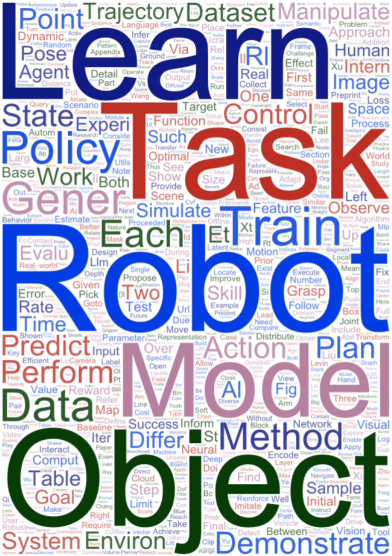


Robot Learning

TAMP & Language

Marc Toussaint
Technical University of Berlin
Summer 2024

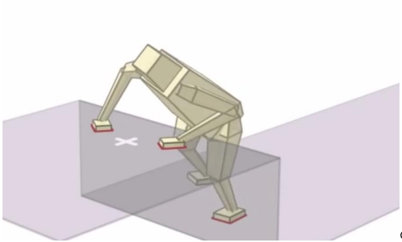
Remaining Lectures

- June 25: TAMP & Language
- July 2: Multi-Robot Learning
- July 9: Robot Learning Discussion – Lecture Feedback – Exam Info

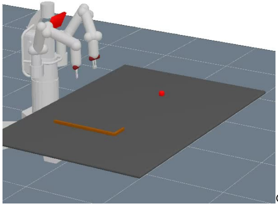
Outline

- Background on Task and Motion Planning (TAMP)
- Learning in TAMP
- Language in Robotics
- LLMs & TAMP

Task and Motion Planning (TAMP) examples:

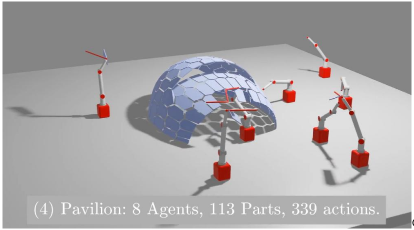


Mordatch et al: CIO (SIGGRAPH'12)



Toussaint at al: LGP (RSS'18)

Garrett et al: PDDLStream (ICAPS'20)



Hartmann et al. (IROS 20)

Task and Motion Planning (TAMP)

- What is the right level of “abstraction” to reason about manipulation?

Task and Motion Planning (TAMP)

- What is the right level of “abstraction” to reason about manipulation?
  - Low-level motor commands? (Torques?)
  - Mid-level kinematic commands? (6D endeff target position/velocity)
  - Actions/skills? (Pick, place, push, throw, hit, *how long is the list?*)

Abstractions

- What does the AI/RL researcher say about abstractions?
  - Hierarchical MDPs, Options, Hierarchical RL
  - (Classical AI: Landmarks in A* search)
  - Abstraction learning is hard:
    - Given action primitives $\to$ state abstractions clear (Konidaris' work)
    - Given state abstractions $\to$ action primitives clear ("skill discovery")
    - Classical ideas for state abstractions: identifying bottlenecks (=doors in configuration space; McGovern, Barto 2001)
  - Modern view: Data-driven: Assume tons of demonstrations and cluster-segment them

Abstractions

- What does the AI/RL researcher say about abstractions?
  - Hierarchical MDPs, Options, Hierarchical RL
  - (Classical AI: Landmarks in A* search)
  - Abstraction learning is hard:
    - Given action primitives $\rightarrow$ state abstractions clear (Konidaris' work)
    - Given state abstractions $\rightarrow$ action primitives clear ("skill discovery")
    - Classical ideas for state abstractions: identifying bottlenecks (=doors in configuration space; McGovern, Barto 2001)
  - Modern view: Data-driven: Assume tons of demonstrations and cluster-segment them

- What does the Roboticist say about abstractions?

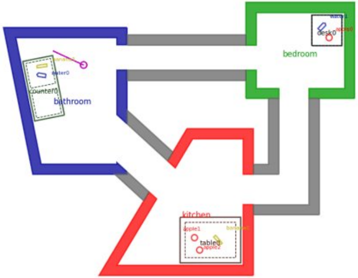
Abstractions

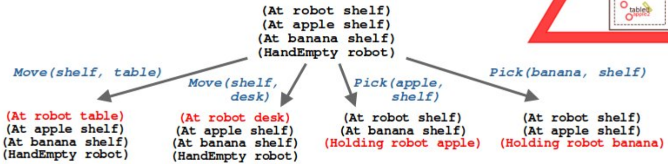
- What does the AI/RL researcher say about abstractions?
  - Hierarchical MDPs, Options, Hierarchical RL
  - (Classical AI: Landmarks in A* search)
  - Abstraction learning is hard:
    - Given action primitives $\rightarrow$ state abstractions clear (Konidaris' work)
    - Given state abstractions $\rightarrow$ action primitives clear ("skill discovery")
    - Classical ideas for state abstractions: identifying bottlenecks (=doors in configuration space; McGovern, Barto 2001)
  - Modern view: Data-driven: Assume tons of demonstrations and cluster-segment them

- What does the Roboticist say about abstractions?
  - Force level, motion level, task level
  - Task level: discrete symbolic state and actions (STRIPS/PDDL)

