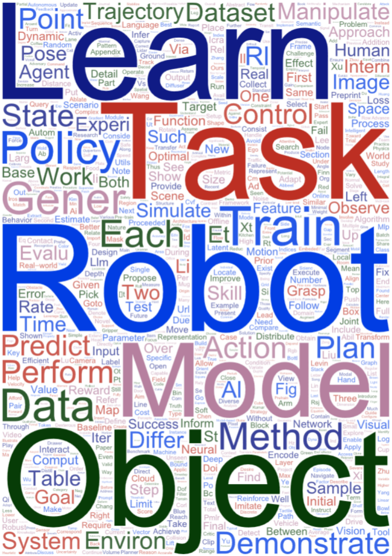


Robot Learning

Imitation Learning

Learning from Demonstration, Behavior Cloning, Direct (Interactive) Policy Learning, Traj. Dist. Learning, Constraint Learning, (excluded: Inv. RL)

Marc Toussaint
Technical University of Berlin
Summer 2024

General Idea

- Given expert demonstration data $D = \{(x^i_{1:T_i}, u^i_{1:T_i})\}_{i=1}^n$

$$\begin{aligned} i &: \quad \text{episode/demonstration} \\ x^i_{1:T_i} &: \quad i\text{th state trajectory} \\ u^i_{1:T_i} &: \quad i\text{th control trajectory} \end{aligned}$$

  without external rewards/objectives/costs defined
  → extract the “relevant information/model/policy” to reproduce demonstrations

General Idea

- Given expert demonstration data $D = \{(x^i_{1:T_i}, u^i_{1:T_i})\}_{i=1}^n$

$$\begin{aligned} i &: \quad \text{episode/demonstration} \\ x^i_{1:T_i} &: \quad i\text{th state trajectory} \\ u^i_{1:T_i} &: \quad i\text{th control trajectory} \end{aligned}$$

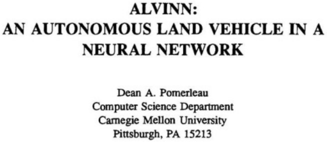
  without external rewards/objectives/costs defined
  $\to$ extract the “relevant information/model/policy” to reproduce demonstrations

- Reproducing could mean various things
  - Move along similar trajectories (e.g. imitate a gesture)
  - Reproduce the *effect* of the demonstration (manipulation, flight maneuver, no traffic collisions)

Early Work

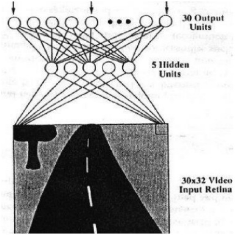
Deep Imitation Learning in 1989

- ❑ A CMU paper!
  - CMU has incubated many self-driving companies



**ALVINN:
AN AUTONOMOUS LAND VEHICLE IN A
NEURAL NETWORK**

Dean A. Pomerleau
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213



(Shi's lecture 5)

https://www.youtube.com/watch?v=ntIczNQKfjQ

Early Work

- Behavior Cloning (later called so):

  Dean A. Pomerleau, (1988). Alvinn: An autonomous land vehicle in a neural network.
  Advances in neural information processing systems, 1

- Early review paper:

  Stefan Schaal, Auke Ijspeert, and Aude Billard, (2003). Computational approaches to motor learning by imitation.
  Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences, 358(1431):537–547

  [clarifies direct policy learning (BC) vs. trajectory imitation (and auto-control); mentiones work from the 60ies, but esp. 90ies]

- Early work named **Learning from Demonstration** (or Programming by Demonstration)

  Christopher G. Atkeson and Stefan Schaal, (1997). Robot learning from demonstration.
  In *ICML*, volume 97, pages 12–20

  [Idea: Avoid explicit programming → teach by demonstration. See also entries in "Handbook of Robotics"...]

