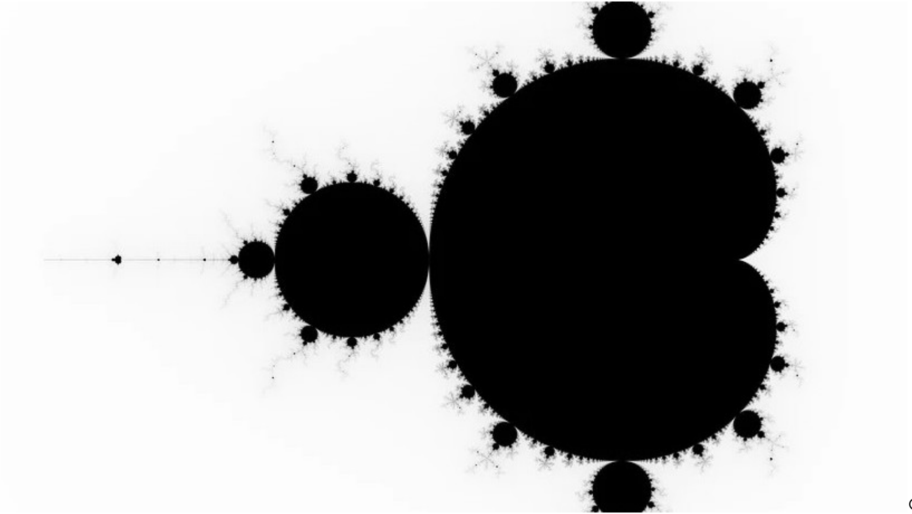
Do we need a “cognitive architecture” debate again?

Marc Toussaint

Learning & Intelligent Systems Lab, TU Berlin

Berkeley, March 6, 2023



- Imagine want to model $\mathcal{M}$

- Imagine want to model $\mathcal{M}$
- Imagine you want to train a neural network to predict $x \in \mathcal{M}$

- Imagine want to model $\mathcal{M}$
- Imagine you want to train a neural network to predict $x \in \mathcal{M}$

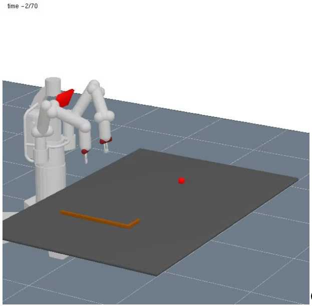
- Would you come up with that **representation**:

$$\mathcal{M} = \{c \in \mathbb{C} \ : \ [z_{k+1} \leftarrow z_k^2 + c] \text{ converges, } \text{with } z_0 = 0\}$$

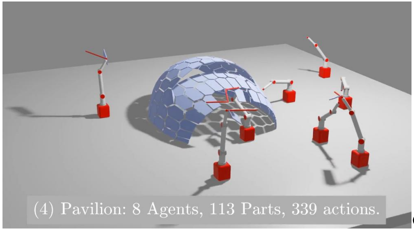
[implicit, requires computation for decoding, extremely powerful]

Reasoning as computational decoding of an implicited representation of behavior

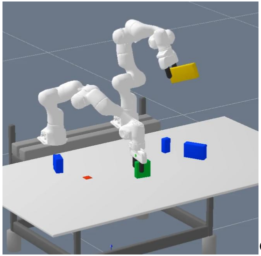
“Goals”, “tasks”, “constraints” are the latent code (latent variables) of that representation



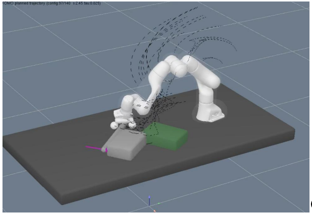
(R:SS 18)

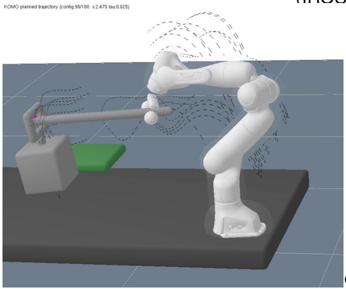


(IROS 20)

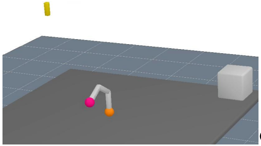


(R:SS 20)