STRIPS/PDDL

```
(:action move
    :parameters (?r ?loc1 ?loc2)
    :precondition (and (Robot ?r)
                       (Location ?loc1)
                       (Location ?loc2)
                       (At ?r ?loc1))
    :effect (and (At ?r ?loc2)
                 (not (At ?r ?loc1)))
)
```





– A symbolic state s_t is a set of grounded literals

– A symbolic action operators defines a precondition and effect

– Eventually, **his defines the set of possible successor states** $s_{t+1} \in$ **succ**(s_t)

Task and Motion Planning

- Task-level is defined by
  - symbols (predicates), objects (constants), and action operators
  - initial state s_0, goal sentence, action operators imply succ(s_t)
- Motion-level is defined by
  - world configuration space $\mathcal{X}$, goal configurations $\mathcal{X}_{\text{goal}} \subseteq \mathcal{X}$
  - feasible space $\mathcal{X}_{s,\theta} \subseteq \mathcal{X}$ depending on logic state s and *entry point* θ (action parameter)

    [$\mathcal{X}_{s,\theta}$ is called *foliation*, or multi-modal space $\rightarrow$ **multi-modal motion planning (MMMP)**]
- Path-Finding formulation of TAMP:
  - Find sequence of (s_i, τ_i) of symbolic states and continuous feasible paths τ_i that lead to goal:
  - Paths: $\tau_i : [0,1] \rightarrow \mathcal{X}_{s_i,\theta_i}$
  - Continuity: $\tau_i(0) = \tau_{i\text{-}1}(1)$
  - Entry points: $\theta_i = \tau_{i\text{-}1}(1)$ (e.g. action parameter, grasp, lower-dim feature of $\tau_{i\text{-}1}(1)$)
  - Goal: $s_K \models \text{goal}, \tau_K(1) \in \mathcal{X}_{\text{goal}}$

C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez. Integrated Task and Motion Planning.
Annual Review of Control, Robotics, and Autonomous Systems, 4(1):265–293, 2021-05-03.
URL: https://www.annualreviews.org/delete_doi/10.1146/annurev-control-091420-084139

TAMP as Logic-Geometric Program (LGP)

$$\min_{\substack{s_{1:K} \\ x:[0,KT]\to\mathcal{X}}} \int_0^{KT} c(\underline{x}(t))\, dt$$

$$\begin{aligned}
\text{s.t.}\quad & x(0) = x_0, \\
& \forall_{t\in[0,T]} : \ \bar\phi(\underline{x}(t), s_{k(t)}) \le 0 \\
& \forall_{k\in\{1,..,K\}} : \ \hat\phi(\underline{x}(t_k), s_{k\text{-}1}, s_k) \le 0 \\
& s_K \models \text{goal},\ \forall_{k\in\{1,..,K\}} : \ s_k \in \text{succ}(s_{k\text{-}1})
\end{aligned}$$

- Skeleton $s_{1:K}$ defines schedule of physical modes
- Constraints $\hat\phi, \bar\phi$ define correct physics **differentiable**

[inequalities subsume equalities; $\underline{x} = (x, \dot x, \ddot x)$]

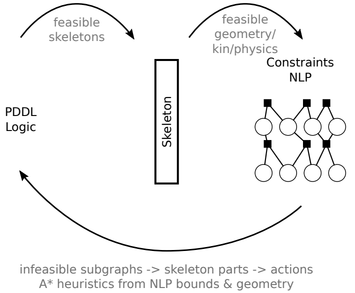
M. Toussaint. Logic-Geometric Programming: An Optimization-Based Approach to Combined Task and Motion Planning. In *IJCAI*, pages 1930–1936, 2015.
URL: https://argmin.lis.tu-berlin.de/papers/15-toussaint-IJCAI.pdf

M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. 2018.
URL: https://dspace.mit.edu/handle/1721.1/126626

TAMP as Logic-Geometric Program (LGP)

$$\min_{\substack{s_{1:K} \\ x:[0,KT]\to\mathcal{X}}} \int_0^{KT} c(\underline{x}(t))\, dt$$

$$\begin{aligned} \text{s.t.}\quad & x(0) = x_0, \\ & \forall_{t\in[0,T]} : \bar\phi(\underline{x}(t), s_{k(t)}) \le 0 \\ & \forall_{k\in\{1,..,K\}} : \hat\phi(\underline{x}(t_k), s_{k\text{-}1}, s_k) \le 0 \\ & s_K \models \text{goal},\ \forall_{k\in\{1,..,K\}} : s_k \in \text{succ}(s_{k\text{-}1}) \end{aligned}$$



- Skeleton $s_{1:K}$ defines schedule of physical modes
- Constraints $\hat\phi, \bar\phi$ define correct physics **differentiable**

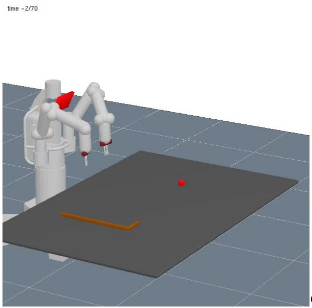
  [inequalities subsume equalities; $\underline{x} = (x, \dot x, \ddot x)$]

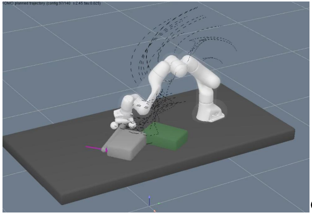
- Solving implies searching over $s_{1:K}$ and solving the corresponding NLP