- Another early survey:

  Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning, (2009). A survey of robot learning from demonstration.
  Robotics and autonomous systems, 57(5):469–483

  [Distinguishes 3 kinds: behavior cloning, use data to learn dynamics (system identification), learn plans (nowadays uncommon)]

Outline

- Types of Imitation Learning
  - Behavior Cloning
  - Trajectory Distribution Learning (& Constraint Learning)
  - Direct (Interactive) Policy Learning
  - Inverse Reinforcement Learning (not covered today)

Outline

- Types of Imitation Learning
  - Behavior Cloning
  - Trajectory Distribution Learning (& Constraint Learning)
  - Direct (Interactive) Policy Learning
  - Inverse Reinforcement Learning (not covered today)

- Data Generation
  - Distributional (domain) shift, “compound errors” in imitation, on-/off-policy
  - Data augmentation or interactive data aggregation
  - Collection techniques: Tele-Operation, Kinesthetic Teaching, Human Demonstrations

Behavior Cloning

- Formulate Imitation Learning literally as *Supervised ML*
- Given data $D = \{(x^i_{1:T_i}, u^i_{1:T_i})\}_{i=1}^n$, find

$$\min_\theta \sum_{i,t} \ell(u^i_t, \pi_\theta(x^i_t)) \ , \tag{1}$$

where $\pi_\theta : x \mapsto u$ is a deterministic policy (e.g. NN) mapping states to controls

Behavior Cloning

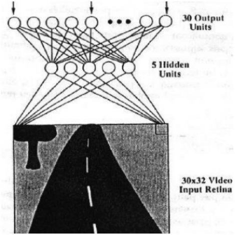
Deep Imitation Learning in 1989

❑ A CMU paper!

- CMU has incubated many self-driving companies

ALVINN:
AN AUTONOMOUS LAND VEHICLE IN A NEURAL NETWORK

Dean A. Pomerleau
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213



(Shi's lecture 5)

Behavior Cloning

- Behavior Cloning literally imitates the demonstrated mapping $x \mapsto u$

Behavior Cloning

- Behavior Cloning literally imitates the demonstrated mapping $x \mapsto u$

- Issues:
  - But does that also imitate the *long term behavior* or *eventual effect* of the demonstrations? (Ignores distributional shift.)
  - Does it capture the "essence" of what is demonstrated?
  - Can it deal with multi-modal demonstrations? (→ next week: multi-modal policies)

Trajectory Distribution Learning

[This is not common terminology, and seemingly skipped in other Imitation Learning lectures – unfortunately. I think this captures an essence of the problem.]

- What does it mean to capture the “essence” of data?

Trajectory Distribution Learning

[This is not common terminology, and seemingly skipped in other Imitation Learning lectures – unfortunately. I think this captures an essence of the problem.]

- What does it mean to capture the “essence” of data?
  - Learn a *distribution model* $p_\theta(x_{1:T})$ of demonstrated trajectories!

$$\max_\theta \prod_i p_\theta(x^i_{1:T_i}) \quad \text{(likelihood maximization (LM))} , \tag{2}$$

where p_θ is some model class powerful enough to represent “essence”

Trajectory Distribution Learning

[This is not common terminology, and seemingly skipped in other Imitation Learning lectures – unfortunately. I think this captures an essence of the problem.]

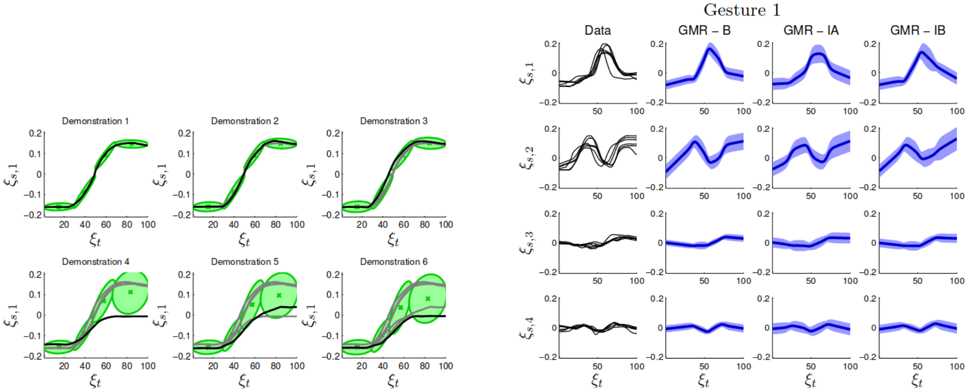
- What does it mean to capture the "essence" of data?
  - Learn a *distribution model* $p_\theta(x_{1:T})$ of demonstrated trajectories!

