or out-of-box thinking, for example, whatever Einstein came up with at the time given all the knowledge; completely different way of thinking.

SL: Short, uncontroversial answer - for language model in particular; theory of communication; language is beneficial for communication. If I want to communicate something to you, I'm transmitting those bits that you don't know that I know. There is absolutely no point transmitting things both of us know. That's inefficient communication; which means, language is a communicating those things that some people know, some people don't know. There is no point transmitting what all people know. But we know, trouble for robots is that it's what all people know, things like body proprioception. I said this is non-controversial, because nobody is actually requiring LM will just be about language; we're increasingly seeing more multi-modal models.

Long, complex answer: models that are trained for predictions (general prediction models) are not doing rational decision making; rational decision making maximizes utility. Prediction does not necessarily lead to rational decision making. It still doesn't make them rational. In the long-run, we need ways to use language models in the context of planning and RL loop; torture the predictive model to more rational decision-making way.

Q: Spicy question. What are the inductive bias that are original from robotics, not stealing ideas from vision and NLP community?

SL: It's not stealing. The entire field of computer vision uses the idea developed for neural network, which were from ML community. Ideas are ideas; just use the best ideas.

ZK: I agree. There is a lot of low-hanging fruit nowadays. This thing came out only a year ago; people are exploring a lot; eventually it will plateau; assume at that point, we will have to come up with new solution.

EJ: When I say end-to-end robot learning, what does that actually mean? There is a neural network; mapping predictions to some actions. Difference between robotics; there is a lot of things we can model and analyze analytically. Robotics isn't necessarily same with other field; for example, physical model of the world. A big breakthrough can be, combining classical analytical methods with learned behaviors; rather than learning everything from scratch. Learning everything from scratch might not be the best approach.

JB: Always good to ask what's at the *end* of the end-to-end; I'd also like to highlight hardware; people recently have come up with so great hybrid solutions. Look at the cool new haptic sensors that came out in last few years (e.g. vision-based haptic sensors); such things are going to make huge difference. Another thing is a cool new hand designs; some crazy hand with active surfaces, anthropomorphic paradigm, etc.

FX: A serious point is, we contribute some of original things to the research; robotics is about embodied AI, it is about being active; for example, the concept of affordance is also very helfup beyond robotics. Robotics also does lots of reasoning and planning, which is also very helpful to other AI communities.

Non-serious point; roboticists are generally very optimistic! Robotics is not working yet! it will work in the future! We're going to bring some optimism to the field.

Q: Question on industry vs academia. Question on bigger model size. What should academic research focus on? What should industry research focus on?

FX: For the foundation model, big advantage is that you only need to train the model once; academics can use that; large model, you can still play a lot with the trained model. For example, prompting tasks. Also need to tackle alignment, safety, etc. Industry and academia can have different responsibilities in development. Collaboration is more needed.

JB: Collaboration. I'm really excited about collaboration like open X-embodiment - I reallly like that idea. That's great. Another answer is... Fei-fei Li once said, you should always think about something that is 10 years ahead that industry is not thinking. Many of the big ideas originally came from academia. Also, study everything. You don't know at some point what's going to be important!

EJ: It does force you to try to be more data-efficient than in industry, where you can access lots of data and engineering and computes; for example in imitation learning a lot of work focuses on very efficient imitation learning methods, which might not be so important in industry. In academia, it forces you to think differently and try to be more efficient.

ZK: Research is like a genetic algorithm. We don't ultimately know which idea will come out. Diffusion models was originally a very math-heavy thing; there are different flavors in the research you can do. Look out what are the 10-year problem that industry is not tackling now. At community-level, we should ensure that there is a right incentive for the long-sighted research, though, in academia.

SL: Panelists have already provided interesting points. I'll just add one. One thing to consider is, it's not much about academic vs industry. You can think about it about fundamental vs applied research; especially in modern ML, if you want to make something big, it might not be fundamental; if you are working on something fundamental; it might not be big. There is some kind of trade-off between fundamental vs big.

Q. For us, as reviewers, what should be the right metrics to evaluate papers on? How do we review the papers (foundation models for robotics)?

SL: Very carefully (laugh). There is not much metric; just remember what science is supposed to do; what is the claim? is this claim sound? how is it validated? etc...; there is nothing particularly profound about that.

ZK: Similar with SL. Does paper make a proper claim? It has to be a significant claim that people will care about. Does the paper provide evidence? What kind of contributions are made? Empirical? Theoretical? Either is fine. Just have an open mind as a reviewer.

EJ: Area chair is so important in robotics. Compared to CV or ML, which have narrower, common benchmarks, in robotics, the benchmark can be wider and diverse. A lot of people create their own benchmarks. Some young reviewers might not understand and just miss it. When a robotics paper did it differently, junior reviewers might think that the paper didn't do enough; in that case, the area chair should be the more responsible person to evaluate methods and compare the benchmark. Rather than a competition between this paper and that paper, more understanding of the paper..

JB: Agree with EJ. One thing we don't want to adopt from CV is benchmark-driven, standard benchmark-driven research. In robotics, it should be different; although benchmark is needed, robotics should aim wider. Robotics has much more space. Define a new problem. New benchmark, beyond the current benchmark.

FX: I wish the review process in robotics to be a little bit less divided. For example, between academic and industry; there are some prejudices; also the background is more diverse. We have different generations of researchers from different schools of thoughts. Let's be more open minded.

JB: Also a good point. I often see that a lot of reviews are really mean; sometimes it's almost a hate speech. Reviewers should always remind themselves that there is probably a young student who's reading this whose soul is literally crushed.

Q. Give us a statement that other panels will disagree with you?

EJ: I'll say something that at least someone here will probably disagree with. Locomotion and navigation is solved.. anything left is manipulation.

JB: I think simulation is a nice tool, but it's not gonna get us all the way.

ZK: I think simulation is one route to scale; if we believe scale is something that other fields made successful.. real world is awful. It's also a big chicken and egg problem. Tesla solved this by.. they actually have cars, they actually drive not by themselves but by human drivers. They gather lots of information from the car data; for in robotics, home robotics, for example, it's not clear how to do that?

Controversial statement is this: nobody is actually working on the biggest bottleneck for getting robots at home; what about reliability? What is the path to having home-robot?

SL: Different problems in robotics are much more similar to each other. If you work on robotic manipulation, you know a lot about manipulation; if you work on navigation, there is a lot of navigation thing going on; I think in the next few years we will increasingly see that they are similar, using similar/same sets of tools for solving manipulation and navigation; cross-section problem. (Autonomous driving is a bit different; they have safety concerns, and they have tons of more data) More and more, the areas are united; if you figure out robust method to one area, it will be relevant with other area.

FX: Optimistic take is, over the next couple years, robotics and embodied AI will be at the central place of AI development. LLMs will soon run out of tokens; people will focus more on active learning and embodied AI.

Q: (Sorry, I didn't hear the question well)

SL: Bigger picture wise: Tesla example is apt. Being smart with collecting data, once it is real. The single best thing we can do, build a system that people actually use (hundred, million robots)..

FX: If you don't have deployment and data, how do you start? probably start with small-scale, and have safety constraints, then gradually scale that up.

ZK: One solution could be putting lots of robots in home, controlled by humans... somehow should have lots of robots in actually home, to scale.

EJ: If human is there to provide rewards and reset the environment, and give feedback to robots, why not spend time providing demonstrations.

Q: What is something that is not able to get from vision and language.

JB: In vision and language, there is fundamentally a lack of actions. What is the effect of your action and executions. That's missing in CV and language.

FX: In embodied AI domain vs LLMs: LLMs use next-token predictions. It's about predicting next tokens. In robotics, people have extensively used optimization to derive something; foundation model for reasoning, planning ability is needed.

SL: While there is a lot we can say about the quality of embodied data over, passive data from web; in the long-run, real answer will be we will want quantity. All of the data you can mine from web, are created from humans. Millions, billions of people are laboring to create all of these stuffs to put on web that we use to train our AI system. If the point of building robot is to automate

things that humans do themselves, ...(sorry, missed this part)...certainly, coming back to autonomous driving example, if every car is partially an autonomous car, then the data collect for autonomous driving has to be larger than the data-collect of humans-driving. In the case of dish-washing, far more data for dishwashing robot needed? In the long-run, we need it much much larger, and keep it bigger.