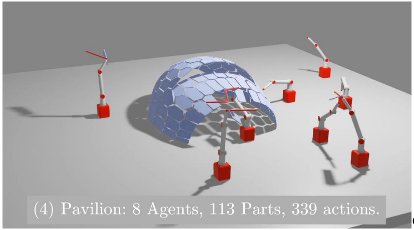
M. Toussaint. Logic-Geometric Programming: An Optimization-Based Approach to Combined Task and Motion Planning. In *IJCAI*, pages 1930–1936, 2015.
URL: https://argmin.lis.tu-berlin.de/papers/15-toussaint-IJCAI.pdf

M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. 2018.
URL: https://dspace.mit.edu/handle/1721.1/126626

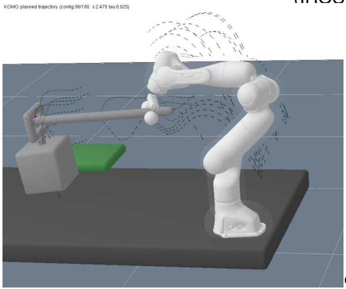
renderings of example solutions...



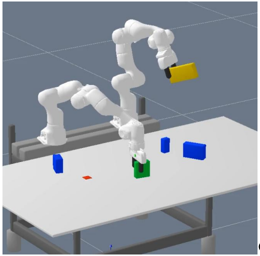




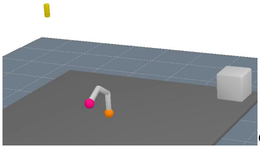
(IROS 20)



(IROS 20)



(R:SS 20)



Abstractions

- What does “LGP” say about abstractions?

Abstractions

- What does “LGP” say about abstractions?
  - There are two levels: the convex level (NLP), and the non-convex (discrete decisions)

Outline

- Intro to Task and Motion Planning (TAMP)
- **Learning in TAMP**
- Language in Robotics
- LLMs & TAMP

Is model-based TAMP a dead end?

- LGP formulates TAMP as model-based optimization problem
  - Assumption of having a world model is unrealistic (state estimation from vision ill-posed...)
  - High computation time for large problems – why plan from scratch every time?

Is model-based TAMP a dead end?

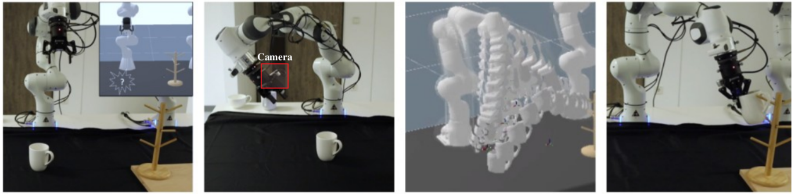
- LGP formulates TAMP as model-based optimization problem
  - Assumption of having a world model is unrealistic (state estimation from vision ill-posed...)
  - High computation time for large problems – why plan from scratch every time?

- Opportunities for learning:
  - **Replace exact model by learned constraints** $\phi(x)$
    - The LGP definition actually only needs constraints $\phi(x)$, no explicit world model
    - Instead of hand-defining these from a model $\to$ image-conditional neural models $\phi_\theta(x; \mathcal{I})$
  - **Learn to predict plans**
    - Instead of solving from scratch, learn to predict promising actions $a_{1:K}$ from the scene image

- Replace exact model by learned constraints $\phi(x)$:

Deep Visual Constraints: Neural Implicit Models for Manipulation Planning from Visual Input

Jung-Su Ha   Danny Driess   Marc Toussaint
Learning & Intelligent Systems Lab, TU Berlin, Germany



(a) No object model   (b) See   (c) Plan   (d) Act

- Learn $\phi(x, \mathcal{I})$ with V input images $\mathcal{I}$ s.t.:
  - $\phi(x; \mathcal{I}) = 0 \quad \Leftrightarrow \quad x$ is correct grasp
  - $\phi(x; \mathcal{I}) = 0 \quad \Leftrightarrow \quad x$ is correct hanging
- Data generating in simulation:
  - Collect trial-and-error data on correct grasps and hanging

J.-S. Ha, D. Driess, and M. Toussaint. Deep visual constraints: Neural implicit models for manipulation planning from visual input. *IEEE Robotics and Automation Letters*, 7(4):10857–10864, 2022.
URL: https://ieeexplore.ieee.org/abstract/document/9844753/

Deep Visual Constraints: Network Architecture

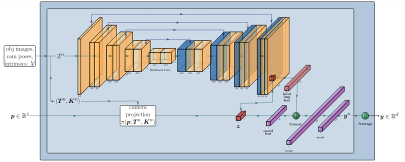


Fig. 3: PIFO (i) encodes the images $\mathcal{I}$ as pixel-wise feature images $\mathcal{F}$ via U-net, (ii) projects the query point $p \in \mathbb{R}^3$ into the pixel coordinate $z \in \mathbb{R}^2$ using known camera geometry, and (iii) computes the object representation vector $y \in \mathbb{R}^d$ by extracting the local image features at the projected points.