$$\max_\theta \prod_i p_\theta(x^i_{1:T_i}) \quad \text{(likelihood maximization (LM))} , \tag{2}$$

  where p_θ is some model class powerful enough to represent "essence"

- What are "powerful" models?
  - Transformer models, diffusion models
  - But we'll start with very basic Gaussian models
  - ...and discuss models specifically for robotic manipulation

Trajectory Distribution Learning: GMMs



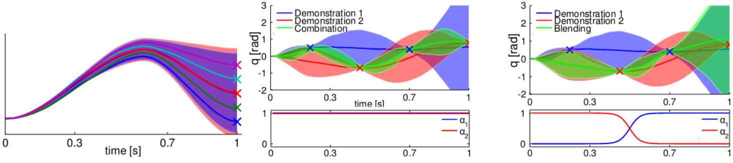
Sylvain Calinon and Aude Billard, (2007). Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, pages 255–262

- Embed trajectories $x_{1:T}$ in "space-time" $\{(t, x_t)\}_{t=1}^T$
- Fit a density estimator to $p(t, x_t)$ (easiest: Gaussian Mixture Model (GMM), LM well studied)
- Can be translated to control policy by reading out conditional $p(x|t)$ and using inverse dynamics

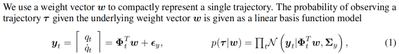
Trajectory Distribution Learning: GMMs

- A simple way to describe the distribution of demonstrated trajectories
- Variance of learned $p(x|t)$ captures “consistent bottlenecks” in demonstrations
  [Is that a key structure in demonstrations? Search also “Calinon constraints”]
- Can be combined with Dynamic Time Warping to temporally align demonstrations
- GMM approach is around for ~ 20 years

Trajectory Distribution Learning: ProMPs



(a) Conditioning (b) Combination (c) Blending



We use a weight vector $\boldsymbol{w}$ to compactly represent a single trajectory. The probability of observing a trajectory $\boldsymbol{\tau}$ given the underlying weight vector $\boldsymbol{w}$ is given as a linear basis function model

$$\boldsymbol{y}_t = \begin{bmatrix} q_t \\ \dot{q}_t \end{bmatrix} = \boldsymbol{\Phi}_t^T \boldsymbol{w} + \boldsymbol{\epsilon}_y, \qquad p(\boldsymbol{\tau}|\boldsymbol{w}) = \prod_t \mathcal{N}\left(\boldsymbol{y}_t | \boldsymbol{\Phi}_t^T \boldsymbol{w}, \boldsymbol{\Sigma}_y\right), \qquad (1)$$

Alexandros Paraschos, Christian Daniel, Jan R. Peters, and Gerhard Neumann, (2013). Probabilistic movement primitives. *Advances in neural information processing systems*, 26

- Nothing but (prob.) linear regression $t \mapsto x_t$ with basis function features (LM↔regression)
- Very simple distribution model over trajectories [could use GPs to kernelize]
- Related to Inference Control (AICO, ICML'09), Path Integral methods (RSS'12)
- Great flexibility to condition, compose, and blend
- Somewhat superseeds earlier work on learning movement primitives from demonstration

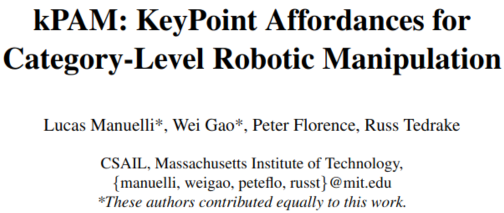
  [typically Dynamic Movement Primitives (DMPs, Schaal et al'03)]

Trajectory Distribution Learning: Features & Constraints

- Think about Manipulation!