(IROS 20)



- “perhaps we can think big again in robotics..”
  - general purpose LLMs
  - general purpose physical reasoning
  - general purpose robotic manipulation

- “perhaps we can think big again in robotics..”
  - general purpose LLMs
  - general purpose physical reasoning
  - general purpose robotic manipulation

- “Do we need a cognitive architecture debate again?”
  - LLMs and TAMP distinguish higher-level decisions from skills/control
  - Raises question of right abstractions/representations/interface
  - Action & scene representations

Action Representation

- SayCan, CLIPort, etc, also TAMP assume given/pre-trained skills

- SayCan, CLIPort, etc, also TAMP assume given/pre-trained skills

- Skill learning, skill discovery, state/action abstraction learning, options, hierarchical RL... discussed for decades

- SayCan, CLIPort, etc, also TAMP assume given/pre-trained skills

- Skill learning, skill discovery, state/action abstraction learning, options, hierarchical RL... discussed for decades
  - Typical RL-researchers view: A skill/option/primitive defined by policy $\pi_i : x \mapsto u$, often maximizing an associated reward R_i, perhaps initiation/termination sets

What are alternative views?

Boston Dynamics (2018)

- **We should have a “glue” between such kind of reactive control and higher-level TAMP/LLMs**

Implicit Action Representations

- Higher level decisions are about which objectives/constraints are active

Implicit Action Representations

- Higher level decisions are about which objectives/constraints are active
- Consider we have a *library* $\{\phi_i(x)\}$ of objective/constraint functions
  - A higher-level decision is set $\phi_i(x)$ as active objective
  - Decisions are *functions* passed to a lower-level control loop

explicit: $\pi_i(u|x)$ implicit: $u = \text{argmin}_u \text{MPC}(\{\phi_i\}_{\text{active}})$

Implicit Action Representations

- Higher level decisions are about which objectives/constraints are active
- Consider we have a *library* $\{\phi_i(x)\}$ of objective/constraint functions
  - A higher-level decision is set $\phi_i(x)$ as active objective
  - Decisions are *functions* passed to a lower-level control loop

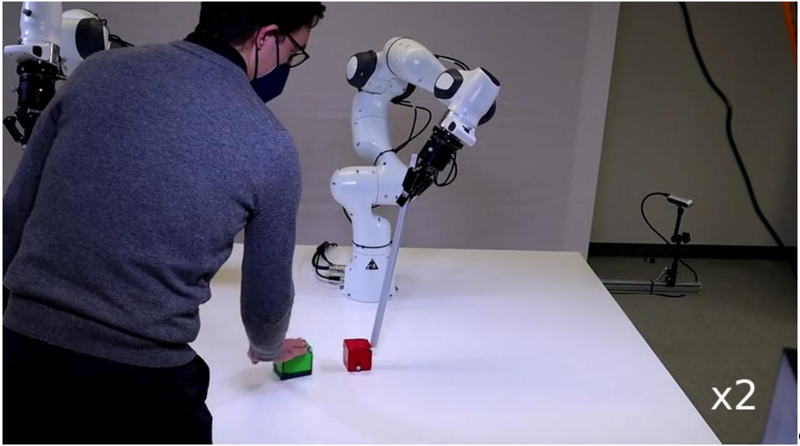
    explicit: $\pi_i(u|x)$ implicit: $u = \text{argmin}_u \text{MPC}(\{\phi_i\}_{\text{active}})$
  - cp. conditional random fields, energy-based models, implicit functions

Implicit Action Representations

- Higher level decisions are about which objectives/constraints are active
- Consider we have a *library* $\{\phi_i(x)\}$ of objective/constraint functions
  - A higher-level decision is set $\phi_i(x)$ as active objective
  - Decisions are *functions* passed to a lower-level control loop

    explicit: $\pi_i(u|x)$ implicit: $u = \mathrm{argmin}_u \text{MPC}(\{\phi_i\}_{\text{active}})$
  - cp. conditional random fields, energy-based models, implicit functions
- Why?
  - ϕ_i are easier to learn (or hand-code)?
  - Great generalization & compositionality