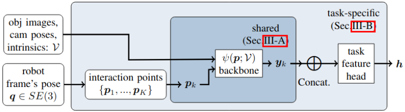


Fig. 2: The interaction feature prediction scheme of DVC

J.-S. Ha, D. Driess, and M. Toussaint. Deep visual constraints: Neural implicit models for manipulation planning from visual input.
IEEE Robotics and Automation Letters, 7(4):10857–10864, 2022.
URL: https://ieeexplore.ieee.org/abstract/document/9844753/

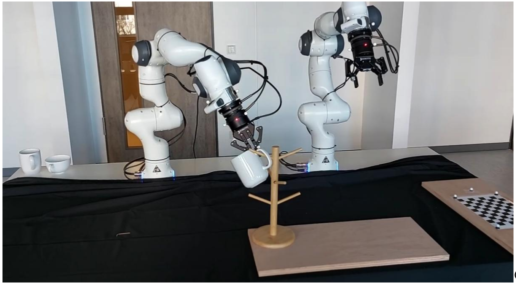
- Camera views $\mathcal{I} = \{(I^1, K^1), ..., (I^V, K^V)\}$
  Wanted: image-based constraint model
  $$\phi(x; \mathcal{I})$$

- First train a d-dimensional **field representation**
  $$y(p; \mathcal{I}) = \frac{1}{V} \sum_i \text{MLP}(\text{UNet}(I^i, K^i(x)), K^i(x))$$
  [$p \in \mathbb{R}^3$, pre-trained for shape decoding (SDF prediction)]

- Function is queried at finite set of *interaction points* $p_1(x), .., p_K(x)$ to get the feature
  $$\phi(x; \mathcal{I}) = \text{MLP}(y(p_1(x); \mathcal{I}), .., y(p_K(x); \mathcal{I}))$$
  [fine-tuned for manipulation success (trial & error in sim)]

Deep Visual Constraints

(No search over skeletons, no reactive MPC, just optimal path for given sequence of constraints.)



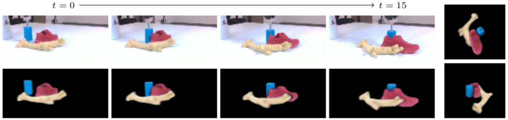
Similar: Learn Dynamics Constraints

Learning Multi-Object Dynamics with Compositional Neural Radiance Fields

Danny Driess TU Berlin | **Zhiao Huang** UC San Diego | **Yunzhu Li** MIT | **Russ Tedrake** MIT | **Marc Toussaint** TU Berlin

D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint. Learning multi-object dynamics with compositional neural radiance fields.
In *Conference on Robot Learning*, pages 1755–1768, 2023.
URL: `https://proceedings.mlr.press/v205/driess23a.html`

`https://dannydriess.github.io/compnerfdyn/`



(a) Bottom row renderings of forward predictions with dynamic model, top row ground truth (b) novel view

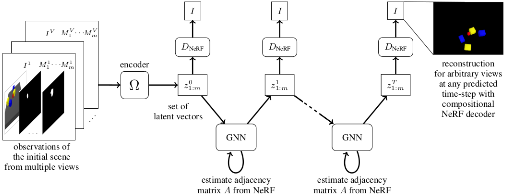


Figure 2: Overview of the dynamics prediction framework. The initial scene observations are encoded with Ω into a set of latent vectors $z_{1:m}$, each representing the objects individually. The GNN dynamics model predicts the evolution of the latent vectors. At each step, the predicted latent vectors can be rendered into an arbitrary view with the compositional NeRF decoder. Refer to the appendix for visualizations of Ω and the GNN.

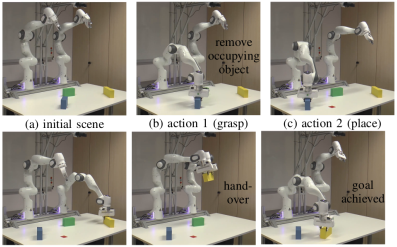
- Each object has a latent code z_i^t
- learn dynamics $z_{1:m}^t \mapsto z_i^{t+1}$!

- Learning to predict plans..

Deep Visual Reasoning: Learning to Predict Action Sequences for Task and Motion Planning from an Initial Scene Image

Danny Driess Jung-Su Ha Marc Toussaint

Machine Learning and Robotics Lab, University of Stuttgart, Germany
Max-Planck Institute for Intelligent Systems, Stuttgart, Germany
Learning and Intelligent Systems Group, TU Berlin, Germany



(a) initial scene (b) action 1 (grasp) (c) action 2 (place)
(d) action 3 (grasp) (e) action 4 (grasp) (f) action 5 (place)

Fig. 1. Typical scene: The yellow object should be placed on the red spot, which is, however, occupied by the blue object. Furthermore, the yellow object cannot be reached by the robot arm that is able to place it on the red spot.

- Data collection $D = \{(S^i, g^i, a^i_{1:K^i}, F^i)\}_{i=1}^n$
  - with scene S^i, goal g^i, actions $a^i_{1:K^i}$, feasibility F^i
  - random generated "in simulation", **model-based TAMP solver used to label feasibility**

- Train a sequential policy:
  $\pi(a_k; g, a_{1:k\text{-}1}, S)$
  $P(\exists_{K>K} \exists_{a_{k+1:K}} : a_{1:K}$ feasible $\mid a_k, g, a_{1:k\text{-}1}, S)$
  - Similar to language model: Predict next "token" a_k given previous $a_{1:k\text{-}1}$ conditional g, S

D. Driess, J.-S. Ha, and M. Toussaint. Deep Visual Reasoning: Learning to Predict Action Sequences for Task and Motion Planning from an Initial Scene Image, 2020-06-09.
URL: `http://arxiv.org/abs/2006.05398`, patharXiv:2006.05398