1.4 <PRL Workshop>

<Invited Talk 3> Vincent Vanhoucke, **"Embodied Foundation Models"**

Robotics in one slide: Perception, Planning, Actuation

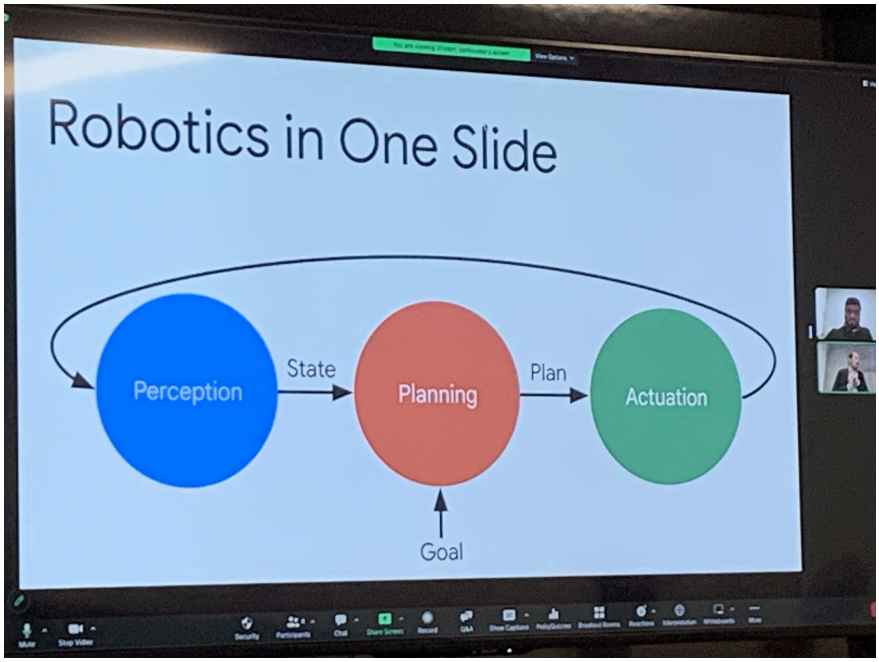

Figure 3: Robotics in one slide

But he argues everything is wrong with this slide.

What he suggests: The solution to Robotics in one slide: Giant AI Models that encompass perception, planning, actuation.

Think about the information flow between perception and planning. Think about a cup in the kitchen. What is the state information robot can get from perception? 6DOF pose? collision information? Is this enough?

Probably you want the robot to know more about it; full/empty, hot/cold, rigid/squishy? whose mug? heavy/light? someone holding it?
All these information are essential. You need a very large, deep amount of information that flows from perception to planning bottleneck.

Think about a dog. What is the state information that we want the robot to know about the dog?

In real-world settings, the API between perception and planning is essentially AI-complete. Reductionist, fixed ontologies don't stand a chance. Now what?

Two lines of inquiry: 1. Natural Language 2. Raw Pixels

First approach is, SayCan: Planning with LLMs. [16] Pairing LLM with value functions → SayCan.

Next approach is PaLM-E: an embodied multimodal language model [17]. Used for mobile manipulation, task and motion planning, etc.
PaLM-E is massive with 562B parameters. Yet we observe positive transfer across robots using little robot data.

Are the abstractions (perception + planning + actuation) useful?

RT-1: Robotics Transformer v1
- RT-1 is able to reach 100% performance on seen tasks,
- while maintaining better robustness to unseen variability

RT-1: diversity is all you need? Do we need more data? Yes, more data is better. But more *diverse* data is better.

RoboCat
Bootstrap and self-improvement.
(https://deepmind.google/discover/blog/robocat-a-self-improving-robotic-agent/)
→ Robocat adapts to unseen tasks and embodiments with limited dataset of demonstrations; incorporates skills back with self-improvement and performs better by scaling the trained data

RT-2; Check at the CoRL 2023 poster session!
RT-2 features emergent transfer.

Closing comments: *Robotics is ready for its ImageNet moment. Time for the community to take inspiration from computer vision and invest in data.*

<Invited Talk 4> Dhruv Batra, **"How does one benchmark foundation models for embodied AI and robotics"**

We want generally intelligent and broadly capable robots. Hypothesis: pretrain-

ing on out-of-domain data as a strong prior.

View from other AI communities:
NLP:
- base task: masked language modeling
- model: transformer;
- dataset: common crawl, wikipedia...
- downstream tasks: sentiment analysis, dialog,
2D vision
- base task: jigsaw, invariance to data augmentation, MAE on patches, self-distillation
- Model: CNNs, transformers
- Dataset: ImageNet, Ego4D, etc,
- Downstream tasks; detection, segmentation, captioning,

What about the Embodied AI?
- base task:
independently at each time t? or over the entire trajectories?

Dataset?
- which trajectories? offline dataset? oracle and agent in sim?

Model: transformer?

Downstream tasks: navigation, rearrangement, manipulation? offline datasets or sim or deployed on robots?

Question to ask: *If I claim I have made progress towards a foundation model for Embodied AI, what evidence would you demand from me?*

Generalization!

- new starting states of the robot
- new environment components (objects, scenes, etc)
- new goals, tasks, skills
- new embodiments

How does one benchmark? simulation-only? sim2real transfer and predictivity? hardware-only?

Simulation-only
VC-1: A visual foundation model for embodied AI.

Which one is the best?

- different pretraining algorithms

- evaluation on different tasks
- with different settings: few-shot vs many shots; imitation learning vs RL; with vs without fine-tuning

Evaluating PVRs
MVP; R3M; CLIP

Scaling datasets
- overall dataset scaling is helpful
- adding a computer vision dataset like ImageNet helps
- cross-domain diversity > within-domain diversity

VC-1 adapted
- End2End fine-tuning
- MAE-based adaption on in-demand data

Sim2Real transfer and predictivity [18]
Question: *If method A outperforms method B in simulation, how likely is the trend to hold in reality?*

2 Day 2 (Nov 7th)

2.1 <Oral Session 1: Manipulation >

< Talk 1> Jennifer Grannen, Yilin Wu, Brandon Vu, Dorsa Sadigh. **Stabilize to Act: Learning to Coordinate for Bimanual Manipulation** [19]
https://openreview.net/pdf?id=86aMPJn6hX9F

Goal: Learning Bimanual Tasks
Exploring to learn; Collecting demos for learning.

Make the insight for stabilizing for bimanual tasks. When one arm is stabilized, the other arm is acting. There is a stabilizing arm, and an acting arm.

BUDS: Bimanual Dexterity from Stabilization Two subproblems

- Stabilizing: stabilizing position model → stabilizing keypoints → noncompliant stabilizing → stabilizing actions
- Stabilizing actions to go the acting module, which does imitation learning. Outputs acting action to form the bimanual policy
- There is a restabilizing classifier, that takes images as inputs, and outputs binary output (update stabiliizng position or continue?)

Experiment shows how important it is to have stabilizing arm / restablilization. (When zipping a jacket); another task is a marker cap, which requires a high precision. Cut vegetable task - baseline doesn't learn any effective stabilization policy; BUDS does restabilization after every cut. Also very generalizable to OOD.

<Talk 2> Vainavi Viswanath , Kaushik Shivakumar , Mallika Parulekar , Jainil Ajmera , Justin Kerr, Jeffrey Ichnowski , Richard Cheng, Thomas Kollar, **HANDLOOM: Learned Tracing of One-Dimensional Objects for Inspection and Manipulation** https://openreview.net/pdf?id=WWiKBdcpNd

What is cable tracing? Applications: broadly used in home, hospital, etc.

Objective: given grey-scale image and one endpoint position, find the trace.

GPT-4V can't be a solution to the complex cable tracing tasks.

Approach: branch points. When we encounter crossings, explore the multiple branches. Previous works are limited to orthogonal crossings.