Sequence-of-Constraints MPC



- MPC through a given *sequence* of constraints $\phi_{1:K}$

  Sequence-of-Constraints MPC: Reactive Timing-Optimal Control of Sequential Manipulation, Toussaint, Harris, Ha, Driess, Hnig. IROS 2022

Related Work

Using pre-defined controllers per action:

Representing robot task plans as robust logical-dynamical systems, Paxton, Ratliff, Eppner, Fox. IROS 2019

Reactive task and motion planning under temporal logic specifications, Li, Park, Sung, Shah, Roy. ICRA 2021

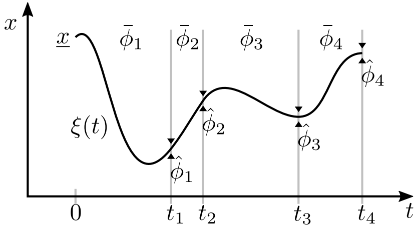
... or interpreting reference trajectories relative to objects:

Object-centric task and motion planning in dynamic environments, Migimatsu, Bohg. RAL 2020

... or online kino-dynamic replanning:

Modeling and planning manipulation in dynamic environments, Schmitt, Wirnshofer, Wurm, v Wichert, Burgard. ICRA 2019

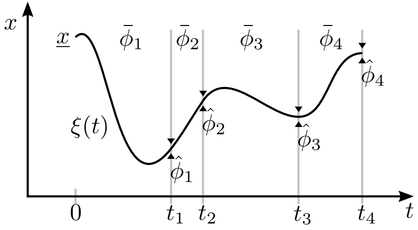
Sequence-of-Constraints MPC (SecMPC)



- Provided from higher-level (e.g. TAMP): **Sequence of**
  - **waypoint constraints** $\hat{\phi}_{1:K}$
  - **running constraints** $\bar{\phi}_{1:K}$

  which impose $\hat{\phi}_i(\xi(t_i)) \leq 0$ and $\bar{\phi}_i(\xi(t_{i\text{-}1} < t \leq t_i)) \leq 0$

Sequence-of-Constraints MPC (SecMPC)



- Provided from higher-level (e.g. TAMP): **Sequence of**
  - **waypoint constraints** $\hat\phi_{1:K}$
  - **running constraints** $\bar\phi_{1:K}$

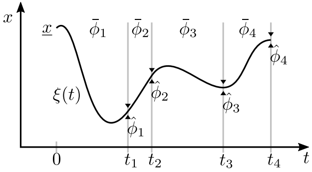
  which impose $\hat\phi_i(\xi(t_i)) \le 0$ and $\bar\phi_i(\xi(t_{i\text{-}1} < t \le t_i)) \le 0$

- Problem:
  - robustly transition through them
  - backtrack if a constraint is missed ("re-initiation")

Problem Formulation

- Org. SecMPC Problem:

$$
\begin{aligned}
\min_{\xi, t_{1:K}} \quad & t_K + \alpha \int_0^{t_K} c(\xi(t), \dot\xi(t), \ddot\xi(t))\, dt \\
\text{s.t.} \quad & \xi(0) = x,\ \dot\xi(0) = \dot x,\ \dot\xi(t_K) = 0\ , \\
& \forall_k : 0 < t_k < t_{k+1}\ , \\
& \forall_k : \hat\phi_k(\xi(t_k)) \le 0, \\
& \forall_{t \in [t_{k\text{-}1}, t_k]} : \bar\phi_k(\underline{\xi}(t)) \le 0\ .
\end{aligned}
$$



- *Approximate* decomposition:
  In each MPC cycle, sequentially solve:
  1) **Waypoints** sub-problem:

$$
\begin{aligned}
\min_{x_{1:K}} \quad & \textstyle\sum_{k=1}^K \tilde c(x_{k\text{-}1}, x_k) \\
\text{s.t.} \quad & \forall_k : \hat\phi_k(x_k) \le 0,\ \bar\phi_k(x_{k\text{-}1}, x_k) \le 0
\end{aligned}
$$