Deep Visual Reasoning: Network Architecture

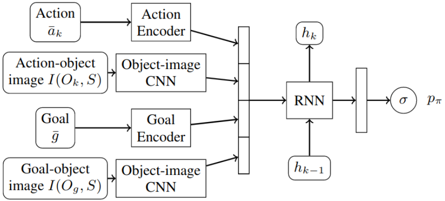
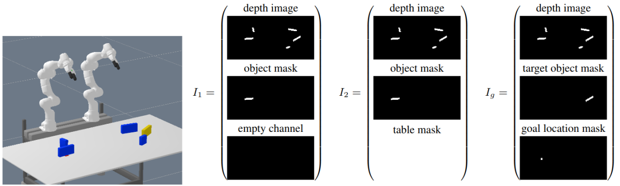


Figure 3. Proposed neural network architecture.



(a) Initial scene

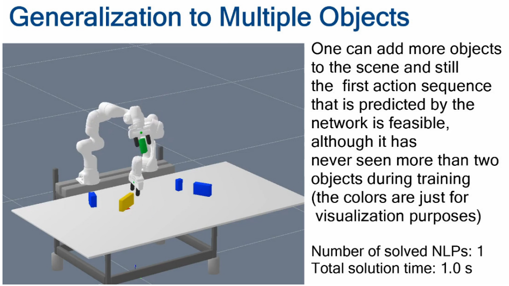
(b) Action-object image for `grasp` action with the left robot arm and the blue object that occupies the red goal location

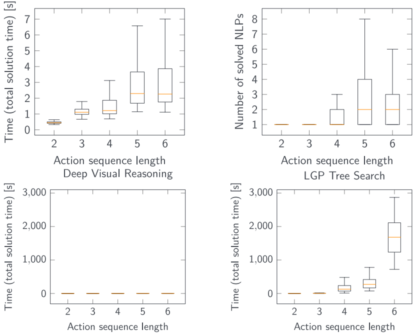
(c) Action-object image for `place` action with the left robot arm representing placing the blue object on the table

(d) Goal- and action-object image (for `place` action) representing placing the yellow target object on the red goal location

- Uses RNN – modern version would use transformer
- Special encoding of predicates $\bar{a}, \bar{g}$ and references O (as masks)

Deep Visual Reasoning: Results





- Often, the first proposed action sequence is feasible

Outline

- Intro to Task and Motion Planning (TAMP)
- Learning in TAMP
- **Language in Robotics**
- LLMs & TAMP

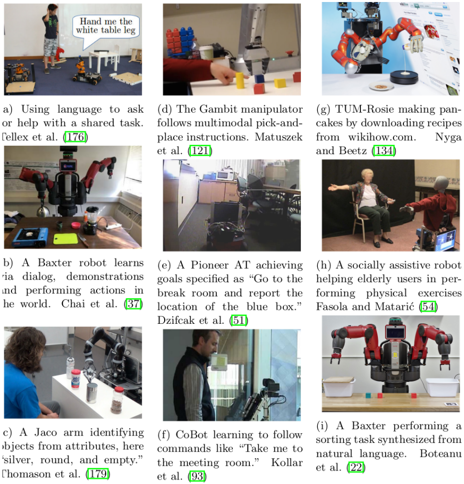
Robots That Use Language: A Survey

Stefanie Tellex[1], Nakul Gopalan[2], Hadas Kress-Gazit[3], and Cynthia Matuszek[4]

S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek. Robots That Use Language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):25–55, 2020-05-03.
URL: https://www.annualreviews.org/delete_doi/10.1146/annurev-control-101119-071628

- Great survey on Natural Language Robot Interaction
  - Using natural language to command robots, set tasks
  - Using natural language to instruct robots, e.g. as part of demonstrations
  - Different to standard NLP or dialog systems: **language needs to be physically grounded**

Natural Language Robot Interaction: Examples



a) Using language to ask or help with a shared task. Tellex et al. (176)

(d) The Gambit manipulator follows multimodal pick-and-place instructions. Matuszek et al. (121)

(g) TUM-Rosie making pancakes by downloading recipes from wikihow.com. Nyga and Beetz (134)

b) A Baxter robot learns via dialog, demonstrations and performing actions in the world. Chai et al. (37)

(e) A Pioneer AT achieving goals specified as “Go to the break room and report the location of the blue box.” Dzifcak et al. (51)

(h) A socially assistive robot helping elderly users in performing physical exercises Fasola and Matarić (54)

c) A Jaco arm identifying objects from attributes, here 'silver, round, and empty.” Thomason et al. (179)

(f) CoBot learning to follow commands like “Take me to the meeting room.” Kollar et al. (93)

(i) A Baxter performing a sorting task synthesized from natural language. Boteanu et al. (22)

Figure 1: Robots used for language-based interactions.

from [9]

- robot asks for help
- human sets task (with language & gesture)
- robot “reads/comprehends” wikihow
- demonstrations via dialog
- human sets task (nagivation)
- ...
- human sets task (object identification)
- human sets task (navigation)
- human sets task (manipulation)