Non-orthogonal crossings: when cable segments are twisted with one another.

HANDLOOM: 1. autoregressive cable tracing. → 2. crossing classification → 3. full cable state estimation.
Key idea: *inductive bias in local crops is effective for 1D deformable object state estimation.*

Architecture: sim+real data → UNET → autoregressive trace predictor that outputs heatmap → BCE loss.

Notable Experiments:

- Generalization to other tables.
- Multiple Cables
- Untangling Long Cables

Summary: novel methods for cable inspection, state-based policies, knot detection, and autonomous untangling.

Q: How big are the crops? how do you choose the size?
A: Crops are approximately 30 pixels wide; we ran some experiments to find the size; if they are too big; lots of distraction; if too small; can't effectively leverage the context.

<Talk 3> Haochen Shi, Huazhe Xu, Samuel Clarke, Yunzhu Li, Jiajun Wu, **RoboCook: Long-Horizon Elasto-Plastic Object Manipulation with Diverse Tools** [20] https://openreview.net/pdf?id=69y5fzvaAT

Dumpling making is an extremely long-horizon tasks. It requires diverse tools with deformable objects.
Human approach? flour, dough, segment, cut, flatten, fill them with ingredients... very complex and long tasks.. Can robot achieve this?

Tool design inspired by real-world tools.

Robot achieves the task by sequentially going through the jobs of dough, knife, gripper, press, roller, circle cutter, pusher, skin spatula, filling spatula, hook, dumpling..

Skill-level planning: given current input and target, classifier network outputs which tools to use (like large roller, circle press, punch, knife..). Then, policy network determines precise actions.

Motion-level planning: key idea is a *tool-conditioned future prediction based on GNN.*

<Talk 4> Yunhai Han, Mandy Xie, Ye Zhao, Harish Ravichandar. **On the utility of koopman operator theory in learning dexterous manipulation skills.** [21] https://openreview.net/pdf?id=pw-OTIYrGa

Why dexterous manipulation? World is designed by and for humans. Multiple fingers increase control. Firm grasping for tool use.

Challenges:

- complex interdependent hand and object motions
- high dimensional state and action spaces
- highly nonlinear dynamics

Deep learning has shown progress, but the limitations are:

- computation burden
- hyperparameter tuning
- sensitivity to initialization
- inscrutable behaviors (blackbox)

Question: *Could we leverage well-understood structures to address the challenges in dexterous manipulation?* The answer is yes, and use Koopman operator.

What is Koopman operator? arbitrary nonlinear system is lifted to linear system. (transformation).

With lifting functions and training data, we can get an analytical solution. Koopman operator reduces the problems

- computation burden → least-squares optimization
- hyperparameter tuning → limited sensitivity to choice of lifting functions
- sensitivity to initialization → no sensitivity to analytical solution
- inscrutable behaviors (blackbox) → Linear dynamical system

KODex - Koopman operator-based dexterous manipulation

- the koopman operator is used to encode interdependent reference motions
- inverse dynamics controller is used to track the hand motion.

Q: what are the disadvantages?
A: Right now, it's not general yet; it's not evaluated on other benchmarks, other tasks, including locomotion.

< Talk 5 > Xinghao Zhu, JingHan Ke, Zhixuan Xu, Zhixin Sun, Bizhe Bai, Jun Lv, Qingtao Liu, Yuwei Zeng, Qi Ye, Cewu Lu, Masayoshi Tomizuka, Lin Shao, **Diff-LfD: Contact-aware Model-based Learning from Visual Demonstration for Robotic Manipulation via Differentiable Physics-based Simulation and Rendering** [22] https://openreview.net/pdf?id=DYPOvNot5F

Learning from demo: large-scale videos pf human manipulation; LfD reduces the need for reward engineering. extensive skill acquisition for foundation robot models

Challenge: RGB only, no object shape, camera pose; no task-specific rewards; limited videos for learning each task

Key Idea: *Contact-aware model-based learning from visual demo for robotic manipulation via differentiable physics-based simulation and rendering.*

Pose and shape estimation: differentiable rendering → differentiable mesh and texture rendering → reconstruction and pose estimation.
Employ diffusion models for generating unobserved angles.

Contact-aware manipulation policy; object 6d poses as subgoals → model-based policy based on graph searches.
Graph node is robot contact position and object 6d pose
Manipulation primitives as graph edges; contact point localization and contact wrench optimization.

Experiment: Comparison with NeRF
Diffusion model is necessary in that it completes occluded area
Compared to other planners: generates dexterous manipulation trajectories
Comparison to RL: planning + behavior cloning + RL fine-tuning requires less

training time than RL, generalizes better to new objects, and is deployable to real robots.

<Talk 6> Haonan Chen, Yilong Niu, Kaiwen Hong, Shuijing Liu, Yixuan Wang, Yunzhu Li, Katherine Driggs-Campbell. **Predicting Object Interactions with Behavior Primitives: An Application in Stowing Tasks.** [23] https://openreview.net/pdf?id=VH6WIPF4Sj

Modeling multi-object interactions is challenging. Heuristics alone cannot solve the task.

Learning forward dynamics via GNN. State $s = (O, E)$. Object vertex O consists of position, attribute; Gripper vertex consists of position, motion, attribute.

Pipeline: state, action → GNN → next state → MSE

Method works with a variety of objects and setups (with tiny object, with heavy and trapezoid objects; with bottle to be grasped, etc)

Summary: *dynamic model can enable robot to learn from minimal demo; stowing task for long-horizon manipulation; effectiveness and generalization.*

2.2 <Welcome Ceremony>

History of CoRL:
- 2017 Mountain view → 2018 Zurich → 2019 Osaka
→ 2020 Virtual, MIT → 2021 London
→ 2022 Auckland → 2023 Atlanta

Statistics: This year's CoRL is the largest CoRL

81% increase in in-person attendance from 2022;
912 total in person participants.
78% decrease in virtual attendance;
650 workshop attendees.

Papers:
498 submissions this year.
199 papers were accepted. (39.9%)
33 oral presentation (6.6%) + 166 posters

Review Process:
83 days with 60 area chairs 551 reviewers and 1658 reviews
79% papers received 4+ reviews; each paper received at least 3 reviews;
rebuttal improved paper quality (each paper received on average 6 reviewer responses); 203 papers improved score

Workshops:
32 workshop proposals - Strong emphasis on diversity and extra activities
Popular keywords: foundational models, representation learning, manipulation, agility;
11 accepted workshops (34%); total 225 workshop papers

2.3 < Oral Session 2: Reinforcement Learning >

<Talk1>
Franck Djeumou, Cyrus Neary, Ufuk Topcu, **How to Learn and Generalize From Three Minutes of Data: Physics-Constrained and Uncertainty-Aware Neural Stochastic Differential Equations** [24]
https://openreview.net/pdf?id=770xKAHeFS

Learning to predict system dynamics
Objective: *learn to a control-oriented dynamics model from offline trajectory data*

Questions:
- How can we leverage physics knowledge in data-driven models?
- How to build useful approximations of model uncertainty?
- how to control real systems using limited data?

Physics-aware neural stochastic differential equations with uncertainty estimates:

- physics-informed structure + state s and control input u goes into the physics-informed neural ODE, which outputs physics-aware drift term,
- training dataset $\rightarrow$ learned distance-aware uncertainty estimate
- together it solves SDE

Training of the distance-aware uncertainty term Objective: parameterize and train a diffusion term with the following properties:
1. efficient evaluation at inference time,
2. low diffusion near the dataset
3. diffusion increases with the distance to the nearest training datapoint
4. diffusion saturates at a maximum value

Neural SDEs for model-based RL: model-free performance with a fraction of the system interactions; using dynamics dataset, train neural SDE $\rightarrow$ control policy

<Talk2>
Yunhai Feng, Nicklas Hansen, Ziyan Xiong, Chandramouli Rajagopalan, Xiaolong Wang, **Finetuning Offline World Models in the Real World** [25]
https://openreview.net/pdf?id=JkFeyEC6VXV

Current Challenges in robot learning:

- Interactions are costly
- Random exploration is inefficient

Question: how to leverage existing data for learning new tasks quickly?
Pipeline: Offline pretraining of existing data → world model (TD-MPC) → online fine-tuning using the world model for an unseen task

Planning with the world model faces model uncertainty issues
→ use Q-ensemble → regularize the reward terms

<Talk3>
Zixuan Wu, Sean Charles Ye, Byeolyi Han, Matthew Gombolay, **Hijacking Robot Teams Through Adversarial Communication** [26]
https://openreview.net/pdf?id=bIvIUNH9VQ

Multi-agent system with communication; some perturbation of communication messages. In the work, learn to hijack the coordination behaviors.