Trajectory Distribution Learning: Features & Constraints

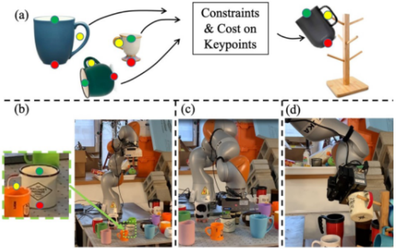
- Think about Manipulation!



kPAM: KeyPoint Affordances for Category-Level Robotic Manipulation

Lucas Manuelli*, Wei Gao*, Peter Florence, Russ Tedrake

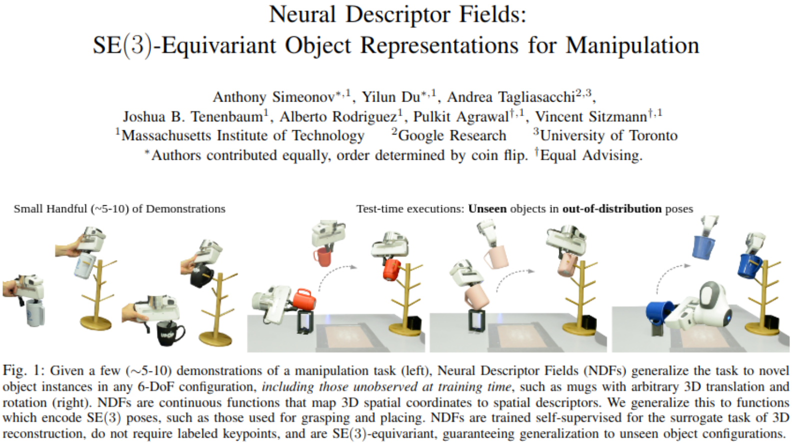
CSAIL, Massachusetts Institute of Technology,
{manuelli, weigao, peteflo, russt}@mit.edu
*These authors contributed equally to this work.



Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake, (2022). KPAM: KeyPoint Affordances for Category-Level Robotic Manipulation.
In Tamim Asfour, Eiichi Yoshida, Jaeheung Park, Henrik Christensen, and Oussama Khatib, editors, *Robotics Research*, volume 20, pages 132–157

Trajectory Distribution Learning: Features & Constraints

- Think about Manipulation!



Neural Descriptor Fields:
SE(3)-Equivariant Object Representations for Manipulation

Anthony Simeonov[*,1], Yilun Du[*,1], Andrea Tagliasacchi[2,3],
Joshua B. Tenenbaum[1], Alberto Rodriguez[1], Pulkit Agrawal[†,1], Vincent Sitzmann[†,1]
[1]Massachusetts Institute of Technology [2]Google Research [3]University of Toronto
[*]Authors contributed equally, order determined by coin flip. [†]Equal Advising.

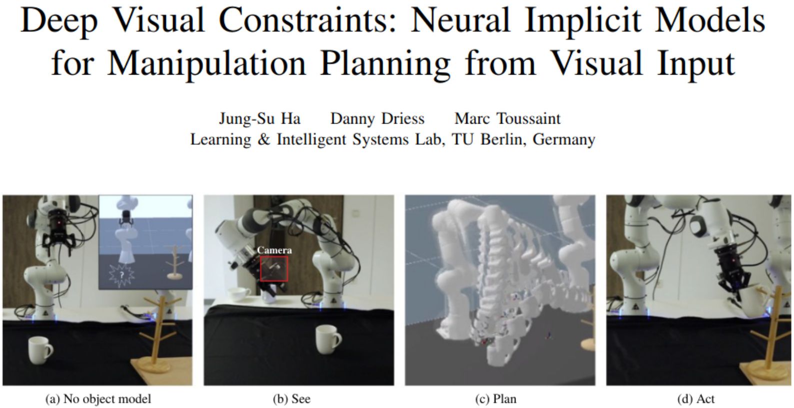
Small Handful (~5-10) of Demonstrations
Test-time executions: **Unseen** objects in **out-of-distribution** poses