Natural Language Robot Interaction: Datasets

Dataset	Type of Data	Link to dataset
MARCO dataset (111)	Navigation instructions given to a robot to navigate a map, and the route followed.	www.cs.utexas.edu/users/ml/clamp/navigation/
Scene dataset(98)	Images and descriptions of objects in the image.	rtw.ml.cmu.edu/tacl2013_lsp/
Cornell NLVR dataset (168)	Pairs of images and logical statements about them which are true or false.	lic.nlp.cornell.edu/nlvr/
CLEVR dataset (84)	Images and question-answer pairs.	cs.stanford.edu/people/jcjohns/clevr/
Embodied Question Answering (47)	Pairs of questions and answers in simulated 3D environments. The agent needs to search the environment to find the answer.	embodiedqa.org
Visual Question Answering in Interactive Environments (65)	Pairs of questions and answers in different simulated 3D environments.	github.com/danielgordon10/thor-iqa-cvpr-2018
Room-to-Room (R2R) Navigation (4)	Panoramic views in real buildings, paired with instructions to be followed.	bringmeaspoon.org/
H2R lab language grounding datasets (9, 64)	Predicate based sub-goal conditions paired with natural language instructions.	github.com/h2r/language_datasets
Cornell Instruction Following Framework (17, 125)	Data for three separate navigation domains in 3D environments, containing instructions paired with trajectories.	github.com/clic-lab/ciff
MIT Spatial Language Understanding dataset (92, 172)	Pairs of language command and trajectories for navigation and mobile manipulation.	people.csail.mit.edu/stefie10/slu/

Table 2: Datasets used in Language Grounding and Robotics

"Data sets typically consist of natural language paired with some form of sensor-based context information about the physical environment"

- Previous survey highlights substantial literature on Natural Language Robot Interaction *before* rise of LLMs

  Example: `https://youtu.be/VqSb-ZZuIwI?t=2523`

CLIP (Contrastive Language-Image Pre-training)

Learning Transferable Visual Models From Natural Language Supervision

Alec Radford [*1] Jong Wook Kim [*1] Chris Hallacy [1] Aditya Ramesh [1] Gabriel Goh [1] Sandhini Agarwal [1]
Girish Sastry [1] Amanda Askell [1] Pamela Mishkin [1] Jack Clark [1] Gretchen Krueger [1] Ilya Sutskever [1]

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, and J. Clark. Learning transferable visual models from natural language supervision.
In *International Conference on Machine Learning*, pages 8748–8763, 2021.
URL: `http://proceedings.mlr.press/v139/radford21a`

“We demonstrate that the simple pre-training task of predict- ing which caption goes with which image is an efficient and scalable way to learn SOTA image rep- resentations from scratch on a dataset of 400 million (image, text) pairs collected from the internet.”

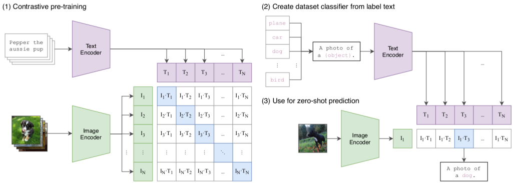


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset’s classes.

[Contrastive Training: “maximize the cosine similarity of the image and text embeddings of the N real pairs in the batch while minimizing the cosine similarity of the embeddings of the $N^2 - N$ incorrect pairings.]

CLIPort

CLIPort: What and Where Pathways for Robotic Manipulation

Mohit Shridhar [1,†] **Lucas Manuelli** [2] **Dieter Fox** [1,2]
[1]University of Washington [2]NVIDIA
mshr@cs.washington.edu lmanuelli@nvidia.com fox@cs.washington.edu

M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation.
In *Conference on Robot Learning*, pages 894–906, 2022.
URL: https://proceedings.mlr.press/v164/shridhar22a.html

https://cliport.github.io/

"CLIPort: a language-conditioned imitation-learning agent that combines the broad semantic understanding (what) of CLIP with the spatial precision (where) of Transporter"

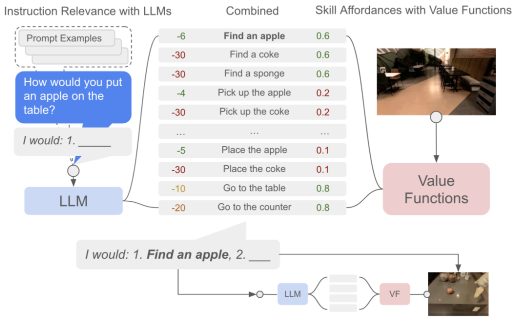
- Trains a policy $\pi : (y_i, l_l) \mapsto a_t$
  - top-down orthographic RGB-D y_t, language instruction l_t, pick-n-place 2D coordinates a_t

SayCan

Do As I Can, Not As I Say: Grounding Language in Robotic Affordances

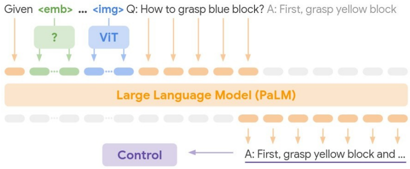
A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, and R. Julian. Do as i can, not as i say: Grounding language in robotic affordances.
In *Conference on Robot Learning*, pages 287–318, 2023.
URL: https://proceedings.mlr.press/v205/ichter23a.html

https://say-can.github.io/



- Use a LLM (PaLM) to predict *multiple* actions (with probabilities)
- Multiply each option with *affordance prediction* (= probability of success)