Communicative multi-agent problem setting. Each agent tries to catch prey; communication (binary data). Team collectively maximizes the reward. Adversarial policy: intercepts the intended message from agents; send back altered message.

Previous works:
1. assume online poisoning RL training where the adversary's policy can explore how altered messages change the agent actions in the environment
2. adversary's policy can learn from the ground truth reward signal.
→ seek to remove these assumption.

Contribution:
1. learn without environment interactions by learning a surrogate policy
2. learn without ground truth
3. design a differentiable representation for adversarial bit flipping
→ Learn a model of agent behaviors (surrogate policy) through behavior cloning
Use the surrogate policy for reward estimation.

Differentiable adversary's policy: direct mode and flipping mode

Limitations:

- attacking interpretable communication vectors
- polymorphic encryption
- low swap devices

Summary:

- trained full black-box by learning a surrogate policy
- train without access to gt rewards by inferring the reward from surrogate policies
- differentiable representation for adversarial bit flipping

<Talk4>
Robert Gieselmann, Florian T. Pokorny. **Expansive Latent Planning for Sparse Reward Offline Reinforcement Learning** [27]
https://openreview.net/pdf?id=xQx1O7WXSA

Value-guided expansive latent planning

Problem setting: sparse reward visual off-line RL method:
- Contrastive state representation
- Guided tree search in latent space
- Local Q $\rightarrow$ temporal distance
- Global Q $\rightarrow$ cost-to-go

Reject n-step rollout if
- Not reachable (with certain number of steps)
- Model uncertainty is too high
- New node is not novel enough

Guided node and action sampling
Sparsity and value heuristics improve planner efficiency

<Talk5>
Wenxuan Zhou, Bowen Jiang, Fan Yang, Chris Paxton, David Held. **HAC-Man: learning hybrid actor-critic maps for 6d non-prehensile manipulation** [28] https://openreview.net/pdf?id=fa7FzDjhzs9

non-prehensile manipulation: manipulating object without grasping; pushing, flipping, toppling, sliding...

Task: 6D object pose alignment with non-prehensile skills. 6D object pose alignment with point cloud observations. Challenges:
- requires sequential contact decision
- requires dynamic interaction with the objects
- reasoning about geometry

Standard action space in manipulation: robot-centric; extremely inefficient!
Most of the actions are free-space motions. actions are robot-centric instead of object-centric. Difficult to generalize.

Instead, in this work, define an object-centric action space.

- Contact Location selected from the object point cloud.
- Motion parameter (continuous) from delta EE movement.

Benefits: 1. temporally abstracted 2. spatially grounded.

Spatial action map:
- densely connecting actions to visual input using segmentation-style networks
- discrete-continuous action space
- learned with RL; no expert demonstrations and just with sequential decision.

Idea: obtain per-point contact critic
Critic map: embedding the choice of contact location in a segmentation-style critic.

3 Day 3 (Nov 8th)

3.1 < Oral Session 3: Driving, Navigation, Locomotion >

<Talk 1> Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, Sergey Levine. **ViNT: A Foundation Model for Visual Navigation** [29] https://openreview.net/pdf?id=-K7-1WvKO3F

Simulation training = model environment in sim; generalize to zero-shot
vs.
real-world training = on-robot data collection; difficult to generalize zero-shot

Why is real-world robot learning hard?
- The real world is hard to simulate
- Collecting data is a high barrier for entry
- may be infeasible for many system
- learned policies can be fragile

How AI used to work; all of different models (segmentation, classification, captioning, visual QA, sentiment,..) → Now: giant self-supervised pretrained model (foundation model) → prompting,..

How mobile robots should learn in the future? Train a giant robot foundation model → finetuning → tasks

Navigation Foundation Model is composed of algorithm, data, model
- Algorithm: self-supervised objective
- Data: cross-embodiment data
- Model: Transformer backbone; prompt-tuning and adaptation

ViNT: Visual Navigation Transformer (30M trainable parameters

Exploration with ViNT; train a generative model to suggest subgoal candidates; score subgoals using a goal-directed heuristics

Interesting part: transformer prompt-tuning?

<Talk 2>

Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher G Atkeson, Sören Schwertfeger, Chelsea Finn, Hang Zhao. **Robot Parkour Learning** [30] https://openreview.net/pdf?id=uo937r5eTE

Why parkour? Isn't walking enough? The answer is 'not enough'.
→ Parkour skills are necessary for the open world. (Leap, Climb, Crawl, Tilt..)

Recent success on locomotion rely on heavy reward engineering, and customized for one task: walking. Not scalable to the parkour skills.
→ RL with soft dynamics constraints, inspired by direct collocation methods.
→ softly penalize penetration;
reward -= (# of penetration points + penetration depth)

Next step: RL fine-tuning with physics
→ Then, One parkour policy (egocentric depth and proprioception as inputs)

<Talk 3> Kevin Huang, Rwik Rana, Alexander Spitzer, Guanya Shi, Byron Boots. **DATT: Deep Adaptive Trajectory Tracking for Quadrotor Control**[30] https://openreview.net/pdf?id=XEw-cnNsr6

Agile drone flight; problem setting - minimize the distance from location of drone to the reference point, at each time

Need a framework that (1) tracks arbitrary trajectories (2) adapts online to unknown conditions. (3) leverage domain knowledge and inductive biases (4) Little/no fine tuning (5) Fast

Previous approaches:

- classical nonlinear controller (requires bounded higher order derivatives of reference position
- Model Predictive Control (MPC), reliant on accuracy of dynamics model and optimality of OCP solver
- RL, requires relatively minimal priors about tasks. but lack of inductive biases can cause sim2real issues

Their approach: condition on the embedded reference trajectory → feedforward model; disturbance estimation residual; modular adaptation strategy can easily

be changed.

Limitations: High variance in training; everything must be in body frame; learning curriculum is important; stability/safety guarantees?

<Talk 4> Xiyang Wu, Rohan Chandra, Tianrui Guan, Amrit Bedi, Dinesh Manocha. **Intent-Aware Planning in Heterogeneous Traffic via Distributed Multi-Agent Reinforcement Learning** [31]
https://openreview.net/pdf?id=EvuAJ0wD98

Motivation: heterogeneous traffic.
It is challenging to model the interaction between heterogeneous agents.
Real-world datasets are insufficient (most datasets are homogeneous traffic)
Improve MARL(multi-agent RL) framework

Contribution: MARL approaches for joint trajectory and intent prediction

Method:

- ego vehicle: traffic state at time t - (1) current observation (2) behavioral incentive (3) instant incentive
- behavioral incentive inference; use historical observation sequence and previous behavior incentive inferences to generate new behavior incentive inference.
- instant incentive inference.
- controller

Summary: iPLAN is first MARL algorithm embedded with a trajectory and intent prediction, in dense traffic situations.

<Talk 5> Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, Baishakhi Ray. **CTG++: Language-Guided Traffic Simulation via Scene-Level Diffusion**. [32]
https://openreview.net/pdf?id=nKWQnYkkwX

Why traffic simulation? (1) road test for AVs can incur traffic accidents
(2) road test can miss corner cases

Desirable properties of traffic simulation model →
context; traffic trajectories leading to collision; realism; controllability; user-friendly interface.

heuristic-based models is controllable but not realistic; data-driven traffic models are not controllable; CTG is controllable, realistic, but not user-friendly; CTG++ solves these issues.