Fig. 1: Given a few (~5-10) demonstrations of a manipulation task (left), Neural Descriptor Fields (NDFs) generalize the task to novel object instances in any 6-DoF configuration, *including those unobserved at training time*, such as mugs with arbitrary 3D translation and rotation (right). NDFs are continuous functions that map 3D spatial coordinates to spatial descriptors. We generalize this to functions which encode SE(3) poses, such as those used for grasping and placing. NDFs are trained self-supervised for the surrogate task of 3D reconstruction, do not require labeled keypoints, and are SE(3)-equivariant, guaranteeing generalization to unseen object configurations.

Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B. Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann, (2022). Neural descriptor fields: Se (3)-equivariant object representations for manipulation.
In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400

Trajectory Distribution Learning: Features & Constraints

- Think about Manipulation!



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

Jung-Su Ha, Danny Driess, and Marc Toussaint, (2022). Deep visual constraints: Neural implicit models for manipulation planning from visual input. *IEEE Robotics and Automation Letters*, 7(4):10857–10864

Trajectory Distribution Learning: Features & Constraints

- Connects to large body of literature:
  - More examples: FlowBot3D, UMPNet, Bi-KVIL, ”Waypoint-based imitation learning”, ..

Trajectory Distribution Learning: Features & Constraints

- Connects to large body of literature:
  - More examples: FlowBot3D, UMPNet, Bi-KVIL, ”Waypoint-based imitation learning”, ..
  - Human Activity Modelling, Action Segmentation:

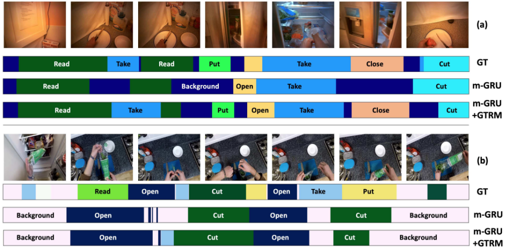


Figure 4. Qualitative comparison of results for action segmentation task on (a) EGTEA, and (b) EPIC dataset. Only part of the whole video is shown for clarity. We can see in (a) that the *take*, *put* and *close* actions are correctly detected by adding GTRM.

Trajectory Distribution Learning: Features & Constraints

- Connects to large body of literature:
  - More examples: FlowBot3D, UMPNet, Bi-KVIL, ”Waypoint-based imitation learning”, ..
  - Human Activity Modelling, Action Segmentation:

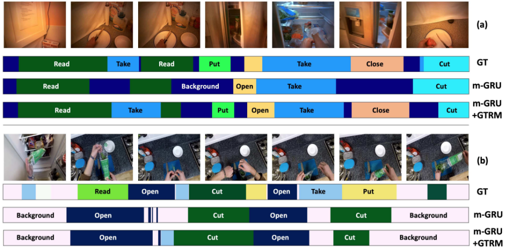


Figure 4. Qualitative comparison of results for action segmentation task on (a) EGTEA, and (b) EPIC dataset. Only part of the whole video is shown for clarity. We can see in (a) that the *take*, *put* and *close* actions are correctly detected by adding GTRM.

- What really is the essence to extract from demonstrations?

- Back to Behavior Cloning...

- Back to Behavior Cloning...

- Issues:
  - But does that also imitate the *long term behavior* or *eventual effect* of the demonstrations? **(Ignores distributional shift.)**
  - Does it capture the “essence” of what is demonstrated?