PaLM-E



PaLM-E: An Embodied Multimodal Language Model

Danny Driess [1,2] **Fei Xia** [1] **Mehdi S. M. Sajjadi** [3] **Corey Lynch** [1] **Aakanksha Chowdhery** [3]
Brian Ichter [1] **Ayzaan Wahid** [1] **Jonathan Tompson** [1] **Quan Vuong** [1] **Tianhe Yu** [1] **Wenlong Huang** [1]
Yevgen Chebotar [1] **Pierre Sermanet** [1] **Daniel Duckworth** [3] **Sergey Levine** [1] **Vincent Vanhoucke** [1]
Karol Hausman [1] **Marc Toussaint** [2] **Klaus Greff** [3] **Andy Zeng** [1] **Igor Mordatch** [3] **Pete Florence** [1]

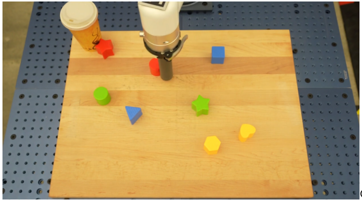
[1]Robotics at Google [2]TU Berlin [3]Google Research

D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence. PaLM-E: An Embodied Multimodal Language Model, 2023-03-06. URL: `http://arxiv.org/abs/2303.03378`, patharXiv:2303.03378

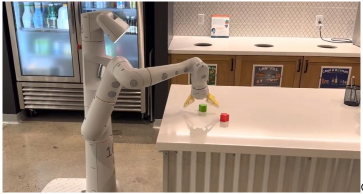
`https://palm-e.github.io/`

- Input: *Multi-modal sentence:*
  - Interleaves words, images (with segmentation), vectors, reference-keywords
  - All token-encoded
  - Various image encodings (ViT, object-centric ViT, OSRT, NeRFs pre-trained)
- Output:
  - Sequences of action primitives (previously trained, RT-1)

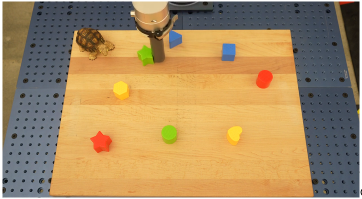
"Bring me the rice chips from the drawer"



"Push red blocks to the coffee cup"



"Bring me the green star"



"Push green blocks to the turtle"

Example input/output

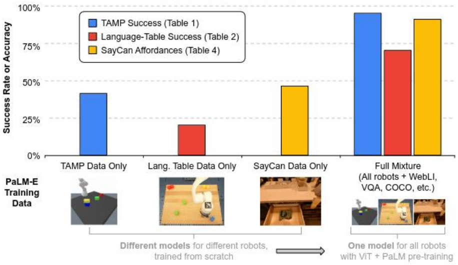
- Prompt: `Given <img>. Q: How to grasp the green object?.`
  Target: `A: First grasp the orange object and place it on the table, then grasp the green object.`

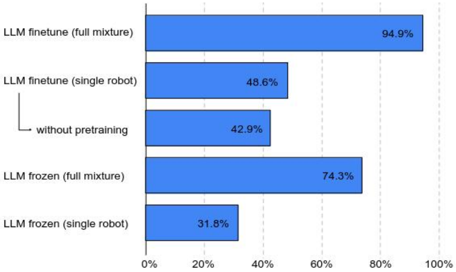
- Prompt: `Given <img>. Q: How to stack the white object on top of the red object?.`
  Target: `A: First grasp the green object and place it on the table, then grasp the white object and place it on the red object.`

PaLM-E Evaluations

- Data sets:
  - **TAMP data (generated by our LGP-TAMP planner)**
  - Table data (previous RT1 paper)
  - SayCan data
  - Other visual/language data: WebLI, VQA, COCO, etc.
- Pre-taining:
  - LLM backbone: language, VQA (WebLI, VQA, COCO)
  - Encodings: reconstruction, auto-encoding
- Ablation studies:
  - Varying transformer sizes
  - generalization (to unseen object situations, esp. higher number of objects)
  - freezing, refining, full-learning of backbone LLM or encodings
  - with full/partial choice of data sets & sizes
  - various image encodings

PaLM-E evaluations





	Object-centric	LLM pre-train	Embodied VQA q_1	q_2	q_3	q_4	Planning p_1	p_2
SayCan (oracle afford.) (Ahn et al., 2022)		✓	-	-	-	-	38.7	33.3
PaLI (zero-shot) (Chen et al., 2022)		✓	-	0.0	0.0	-	-	-
PaLM-E (ours) w/ input enc:								
State	✓(GT)	✗	99.4	89.8	90.3	88.3	45.0	46.1
State	✓(GT)	✓	**100.0**	96.3	95.1	93.1	55.9	49.7
ViT + TL	✓(GT)	✓	34.7	54.6	74.6	91.6	24.0	14.7
ViT-4B single robot	✗	✓	-	45.9	78.4	92.2	30.6	32.9
ViT-4B full mixture	✗	✓	-	70.7	93.4	92.1	74.1	74.6
OSRT (no VQA)	✓	✓	-	-	-	-	71.9	75.1
OSRT	✓	✓	99.7	**98.2**	**100.0**	**93.7**	**82.5**	**76.2**

Baselines				Failure det.	Affordance
PaLI (Zero-shot) (Chen et al., 2022)				0.73	0.62
CLIP-FT (Xiao et al., 2022)				0.65	-
CLIP-FT-hindsight (Xiao et al., 2022)				0.89	-
QT-OPT (Kalashnikov et al., 2018)				-	0.63
PaLM-E-12B trained on	from scratch	LLM+ViT pretrain	LLM frozen		
Single robot	✓	✗	n/a	0.54	0.46
Single robot	✗	✓	✓	**0.91**	0.78
Full mixture	✗	✓	✓	**0.91**	0.87
Full mixture	✗	✓	✗	0.77	**0.91**