  2) **Timing** sub-problem:

$$
\min_{\tau_{1:K}, v_{1:K\text{-}1}} \textstyle\sum_{k=1}^K \tau_K + \alpha \sum_{k=1}^K \psi(x_{k\text{-}1}, v_{k\text{-}1}, x_k, v_k, \tau_k)\ ,
$$

  with cubic spline piece cost ψ

  3) **Receding horizon path** sub-problem:

$$
\begin{aligned}
\min_{\xi} \quad & \textstyle\int_0^H \alpha\, \ddot\xi(t)^2 + \|\xi(t) - \xi^*(t)\|^2\, dt \\
\text{s.t.} \quad & \xi(0) = x,\ \dot\xi(0) = \dot x\ , \\
& \forall_{t \in [0,H]} : \bar\phi_{k(t)}(\underline{\xi}(t)) \le 0
\end{aligned}
$$

  where ξ^* is defined by timed waypoints.

MPC cycle

- In each cycle ($\sim$10Hz in experiments)
  - solve for sequence of waypoints $x_{\kappa:K}$ given $\underline{x}, \tau_{\kappa:K}$
  - solve for timing $\tau_{\kappa:K}, v_{\kappa:K}$ given $x_{\kappa:K}$
  - solve for path ξ^H given $\tau_{\kappa:K}, x_{\kappa:K}, v_{\kappa:K}$

MPC cycle

- In each cycle ($\sim$10Hz in experiments)
  - solve for sequence of waypoints $x_{\kappa:K}$ given $\underline{x}, \tau_{\kappa:K}$
  - solve for timing $\tau_{\kappa:K}, v_{\kappa:K}$ given $x_{\kappa:K}$
  - solve for path ξ^H given $\tau_{\kappa:K}, x_{\kappa:K}, v_{\kappa:K}$

But also:

  - Maintain which constraints $\kappa : K$ are still ahead
  - Loosing a constraint, $\|\bar\phi_\kappa(\underline{x})\| > \bar\theta$, $\quad\Rightarrow\quad$ backtrack $\kappa \leftarrow \kappa - 1$

MPC cycle

- In each cycle ($\sim$10Hz in experiments)
  - solve for sequence of waypoints $x_{\kappa:K}$ given $\underline{x}, \tau_{\kappa:K}$
  - solve for timing $\tau_{\kappa:K}, v_{\kappa:K}$ given $x_{\kappa:K}$
  - solve for path ξ^H given $\tau_{\kappa:K}, x_{\kappa:K}, v_{\kappa:K}$

  But also:
  - Maintain which constraints $\kappa : K$ are still ahead
  - Loosing a constraint, $\|\bar{\phi}_\kappa(\underline{x})\| > \bar{\theta}$, $\quad \Rightarrow \quad$ backtrack $\kappa \leftarrow \kappa - 1$

- Could be viewed as continuous TAMP replanning, but only “within skeleton”

More details in the paper

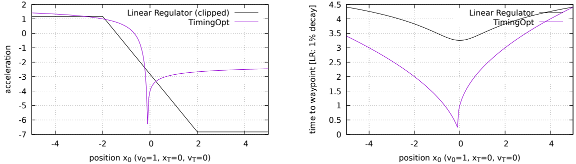
- Alternating MPC steps $\sim$ coordinate descent, but sub-problems are **approximations**!

More details in the paper

- Alternating MPC steps $\sim$ coordinate descent, but sub-problems are **approximations**!
- **Temporal** (Bellman) **Consistency** of timing MPC:
  "Sanity check theory": *Iff* there were no perturbations, opt. timings in two consecutive MPC steps are consistent; stationarity of control except for phase variable (backtracking).

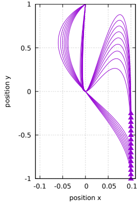
More details in the paper

- Alternating MPC steps $\sim$ coordinate descent, but sub-problems are **approximations**!
- **Temporal** (Bellman) **Consistency** of timing MPC:
  "Sanity check theory": *Iff* there were no perturbations, opt. timings in two consecutive MPC steps are consistent; stationarity of control except for phase variable (backtracking).
- **Comparison of gains/convergence with Linear Regulator**
  Timing optimization implies interesting gain profile – cp. to (clipped) linear regulator: much more explicit convergence within *finite* time with low gains