Overview of CTG++:
predefined APIs and Examples → GPT4 → guidance loss → CTG++ module

- Motivation for language guidance → no paired text-to-traffic data; GPT4 can translate text to code; diffusion can be guided by loss function
- Approach → leverage GPT4 to convert text to loss function in code to guide diffusion.

Scene-level diffusion: spatial-temporal transformer backbone.
Motivation: we want to model agents jointly.
Approach: propose a spatial temporal transformer to capture agent interactions.

Spatial Attention
- Motivation: directly using scene-centric coordinates or agent-centric coordinates have limitations
- Approach: agent-centric coordinate+ spatial attention incorporate relative information of vehicle pairs

3.2 <Oral Session 4: Large Language Models >

<Talk 1> Adam Rashid, Satvik Sharma, Chung Min Kim, Justin Kerr, Lawrence Yunliang Chen, Angjoo Kanazawa, Ken Goldberg. **Language Embedded Radiance Fields for Zero-Shot Task-Oriented Grasping** [33]
https://openreview.net/pdf?id=k-Fg8JDQmc

Task-oriented grasping; must pick up a particular part of the object. Can we scale object part datasets?

What about unlabeled parts? → natural language... but in 3D?

Language in 3D? objects? but what about sub-parts?

Closely related work: Language Embedded Radiance Field (LERF) [34]
abstract queries, text reading, **multi-scale semantics**, uncommon objects

(1) reconstruction of scenes. (2) extract point cloud for inputs.

Why don't naive queries work? CLIP Bag-of-words. rose and stem, respectively, works well, but 'rose stem' doesn't.

grasp selection: geometric candidates → reweight them semantically

Limitations:

- computationally heavy
- static scene representation

- pre-scripted primitive actions
- unreliable for existing queries

<Talk 2> Wenhao Yu, Nimrod Gileadi, Chuyuan Fu et al. **Language to rewards for robotic skill synthesis** [35]
https://openreview.net/pdf/23fe82729ecd694d697caf92d13de1e08a63bbc0.pdf

LLMs have great potential in high-level reasoning, but do not handle low-level actions directly → it is not a fundamental limitation of LLMs.

Language-to-reward: good interface? Reward function is very expressive. LLM can understand via code; rich literature on reward-to-action.

LLM Reward translator → motion descriptor → description of the desired motion in natural language → reward coder → reward code

Takeaway: in-context prompting and zero-shot

Low-level controller: Model predictive control. Interactively synthesizing low-level actions to optimize cost/reward.

Concurrent work: Voxposer [36], Eureka [37]

< Talk 3>
Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, Anirudha Majumdar. **Robots that ask for help: uncertainty alignment for LLM planners** [38]
https://openreview.net/pdf?id=4ZK8ODNyFXx

Robots should know when they don't know and ask for help. LLMs can generate long-horizon plans and write code. But they often don't know when they don't know. Uncertainty estimate can be biased.

Would like to quantify the uncertainty of LLM Planner.
1. Ask enough help to ensure a statistically guaranteed level of success
2. Ask for minimal amount of help
KnowNo: Know When you don't know

Perplexity of samples? Instead, planning as multiple-choice Q&A.
Generate possible plans → query next-token probabilities

Calibrate LLM planner with conformal prediction.
LLM outputs prediction set; statistical guarantee > user-specified probability.

How? collect calibration dataset → confidence score

Conformal prediction minimizes prediction set size.

Extension to multi-step planning setting.

<Talk 4> Jesse Zhang, Jiahui Zhang, Karl Pertsch, Ziyi Liu, Xiang Ren, Minsuk Chang, Shao-Hua Sun, Joseph J Lim.
Bootstrap Your Own Skills: Learning to Solve New Tasks with Large Language Model Guidance [39] https://openreview.net/pdf?id=a0mFRgadGO

Want: solve complex tasks in new settings
Previous approach's problem: dense task-specific supervision

Through practice:
- master skills in new situations
- compose skills into new, complex behaviors
solve complex tasks in new settings

BOSS: Autonomous practice with LLM-Guidance
Prior work like SayCan doesn't adapt to new environments.
BOSS learns new skills through practice.
→ closed-loop; adapts to target environments; no test time LLM

Loop of Mastering and Composing Skills with the LLM guidance in the middle.
Repeats this process iteratively.

<Talk 5> Krishan Rana, Jad Abou-Chakra, Sourav Garg, Jesse Haviland, Ian Reid, Niko Suenderhauf. **SayPlan: Grounding Large Language Models using 3D Scene Graphs for Scalable Task Planning** [40]
https://openreview.net/pdf?id=wMpOMO0Ss7a

Goal: have a robot that can complete many tasks, across large-scale environments.
→ Generation of sequences of high-level actions to achieve a goal.
Classical approach is not scalable.
Recently, we use LLMs. But LLM should be grounded to the environments.
Currently, limited to small scale table-top or single-room environment.

How to ground LLM to larger environments?
Answer is to use 3D scene graphs! [41]
Hierarchical abstraction of an environment captures semantics, affordance, predicates, state, attributes.

Naive approach doesn't scale with environment size and planning horizon.

Answer: use 3D scene graph along with LLM!

Exploit the hierarchical nature of the scene graph. Compress it. Semantic

Search by identifying a task relevant sub-graph suitable for planning.

Planning: generate an executable task plan for a mobile manipulator robot.

- Path planner connects high-level LLM-generate navigational goals
- Scene graph simulator feedback allows the LLM to refine the generated plan.

<Talk 6>
Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, Li Fei-Fei. **VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models** [36]
https://openreview.net/pdf?id=9_8LF30mOC

Motivation: What do we need to achieve a task goal?
Task semantics + physical behaviors.

Aim: No robot training data required; synthesize 6 DoF closed-loop actions

VoxPoser: VLM/LLM generates code
→ code-generated 3D value map
→ motion planning

Also generates rotation map, gripper map, velocity map

Code-generated 3d value map:

- vision: grounded in 3d robot perception;
- language: reasoning capabilities of LLMs; natural interface with human users
- action: directly enables 6 DoF closed-loop actions; no robot data required

What about tasks involving more intricacies of contact?
random exploration is not apt
→ voxposer can generate priors
→ exploration around prior

Limitations
- relies on verbal interaction between LLM and VLM. this can be bottleneck.
- only consider end-effect in motion planning
- insufficient for tasks requiring fine-grained contact modeling

3.3 Early Career Keynotes

<Invited Talk 1> Chelsea Finn, **Amending Moravec's Paradox**

Traditional AI success stories: solving complex problems, Go, Atari..
But truly complex problems? basic manipulation skills are beyond the scope of current methods

Moravec's paradox: *things that are intuitive and easy for humans are often hard for machines*

ChatGPT - holding a dialog; Segment Anything - low-level perception; creative imagery; but we still lack the basic motor skills as babies have

What is hard and easy for machines? What is more difficult for the robot? Putting on shoes? or opening a pot lid? People think putting on shoes is more difficult.

What is hard for robots? Traditionally
- coordinating many degrees of freedom
- many timesteps
- precise control
- situations that are hard to model/simulate

In fact, putting on shoes or prep tape tasks are being tackled.

Opening the pot lid? Maybe it is challenging? Contexts vary a lot; scenes vary a lot; *what makes the task difficult is not the task itself, but the context as well*

Amending the Moravec's Paradox; ***things are hard for machines if it's hard to get data for those things.***

We need to ask what was in the training data. Another thing that is hard: broad robot generalization.
→ 1. Broaden the data distribution 2. generalize beyond the training data

Can we use broad data across institutions, robot embodiments? 1. Put all of the data in a common format → Open X-Embodiment dataset
2. Train a large model

Key Takeaway:
- *One policy can control multiple robots at multiple institutions.*
- *Training on cross-embodiment data improves over training on in-distribution data.*

What about generalization beyond the training data?

Idea: can robots adapt on-the-fly during deployment?
→ motivation for single-life reinforcement learning; given prior data in train envs, agent has one life to autonomously complete task in novel, OOD scenario.

Idea: *Can robots adapt both in the space of parameters and in the space of*

behaviors?