Distributional (Domain) Shift

Distributional (Domain) Shift

- Standard ML: $x, y \sim p(x,y)$ **i.i.d.**; same p for trains & test

Distributional (Domain) Shift

- Standard ML: $x, y \sim p(x, y)$ **i.i.d.**; same p for trains & test

- Sequential Decision Processes: own policy π influences test distrib. $p_\pi(x_t)$!

Distributional (Domain) Shift

- Standard ML: $x, y \sim p(x, y)$ **i.i.d.**; same p for trains & test

- Sequential Decision Processes: own policy π influences test distrib. $p_\pi(x_t)$!
  - Fundamental difference between learning in sequential decision processes and Supervised ML!
  - Also in off-policy & offline RL: We *train* a policy (or Q, V-function) with losses relative to $p_{\pi_\beta}(x_t)$ with *behavior policy* (π_β)
  - Generally called distributional shift, or Out-of-Distribution (OOD) testing

Distributional Shift in Behavior Cloning

- When we train policy π_θ in BC, we minimize

$$\min_\theta \sum_{i,t} \ell(u_t^i, \pi_\theta(x_t^i)) \;\leftrightarrow\; \min_\theta \mathbb{E}_{\pi^*}\{\ell(u, \pi_\theta(x))\} \tag{3}$$

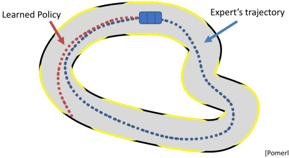
  but when using the policy, we generate fully different distribution

Distributional Shift in Behavior Cloning

- When we train policy π_θ in BC, we minimize

$$\min_\theta \sum_{i,t} \ell(u_t^i, \pi_\theta(x_t^i)) \;\leftrightarrow\; \min_\theta \mathbb{E}_{\pi^*}\{\ell(u, \pi_\theta(x))\} \tag{3}$$

  but when using the policy, we generate fully different distribution



[Pomerleau89,Daume09]

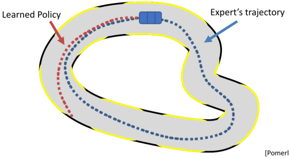
Also called **Compound Error** (Shi's lecture 5)

Distributional Shift in Behavior Cloning

- When we train policy π_θ in BC, we minimize

$$\min_\theta \sum_{i,t} \ell(u_t^i, \pi_\theta(x_t^i)) \;\leftrightarrow\; \min_\theta \mathbb{E}_{\pi^*}\{\ell(u, \pi_\theta(x))\} \tag{3}$$

  but when using the policy, we generate fully different distribution



[Pomerleau89,Daume09]

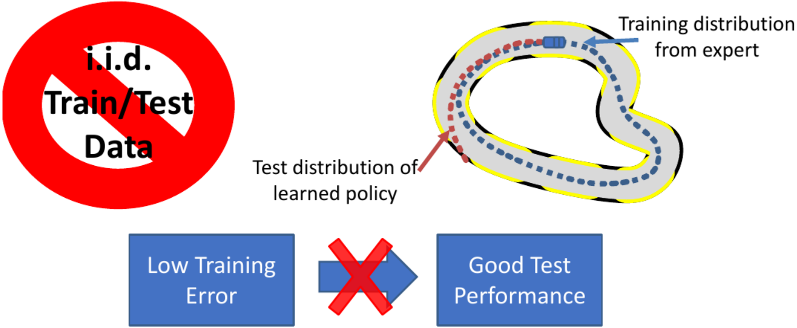
Also called **Compound Error** (Shi's lecture 5)

- What we should train is this:!

$$\min_\theta \mathbb{E}_{\pi_\theta}\{\ell(\pi^*(x), \pi_\theta(x))\} \tag{4}$$

Distributional Shift in Behavior Cloning

- BC formulates a supervised ML problem, but in view of testing, it is not:



(Shi's lecture 5)

How address the Distributional Shift?

How address the Distributional Shift?

- Ensure the data better covers the eventual $p_\pi(x_t)$ of trained π

How address the Distributional Shift?