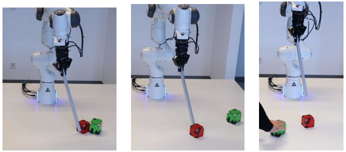
More details in the paper

- Alternating MPC steps $\sim$ coordinate descent, but sub-problems are **approximations**!
- **Temporal** (Bellman) **Consistency** of timing MPC:
  "Sanity check theory": *Iff* there were no perturbations, opt. timings in two consecutive MPC steps are consistent; stationarity of control except for phase variable (backtracking).
- **Comparison of gains/convergence with Linear Regulator**
  Timing optimization implies interesting gain profile – cp. to (clipped) linear regulator: much more explicit convergence within *finite* time with low gains
- **Time-of-no-return when approaching a waypoint**
  You can never thread an infinitesimal needle – you will always slighly miss noisy constraints $\rightarrow$ allow for a margin, or follow no-abort policy if constraint is very soon



Demonstrations

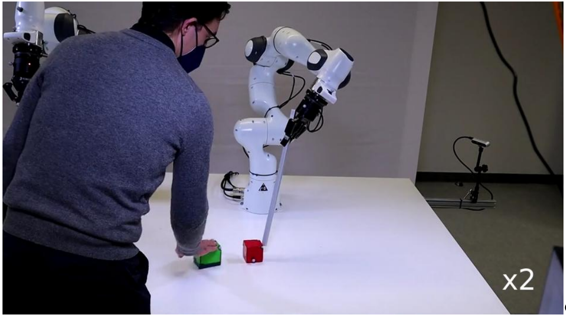
- Pushing scenario:



Sequence of four constraints $\phi_{1:4}$

  - $\hat\phi_4$: red & green touch
  - $\hat\phi_3$: stick touches (into) red *opposite* to the final red pose
  - $\hat\phi_2, \hat\phi_1$: approach *opposite* to final red pose

- Pick-and-place scenario
- Drone-through-gates scenario



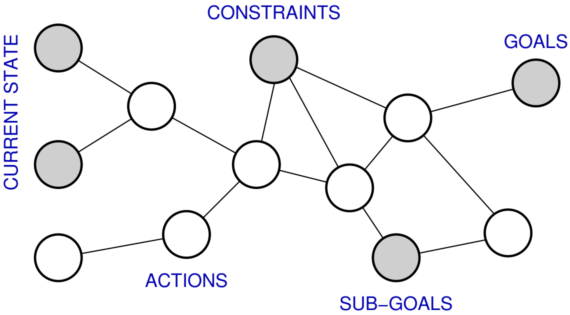
- All looping behavior is implicit!
  - running constraints missed $\to$ backtrack
  - final waypoint constraint lost $\to$ backtrack
- Code & videos:

`https://www.user.tu-berlin.de/mtoussai/22-SecMPC/`

Action Representation – Discussion

- Higher-level decisions concern *sequence* of constraints
- The controls u also depend on future high-level decisions
  - Cp. to skill $\pi_i(u|x)$, where u only depends on current high-level decision i
  - Skills/options/hierRL are strictly temporally hierarchical (sMDP) – *SecMPC is not!*

Side note: Planning-as-Inference



We condition on future goals/constraints and infer actions/motion.

Probabilistic inference for solving discrete and continuous state Markov Decision Processes, Toussaint & Storkey. ICML'06

Planning as probabilistic inference, Botvinick & Toussaint. Trends in CogSci 2012

Scalable Multiagent Planning Using Probabilistic Inference, Kumar, Zilberstein & Toussaint. IJCAI'11 & JAIR 2015

On stochastic optimal control and reinforcement learning by approximate, Rawlik, Toussaint & Vijayakumar. R:SS'12

Action Representation – Discussion

- Is that a promising action representation also for learning systems? LLMs?
- Inherits strong generalization and compositionality of MPC

- "ϕ_i are easier to learn" ?
  - Skill learning $\to$ Constraint learning

Scene Representation

- What is *input* to higher-level decision making *and* control?

- What is *input* to higher-level decision making *and* control?