Arguments:
- What we should not do:
propose fancy method for getting robot to do complex task
- What we should do? *enable scalable data and methods for getting robot do many complex tasks in many scenarios with many objects*

<Invited Talk 2> Jemin Hwangbo, **Building dynamic legged machine**

Legged locomotion: this talk's focus is on dynamics, estimation, adaptation, and hardware design

Progress of legged robot control research (chronological): heuristic rules → zmp-based control → dynamics-based control → trajectory optimization → deep RL

Rigid-body dynamics: raisim (fast rigid-body; fast RL implementation)

The reality gap; when policy was transferred to the real robot, it totally failed.

Actuator dynamics: actuation is so complex that we cannot assume that this is ideal. Yet, complex actuator dynamics can be learned. → More successful real-robot results

Terrain Dynamics: challenge is at deformable terrain. General deformable terrain solution (Chrono).
Deformable train; models for compaction, friction, resistance → simulating diverse terrains using the proposed model → impressive results of legged robot running on the sand.

<Invited Talk 3>
Yuke Zhu, **Pathway to Generalist Robots: Scaling Law, Data Flywheel, and Humanlike Embodiment**

Specialist → Generalist → Specializing Generalist
On our pathway to specialized generalist robot, we need generalist robot models

In NLP,
Semantic parsing, sentiment analysis, text summarization, information extraction → Large Language Model → creative writing, travel planning, source code auto-completion

Recipe for building generalist robot models
- scaling law: powerful robot learning models that scale with data and compute
- data flywheel: new mechanism to collect massive training data
- human-like embodiment: humanoid robot platform

Key Idea: Skills as APIs and Scaling Law; Library of skill APIs, invoking parameterized skills. Data-efficient imitation learning for sensorimotor skills.

Massively multi-task robot learning for model scaling. VIMA. generalist robot agent for multi-task learning and zero-shot generalization. Transformer (encoder-decoder) architecture that encodes multimodal prompts with a frozen language model; object-centric representation

Data Flywheel:
wider deployments → more training data → better models → more capable robots → wider deployments... cycle

- How can we ensure trustworthy deployment?
- How can robots learn efficiently?

Human-like Embodiment: Draco3 Humanoid robot development

<Invited Talk 4>
Karol Hausman, **Bitter Lessons & Sweet Future in Robot Learning**

Bitter Lessons - famous essay by Richard Sutton
computation are ultimately the most effective..

New bitter lesson?
1. robotics will need to understand the world through data
2. trend:?
3. work on robot learning methods that leverage that trend

What is the trend? New trend?
Foundation Models.
- Interest outside of robotics; scale with data and compute (1st bitter lesson); get better at understanding the world

New bitter lesson.. Imagine community is looking back now 70 years later..
"The biggest lesson that can be read from 70 years of robotics research is that general methods that leverage foundation models are ultimately the most effective"..?

Bitter lesson aspect of PaLM-SayCan:
- robotics performance scales with better LLMs!
- chain-of-thought prompting
- solves all kinds of queries

The bottleneck is still on the actions. Can we apply our new bitter lesson to the actions?

RT-1: robotics transformer 1
→ now, RT-2, absorbed much more data, RT-1,..

Representing actions in VLMs.
RT1: positional changes..$(\Delta x, \Delta y, \Delta z)$, rotational changes..
RT2: Just stringified it! vision-language-action (VLA) model

User/agent interfaces:
- RT-Sketch https://rt-sketch.github.io/
- RT-Trajectory https://rt-trajectory.github.io/

<Talk 5>
Stefan Leutenegger, **From Multi-Sensor SLAM to Spatial AI**

Example: drone-based mapping for digital forestry.

Classical, modular mobile robotics: perception → state estimation and mapping → planning → feedback control
natural introspection and interaction: shared representations (map, knowledge, signals)

radically shifting to learning-based mobile robotics;
- self-emergent, not handcrafted behaviors and representations
- blackbox, no guarantees; very difficult to introspect
- not data efficient enough

Hybrids advocated for in this talk:
natural introspection, interaction + efficient training of modules + end-to-end training and fine-tuning possible + learned modules/policies running in parallel as residuals

Levels in Spatial AI: dense mapping → semantics maps → object-level and dynamic maps

3D occupancy mapping; map visualization
Summary:

- contributions SLAM and Spatial AI; from robot motion to dense geometry, semantics, objects, and human motion
- Deep Learning proves helpful for inference of semantics, objects, people; data association over time; completion of geometry beyond the visible
- the modular architecture allows for integration with robot planning and control

<Talk 6> Shuran Song, **What I wish I had for robot learning**

Pushing the boundaries of manipulation..*by carefully designing action primitives*

Well-designed action primitives: simplifies learning, but requires engineering effort and insights, gradually becomes bottleneck

Changing the workflow: design actions for learning → learn complex skills directly from data → Diffusion policy

Diffusion policy: generative model for robot actions.

If we can collect the data, the robot can do the task...but is it really easy? no
→ Caveat: data is not any type of data
→ a lot of effort and intelligence to get the right data

secret recipes for collecting data:
- train yourself
- cover the task space
- anticipate the robot failures
- protect the environment

What I wish I had for robot learning:

- Data! Scalable
- Data! Reusable
- Data! Robot-Complete

Simulation: it's reusable. what's the bottleneck? accuracy? it can be improved with more compute; setup cost for new tasks! often overlooked.

Robot-complete data: embodied actions + embodied observations + non-zero success rate → easy to scale for one task but hard for many tasks

Using LLMs → scaling up and distilling down
- Scaling up: Use LLMs to guide the exploration.
- Distilling Down: visuomotor policy that
— infers actions from raw sensory input, without simulation state
— does continuous improvements with more experience, without fine-tuning the language model itself

Real-world data: What is the bottleneck? need human demonstrations? it's not a true bottleneck. Rather, lack of an intuitive and standardized interface for demonstrations is a bottleneck.

4 Day 4 (Nov 9th)

4.1 < Oral Session 5: Manipulation 2>

<Talk 1> Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, Anima Anandkumar. **MimicPlay: long-horizon imitation learning by watching human play.**[42]
https://openreview.net/pdf?id=hRZ1YjDZmTo

How can robot tackle long-horizon manipulation tasks?
- Task and motion planning: inflexible to new setups
- Planning with LLMs: flexible to novel tasks and domains; but lack of grounding and low-level control

Key factors for long-horizon manipulation:
(1) real-time planning and control (2) closed-loop reaction

- robot demo → small domain gap, but high cost
- web data → low cost, but large domain gap
We want something in between?

Key features of human play data:
1. rich 3d trajectory plans 2. diverse task goal compositions

Human-play data is faster to collect; it is also large-scale. Robot teleoperation is slow to collect and small-scale.

MimicPlay: high-level planner learned with human play data: low-level control learned from robot data

In high-level: current image and goal image → latent plan → 3D hand trajectory reconstruction; for unpaired robot image, use KL Loss

After training high-level latent model, freeze the model, and use it for low-level control

Limitation: requires third-view camera system

<Talk 2> Shiqi Liu, Mengdi Xu, Peide Huang, Xilun Zhang, Yongkang Liu, Kentaro Oguchi, Ding Zhao. **Continual Vision-based Reinforcement Learning with Group Symmetries** [43] https://openreview.net/pdf?id=flyQ0v8cgC

Continual learning in human? solving streaming tasks in the real life.

Solving streaming tasks in the robot lifespan development!
Goals: learn new tasks quickly & remember learned tasks

Humans are good at handling different configurations; learning skills to solve **equivariant tasks**

In the robot world, equivariant task configurations require equivariant actions.

COVERS: solve equivariant tasks in continual RL. Detect delineations between task groups + one equivariant policy for each group.

Pipeline: after collecting online interaction data → unsupervised data clustering → if distance (between the current rollout data and data for each policy) is below the threshold, append to the data buffer and update policy; otherwise, new policy

Equivariant feature extractor has equivariant convolutional net and linear networks; yields invariant value network and equivariant policy net.