Follow Up: RT-2

RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control

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, Brianna Zitkovich

B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, and A. Wahid. Rt-2: Vision-language-action models transfer web knowledge to robotic control.
In *Conference on Robot Learning*, pages 2165–2183, 2023.
URL: https://proceedings.mlr.press/v229/zitkovich23a.html

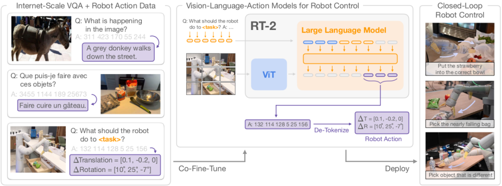


Figure 1: RT-2 overview: we represent robot actions as another language, which can be cast into text tokens and trained together with Internet-scale vision-language datasets. During inference, the text tokens are de-tokenized into robot actions, enabling closed loop control. This allows us to leverage the backbone and pretraining of vision-language models in learning robotic policies, transferring some of their generalization, semantic understanding, and reasoning to robotic control. We demonstrate examples of RT-2 execution on the project website: robotics-transformer2.github.io.

- quasi-continuous actions (trained end-to-end):

$$\text{“terminate } \Delta\text{pos}_x \;\; \Delta\text{pos}_y \;\; \Delta\text{pos}_z \;\; \Delta\text{rot}_x \;\; \Delta\text{rot}_y \;\; \Delta\text{rot}_z \;\; \text{gripper\_extension”.}$$

A possible instantiation of such a target could be: “1 128 91 241 5 101 127”. The two VLMs that we finetune in our experiments, PaLI-X [16] and PaLM-E [17], use different tokenizations. For PaLI-X,

Conclusion

- Levels of abstraction: Force, motion, task
- Task and Motion "Planning": Core problem formulation of robotic AI
  - TAMP theory & solvers are fully model-based
  - Clear opportunities for learning: constraint learning, learning to predict plans
- Language $\leftrightarrow$ task & action level
  - Lots of classical literature on *language grounding*
  - Connecting natural language with typical robot task descriptions (STRIPS/PDDL)
- Huge recent focus on marrying LLMs + TAMP + robotics

[1] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, and R. Julian.
Do as i can, not as i say: Grounding language in robotic affordances.
In *Conference on Robot Learning*, pages 287–318, 2023.
URL: `https://proceedings.mlr.press/v205/ichter23a.html`.

[2] D. Driess, J.-S. Ha, and M. Toussaint.
Deep Visual Reasoning: Learning to Predict Action Sequences for Task and Motion Planning from an Initial Scene Image, 2020-06-09.
URL: `http://arxiv.org/abs/2006.05398`, patharXiv:2006.05398.

[3] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint.
Learning multi-object dynamics with compositional neural radiance fields.
In *Conference on Robot Learning*, pages 1755–1768, 2023.
URL: `https://proceedings.mlr.press/v205/driess23a.html`.

[4] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence.
PaLM-E: An Embodied Multimodal Language Model, 2023-03-06.
URL: `http://arxiv.org/abs/2303.03378`, patharXiv:2303.03378.

[5] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez.
Integrated Task and Motion Planning.
Annual Review of Control, Robotics, and Autonomous Systems, 4(1):265–293, 2021-05-03.
URL: `https://www.annualreviews.org/delete_doi/10.1146/annurev-control-091420-084139`.

[6] J.-S. Ha, D. Driess, and M. Toussaint.
Deep visual constraints: Neural implicit models for manipulation planning from visual input.
IEEE Robotics and Automation Letters, 7(4):10857–10864, 2022.
URL: `https://ieeexplore.ieee.org/abstract/document/9844753/`.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, and J. Clark.
Learning transferable visual models from natural language supervision.
In *International Conference on Machine Learning*, pages 8748–8763, 2021.
URL: `http://proceedings.mlr.press/v139/radford21a`.

[8] M. Shridhar, L. Manuelli, and D. Fox.
Cliport: What and where pathways for robotic manipulation.
In *Conference on Robot Learning*, pages 894–906, 2022.
URL: `https://proceedings.mlr.press/v164/shridhar22a.html`.

[9] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek.
Robots That Use Language.
Annual Review of Control, Robotics, and Autonomous Systems, 3(1):25–55, 2020-05-03.
URL: `https://www.annualreviews.org/delete_doi/10.1146/annurev-control-101119-071628`.

[10] M. Toussaint.
Logic-Geometric Programming: An Optimization-Based Approach to Combined Task and Motion Planning.
In *IJCAI*, pages 1930–1936, 2015.
URL: `https://argmin.lis.tu-berlin.de/papers/15-toussaint-IJCAI.pdf`.

[11] M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum.
Differentiable physics and stable modes for tool-use and manipulation planning.
2018.
URL: `https://dspace.mit.edu/handle/1721.1/126626`.

[12] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, and A. Wahid.
Rt-2: Vision-language-action models transfer web knowledge to robotic control.
In *Conference on Robot Learning*, pages 2165–2183, 2023.
URL: `https://proceedings.mlr.press/v229/zitkovich23a.html`.