- Standard TAMP: full information setting
  "full-information setting" (aka. "privileged setting"): full model knowledge, full state information, no uncertainty, not need for perception or control

- What is *input* to higher-level decision making *and* control?

- Standard TAMP: full information setting
  "full-information setting" (aka. "privileged setting"): full model knowledge, full state information, no uncertainty, not need for perception or control

- SayCan and similar: "only" textual scene description as input $\rightarrow$ missing out direct geometric information

- What is *input* to higher-level decision making *and* control?

- Standard TAMP: full information setting
  “full-information setting” (aka. “privileged setting”): full model knowledge, full state information, no uncertainty, not need for perception or control

- SayCan and similar: “only” textual scene description as input $\to$ missing out direct geometric information

How exploit structure of 3D space, objects, physics?

Field Representations of Objects

- Represent object i as image-conditional field $y_i(x) \in \mathbb{R}^d$ in 3D space
  - e.g. Neural Descriptor Fields, NeRFs, SDF, *Pixel-Aligned Implicit Functional Objects*
  - Captures locality in 3D space, out-projects affordances into space, smooth gradients

Field Representations of Objects

- Represent object i as image-conditional field $y_i(x) \in \mathbb{R}^d$ in 3D space
  - e.g. Neural Descriptor Fields, NeRFs, SDF, *Pixel-Aligned Implicit Functional Objects*
  - Captures locality in 3D space, out-projects affordances into space, smooth gradients
- Two instances:
  - Field representations to learn **task constraints** (grasping, hanging)

    Deep Visual Constraints: Neural Implicit Models for Manipulation Planning from Visual Input, Jung-Su Ha, Danny Driess, Marc Toussaint. arXiv:2112.04812, RAL 2021
  - Field representations to learn **dynamics** (dynamics of pushing & deformation)

    Learning Multi-Object Dynamics with Compositional Neural Radiance Fields, Danny Driess, Zhiao Huang, Yunzhu Li, Russ Tedrake, Marc Toussaint. arXiv:2202.11855, CoRL'22

Danny Driess

Jung-Su Ha

Field Representations of Objects

"High-level decisions are *functions* ϕ passed to a lower-level control loop"

Neural field representations $y_i(x) \;\mapsto\;$ neural constraint functions $\phi_{1:K}$

Pixel-Aligned Implicit Functional Objects (PIFO)

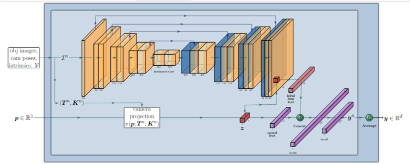


Fig. 3: PIFO (i) encodes the images $\mathcal{I}$ as pixel-wise feature images $\mathcal{F}$ via U-net, (ii) projects the query point $p \in \mathbb{R}^3$ into the pixel coordinate $z \in \mathbb{R}^3$ using known camera geometry, and (iii) computes the object representation vector $y \in \mathbb{R}^d$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

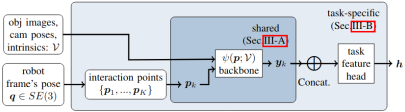


Fig. 2: The interaction feature prediction scheme of DVC

Deep Visual Constraints: Neural Implicit Models for Manipulation Planning from Visual Input, Jung-Su Ha, Danny Driess, Marc Toussaint. arXiv:2112.04812, RAL 2021

- We have V camera views, images $\mathcal{I} = \{(I^1, K^1), ..., (I^V, K^V)\}$, want to learn function

$$\phi(x; \mathcal{I})$$

Pixel-Aligned Implicit Functional Objects (PIFO)

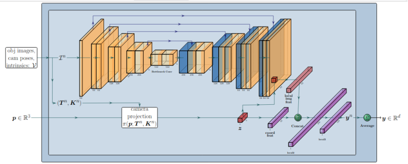


Fig. 3: PIFO (i) encodes the images $\mathcal{I}$ as pixel-wise feature images $\mathcal{F}$ via U-net, (ii) projects the query point $p \in \mathbb{R}^3$ into the pixel coordinate $z \in \mathbb{R}^3$ using known camera geometry, and (iii) computes the object representation vector $y \in \mathbb{R}^d$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