<Talk 3> Yulong Li, Andy Zeng, Shuran Song. **Rearrangement Planning for General Part Assembly** https://openreview.net/pdf?id=pLCQkMojXI

Part-assembly: previous assumptions
(1) fixed targets
(2) fixed categories
(3) fixed parts
(4) precise object models

But should work with non-exact parts, novel target category, visual perception

General Party Assembly → rearrangement planning (different from other planning like motion planning, task planning)
Rather than canonical pose and exact parts → random pose and non-exact parts

Idea 1: optimize for Chamfer distance?
→ local minimum problem, worse on non-exact.
Idea 2: regressing part poses?
→ ambiguities, generalization problem

⇒ Approach: general part assembly transformer

Segmentation prediction → pose estimation → assembly prediction
PointNet and PointNet+Max pools → GPAT layer → MLP

GPAT layer? to address ambiguities, local-to-global attention: consistency of segmentation; Use multi-scale attention

<Talk 4> William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, Phillip Isola. **Distilled Feature Fields Enable Few-Shot Language-Guided Manipulation** [44] https://openreview.net/pdf?id=Rb0nGIt_kh5

Scene representation: generalization to unseen categories
What are in the scene? where they are? we both care about semantics and geometry.

Let's marry 2D foundation models and NeRFs (highly fined-grained 3D geometry) → ***3D neural feature fields***!

Their Method: F3RM

(1) producing feature fields
multi-view images (50 images)
Difference between F3RM and NeRF
: F3RM extract dense features by passing these images to CLIP (VLM)

(2) how to use it for manipulation:
language-guided manipulation with CLIP Feature fields
1. retrieve relevant demos - text features from clip
→ 2. language-guided pose optimization (maximize similarity)

Limitations:

- requires dense views (50 images)
- optimization per scenes (computation)
- CLIP behaving like a bag of words

<Talk 5> Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, Xiaolong Wang. **GNFactor: Multi-Task Real Robot Learning with Generalizable Neural Feature Fields** [45] https://openreview.net/pdf?id=b1tl3aOt2R2

How to learn 3D feature fields from 2D foundation models?
→ Use the idea of NeRF

Can we achieve multi-scene manipulation with few demos? one single policy for two different kitchens? more interestingly, one single-policy for multiple tasks?

Distill 2D feature map into 3D space with generalizable NeRFs.

Learn a unified 3d and semantic representation; dense and rich semantics from stable diffusion feature.
: diffusion feature → 3d space → generalizable neural feature fields.

GNFactor: Learning deep volumetric representation → efficient in encoding dynamic scenes.
Deep 3d volumetric representation → render diffusion feature + action prediction; doing so, it learn multi-task generalizable policy jointly

<Talk 6> Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, Dieter Fox. **RVT: Robotic View Transformer for 3D Object Manipulation** [46] https://openreview.net/pdf?id=0hPkttoGAf

So far, popular representations for 3d manipulation:

- (1) View-based representation (scalable, but cannot handle 3D and don't work well)
- (2) Explicit 3d representations (not scalable, perform well)

→ RVT: robotic view transformer. Represent the scene via set of orthogonal (virtual) images; more scalable and performs well.

RVT pipeline: scene → input RGBD images → reconstructed point cloud → virtual images (virtual view representation; orthographic angles) → predicted 2D heatmap → reverse projection operation → predicted 3D heatmap → gripper location; gripper rotation and state → execution

Key Ideas:

- Re-rendering with virtual orthogonal camera shows better performance than using input views, probably because of inductive bias and 3D augmentations
- (Contrary to perspective projection) in orthographic projection, object sizes remain same independent of distance from camera.

4.2 Sponsor Talk: Google DeepMind

<Talk> **Open X-Embodiment: Present and Future**
https://robotics-transformer-x.github.io/

Context: the tales of large datasets
In CV and NLP:

- Large scale datasets unlock new capabilities in CV and NLP research
- data sources are naturally occurring

But is it equivalent for robotics research?

Addressing data scarcity as a community:
Previously: BridgeV2, RoboTurk, QT-OPT, RoboNet... however they end up being isolated, small datasets.

Diversity on the type of tasks:
...long-horizon tasks, soft objects, dexterity, tactile sensing...

Many Embodiments: mostly robot arms (stretch, xArm, Franka, google robot)
Many Scenes: robot lab, kitchen, outdoors, tabletop
Many Skill: pouring, unfolding, cable routing; screwing

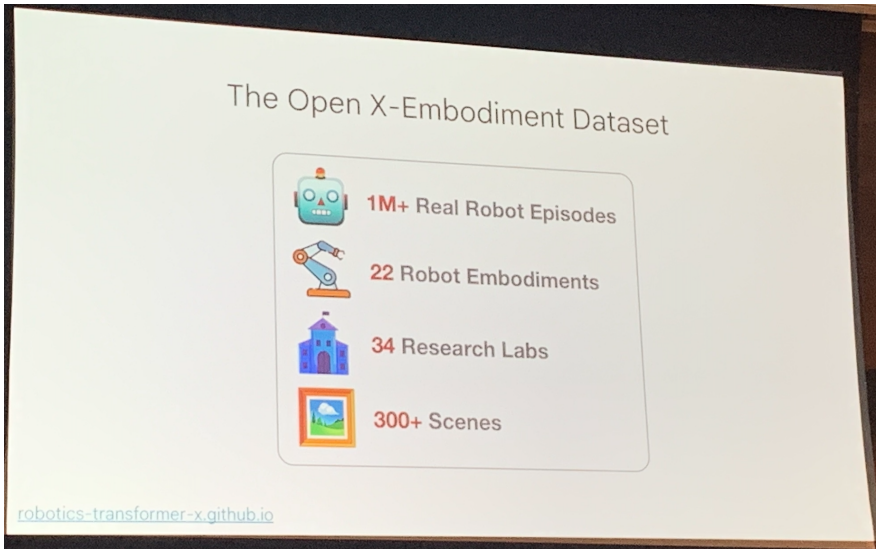

Figure 4: Open X-Embodiment Dataset Overview

Common framework for datasets (specific common format)

Diverse data → new challenges

Specialist models trained on specialized datasets (dexterity, long-horizon + ... many other tasks)
Key Research Questions:
Generalist models better than specialist models?
Is all this work worth our time?
High-capacity architectures; minimal architecture modifications (impacts of data scaling)

Current modeling assumptions: single arm
2-fingers, mostly parallel yaw
still interesting diversity!

Just RT-1 and RT-2 trained on X-embodiment datasets. Images and a text instruction as input; outputs discretized end-effector actions.

Three axes of emergent skills evaluations:
(1) Object-relative position understanding; (2) absolute position understanding; (3) preposition modulates low-level motion

Future Direction

Dirty laundry: open x-embodiment dataset is diverse, but not truly *in-the-wild*

Need more:
- diverse environments
- robot embodiments (mobile manipulation, humanoids, drones..)
- realistic tasks
- simulation data

RT-X model is not 'flexible'. Future models should support:
- flexible input observations (camera, depth, proprioception...)
- flexible task specification (goal, language...)
- flexible action spaces (EEF, join space, base movement, bimanual..)
- efficient fine-tuning

But is diverse data all we need?.. what about evaluations?

Many labs have a small amount of data. By pooling, we get a large and diverse data. → Many labs have a few benchmarking tasks. By pooling, we get a large and diverse benchmark.

Internet of robotics data?
- no lost byte
- responsible data practices?
– lineage tracking, data version control at scale
- contribute data if you have new hardware!

An API for generalist robot models?
RT-2-X is the first model to work for multiple platforms, institutions, and tasks. Can we test it in even more situations?
Prompt, image → RT-2-X API (in the cloud) → actions

References

[1] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in reinforcement learning. In Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 5402–5415, 2021.

[2] Mirco Mutti, Riccardo De Santi, and Marcello Restelli. The importance of non-markovianity in maximum state entropy exploration. In International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore,

Maryland, USA, volume 162 of Proceedings of Machine Learning Research, pages 16223–16239. PMLR, 2022.

[3] Ev Zisselman, Itai Lavie, Daniel Soudry, and Aviv Tamar. Explore to generalize in zero-shot RL. CoRR, abs/2306.03072, 2023.