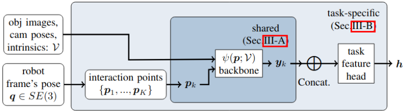


Fig. 2: The interaction feature prediction scheme of DVC

Deep Visual Constraints: Neural Implicit Models for Manipulation Planning from Visual Input, Jung-Su Ha, Danny Driess, Marc Toussaint. arXiv:2112.04812, RAL 2021

- We have V camera views, images $\mathcal{I} = \{(I^1, K^1), ..., (I^V, K^V)\}$, want to learn function

$$\phi(x; \mathcal{I})$$

- We first train a d-dimensional **field** (*"Pixel-Aligned Implicit Functional Object"*), $x \in \mathbb{R}^3$,

$$y(x; \mathcal{I}) = \frac{1}{V} \sum_i \text{MLP}(\text{UNet}(I^i, K^i(x)), K^i(x))$$

Pixel-Aligned Implicit Functional Objects (PIFO)

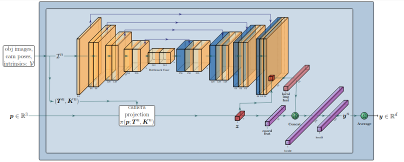


Fig. 3: PIFO (i) encodes the images $\mathcal{I}$ as pixel-wise feature images $\mathcal{F}$ via U-net, (ii) projects the query point $\boldsymbol{p} \in \mathbb{R}^3$ into the pixel coordinate $\boldsymbol{z} \in \mathbb{R}^3$ using known camera geometry, and (iii) computes the object representation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

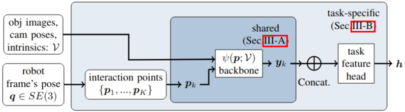


Fig. 2: The interaction feature prediction scheme of DVC

Deep Visual Constraints: Neural Implicit Models for Manipulation Planning from Visual Input, Jung-Su Ha, Danny Driess, Marc Toussaint. arXiv:2112.04812, RAL 2021

- We have V camera views, images $\mathcal{I} = \{(I^1, K^1), ..., (I^V, K^V)\}$, want to learn function

$$\phi(x; \mathcal{I})$$

- We first train a d-dimensional **field** (*"Pixel-Aligned Implicit Functional Object"*), $x \in \mathbb{R}^3$,

$$y(x; \mathcal{I}) = \frac{1}{V} \sum_i \text{MLP}(\text{UNet}(I^i, K^i(x)), K^i(x))$$

- Field is queried at finite set of *interaction points* $x_1, .., x_K$ to get the feature

$$\phi(x) = \text{MLP}(y_1(x; \mathcal{I}), .., y_K(x; \mathcal{I}))$$

Training Objectives: Distance Decoding & Task Constraints

- Field representation is trained for SDF decoding – cheap supervised data

Training Objectives: Distance Decoding & Task Constraints

- Field representation is trained for SDF decoding – cheap supervised data
- Constraint heads ϕ_i are trained for prediction of stable hanging, succesful grasping
  - More expensive supervision data: empirical success in simulation
  - Random grasp / hanging configurations in simulation $\rightarrow$ run $\rightarrow$ evaluate success

$\rightarrow$ *supervised* learning of ϕ_i

Training Objectives: Distance Decoding & Task Constraints

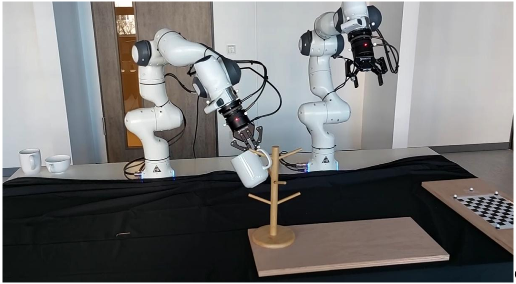
- Field representation is trained for SDF decoding – cheap supervised data
- Constraint heads ϕ_i are trained for prediction of stable hanging, succesful grasping
  - More expensive supervision data: empirical success in simulation
  - Random grasp / hanging configurations in simulation $\to$ run $\to$ evaluate success

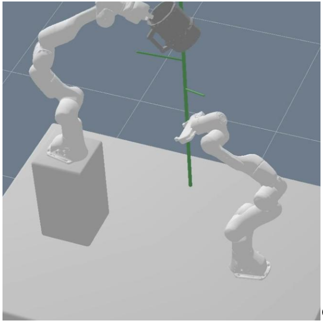
  $\to$ *supervised* learning of ϕ_i

"Skill learning $\to$ Constraint learning" (cp. reward learning, invRL)

Execution with Learned Constraints

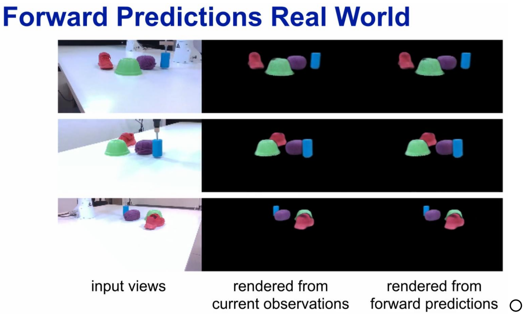
(No search over skeletons, no reactive MPC, just optimal path for given sequence of constraints.)





Deep Visual Constraints: Neural Implicit Models for Manipulation Planning from Visual Input, Jung-Su Ha, Danny Driess, Marc Toussaint. arXiv:2112.04812, RAL 2021

Image-Based Multi-Object Dynamics



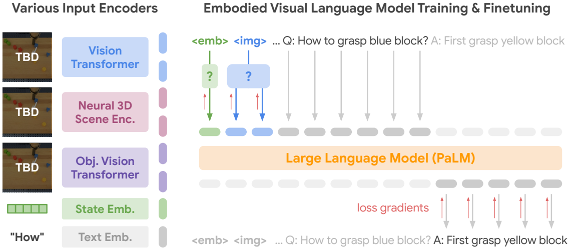
https://dannydriess.github.io/compnerfdyn/

Learning Multi-Object Dynamics with Compositional Neural Radiance Fields, Danny Driess, Zhiao Huang, Yunzhu Li, Russ Tedrake, Marc Toussaint. arXiv:2202.11855, CoRL'22

- Similar in spririt, but learn multi-object dynamics (also deformable)
  - Cheap supervision of representation learning via compositional NeRFs
  - Training GNNs on latent NeRF encoding to predict dynamics

Embodied Multimodal Language Model

- Multi-modal sentences (interleaved text & images & state) as input to LLMs



on arxiv today...

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

Discussion

Discussion

"Reasoning as decoding of an implicited representation of behavior"

Discussion

"Reasoning as decoding of an implicited representation of behavior"

- Do we need reasoning? (TAMP vs. LLM)
  - TAMP: Understand structure of sequential manipulation
  - TAMP: Impressive *generalization* – target for what we should achieve in real, BUT full-information planning

Discussion

"Reasoning as decoding of an implicited representation of behavior"

- Do we need reasoning? (TAMP vs. LLM)
  - TAMP: Understand structure of sequential manipulation
  - TAMP: Impressive *generalization* – target for what we should achieve in real, BUT full-information planning
  - LLM: exploit massive data, sensor-based high-level reasoning
  - LLMs as heuristic for TAMP? Or TAMP as data generator for LLM?

Discussion

"Reasoning as decoding of an implicited representation of behavior"

- Do we need reasoning? (TAMP vs. LLM)
  - TAMP: Understand structure of sequential manipulation
  - TAMP: Impressive *generalization* – target for what we should achieve in real, BUT full-information planning
  - LLM: exploit massive data, sensor-based high-level reasoning
  - LLMs as heuristic for TAMP? Or TAMP as data generator for LLM?

- Lower level:
  - Implicit Action Representations, e.g., MPC
  - Supervised (constraint/reward learning, BC) over RL?

Thanks

for your attention!

- Team:

Danny Driess

Jung-Su Ha

Jason Harris

Quim Ortiz de Haro

Svetlana Levit

Valentin Hartmann

Andreas Orthey

Ingmar Schubert

Wolfgang Hönig

- Funding: This work was supported by the DFG excellence cluster **Science of Intelligence, Berlin**, the **IMPRS** (MPI), and as well as **IntCDC, Stuttgart**.