[4] Xiaohan Zhang, Yifeng Zhu, Yan Ding, Yuqian Jiang, Yuke Zhu, Peter Stone, and Shiqi Zhang. Symbolic state space optimization for long horizon mobile manipulation planning. CoRR, abs/2307.11889, 2023.

[5] Alex Licari LaGrassa and Oliver Kroemer. Learning model preconditions for planning with multiple models. In Conference on Robot Learning, 8-11 November 2021, London, UK, volume 164 of Proceedings of Machine Learning Research, pages 491–500. PMLR, 2021.

[6] Ge Yang, Amy Zhang, Ari S. Morcos, Joelle Pineau, Pieter Abbeel, and Roberto Calandra. Plan2vec: Unsupervised representation learning by latent plans. In Alexandre M. Bayen, Ali Jadbabaie, George J. Pappas, Pablo A. Parrilo, Benjamin Recht, Claire J. Tomlin, and Melanie N. Zeilinger, editors, Proceedings of the 2nd Annual Conference on Learning for Dynamics and Control, L4DC 2020, Online Event, Berkeley, CA, USA, 11-12 June 2020, volume 120 of Proceedings of Machine Learning Research, pages 935–946. PMLR, 2020.

[7] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: towards universal visual reward and representation via value-implicit pre-training. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023.

[8] Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: language-image representations and rewards for robotic control. In International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, volume 202 of Proceedings of Machine Learning Research, pages 23301–23320. PMLR, 2023.

[9] Archit Sharma, Kelvin Xu, Nikhil Sardana, Abhishek Gupta, Karol Hausman, Sergey Levine, and Chelsea Finn. Autonomous reinforcement learning: Formalism and benchmarking. In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net, 2022.

[10] Archit Sharma, Ahmed M. Ahmed, Rehaan Ahmad, and Chelsea Finn. Self-improving robots: End-to-end autonomous visuomotor reinforcement learning. CoRR, abs/2303.01488, 2023.

[11] Jingyun Yang, Max Sobol Mark, Brandon Vu, Archit Sharma, Jeannette Bohg, and Chelsea Finn. Robot fine-tuning made easy: Pre-training rewards and policies for autonomous real-world reinforcement learning. CoRR, abs/2310.15145, 2023.

[12] Annie S. Chen, Archit Sharma, Sergey Levine, and Chelsea Finn. You only live once: Single-life reinforcement learning. In NeurIPS, 2022.

[13] Kristen Grauman, Andrew Westbury, Eugene Byrne, et al. Ego4d: Around the world in 3, 000 hours of egocentric video. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pages 18973–18990. IEEE, 2022.

[14] Kumar Ashutosh, Rohit Girdhar, Lorenzo Torresani, and Kristen Grauman. Hiervl: Learning hierarchical video-language embeddings. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023, pages 23066–23078. IEEE, 2023.

[15] Zihui Xue and Kristen Grauman. Learning fine-grained view-invariant representations from unpaired ego-exo videos via temporal alignment. CoRR, abs/2306.05526, 2023.

[16] Brian Ichter, Anthony Brohan, Yevgen Chebotar, et al. Do as I can, not as I say: Grounding language in robotic affordances. In Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand, volume 205 of Proceedings of Machine Learning Research, pages 287–318. PMLR, 2022.

[17] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, et al. Palm-e: An embodied multimodal language model. In International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, volume 202 of Proceedings of Machine Learning Research, pages 8469–8488. PMLR, 2023.

[18] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? IEEE Robotics Autom. Lett., 5(4):6670–6677, 2020.

[19] Jennifer Grannen, Yilin Wu, Brandon Vu, and Dorsa Sadigh. Stabilize to act: Learning to coordinate for bimanual manipulation. CoRR, abs/2309.01087, 2023.

[20] Haochen Shi, Huazhe Xu, Samuel Clarke, Yunzhu Li, and Jiajun Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. CoRR, abs/2306.14447, 2023.

[21] Yunhai Han, Mandy Xie, Ye Zhao, and Harish Ravichandar. On the utility of koopman operator theory in learning dexterous manipulation skills. CoRR, abs/2303.13446, 2023.

[22] Xinghao Zhu, JingHan Ke, Zhixuan Xu, Zhixin Sun, Bizhe Bai, Jun Lv, Qingtao Liu, Yuwei Zeng, Qi Ye, Cewu Lu, et al. Diff-lfd: Contact-aware

model-based learning from visual demonstration for robotic manipulation via differentiable physics-based simulation and rendering. In 7th Annual Conference on Robot Learning, 2023.

[23] Haonan Chen, Yilong Niu, Kaiwen Hong, Shuijing Liu, Yixuan Wang, Yunzhu Li, and Katherine Rose Driggs-Campbell. Predicting object interactions with behavior primitives: An application in stowing tasks. CoRR, abs/2309.16873, 2023.

[24] Franck Djeumou, Cyrus Neary, and Ufuk Topcu. How to learn and generalize from three minutes of data: Physics-constrained and uncertainty-aware neural stochastic differential equations. arXiv preprint arXiv:2306.06335, 2023.

[25] Yunhai Feng, Nicklas Hansen, Ziyan Xiong, Chandramouli Rajagopalan, and Xiaolong Wang. Finetuning offline world models in the real world. arXiv preprint arXiv:2310.16029, 2023.

[26] Zixuan Wu, Sean Charles Ye, Byeolyi Han, and Matthew Gombolay. Hijacking robot teams through adversarial communication. In 7th Annual Conference on Robot Learning, 2023.

[27] Robert Gieselmann and Florian T Pokorny. Expansive latent planning for sparse reward offline reinforcement learning. In 7th Annual Conference on Robot Learning, 2023.

[28] Wenxuan Zhou, Bowen Jiang, Fan Yang, Chris Paxton, and David Held. Learning hybrid actor-critic maps for 6d non-prehensile manipulation. arXiv preprint arXiv:2305.03942, 2023.

[29] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. arXiv preprint arXiv:2306.14846, 2023.

[30] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Soeren Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. arXiv preprint arXiv:2309.05665, 2023.

[31] Xiyang Wu, Rohan Chandra, Tianrui Guan, Amrit Bedi, and Dinesh Manocha. Intent-aware planning in heterogeneous traffic via distributed multi-agent reinforcement learning. In 7th Annual Conference on Robot Learning, 2023.

[32] Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. Language-guided traffic simulation via scene-level diffusion. arXiv preprint arXiv:2306.06344, 2023.

[33] Adam Rashid, Satvik Sharma, Chung Min Kim, Justin Kerr, Lawrence Chen, Angjoo Kanazawa, and Ken Goldberg. Language embedded radiance fields for zero-shot task-oriented grasping. arXiv preprint arXiv:2309.07970, 2023.

[34] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 19729–19739, 2023.

[35] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, et al. Language to rewards for robotic skill synthesis. arXiv preprint arXiv:2306.08647, 2023.

[36] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. arXiv preprint arXiv:2307.05973, 2023.

[37] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. arXiv preprint arXiv:2310.12931, 2023.

[38] Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. arXiv preprint arXiv:2307.01928, 2023.

[39] Jesse Zhang, Jiahui Zhang, Karl Pertsch, Ziyi Liu, Xiang Ren, Minsuk Chang, Shao-Hua Sun, and Joseph J Lim. Bootstrap your own skills: Learning to solve new tasks with large language model guidance. arXiv preprint arXiv:2310.10021, 2023.

[40] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. arXiv preprint arXiv:2307.06135, 2023.

[41] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In Proceedings of the IEEE/CVF international conference on computer vision, pages 5664–5673, 2019.

[42] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. arXiv preprint arXiv:2302.12422, 2023.

[43] Shiqi Liu, Mengdi Xu, Peide Huang, Xilun Zhang, Yongkang Liu, Kentaro Oguchi, and Ding Zhao. Continual vision-based reinforcement learning with group symmetries. In 7th Annual Conference on Robot Learning, 2023.

[44] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. arXiv preprint arXiv:2308.07931, 2023.

[45] Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. arXiv preprint arXiv:2308.16891, 2023.

[46] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. arXiv preprint arXiv:2306.14896, 2023.