- Ensure the data better covers the eventual $p_\pi(x_t)$ of trained π
  - Enforce the expert to demonstrate also for non-optimal states (cover also non-expert situations)
  - Collect data interactively at exactly the states visited by π (DAgger)

Enforcing wider expert demonstrations

- Occasionally perturb the expert! Add noise!

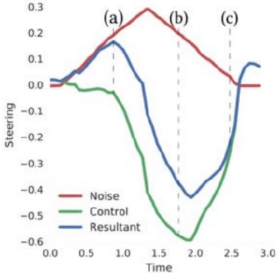
**ALVINN:
AN AUTONOMOUS LAND VEHICLE IN A
NEURAL NETWORK**

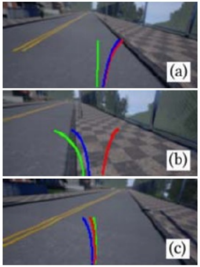
Dean A. Pomerleau
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

> “...the network must not solely be shown examples of accurate driving, but also how to recover (i.e. return to the road center) once a mistake has been made.”

End-to-end Driving via Conditional Imitation Learning

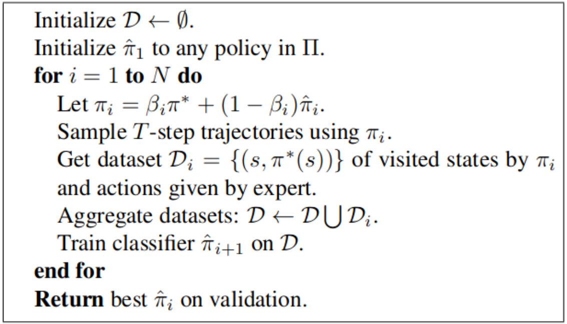
Felipe Codevilla[1,2] Matthias Müller[1,3] Antonio López[2] Vladlen Koltun[1] Alexey Dosovitskiy[1]





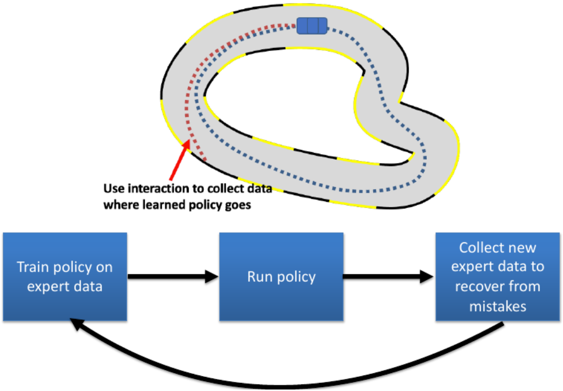
(Shi’s lecture 5)

DAgger



```
Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in $\Pi$.
for $i = 1$ to $N$ do
    Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
    Sample $T$-step trajectories using $\pi_i$.
    Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by $\pi_i$
    and actions given by expert.
    Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
    Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.
```

Algorithm 3.1: DAGGER Algorithm.



Use interaction to collect data where learned policy goes

Train policy on expert data

Run policy

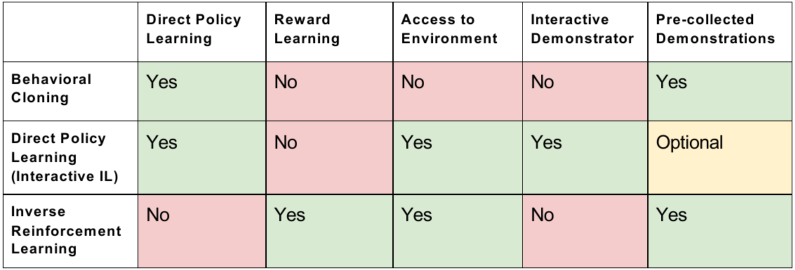
Collect new expert data to recover from mistakes

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell, (2011). A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning

`https://www.youtube.com/watch?v=V00npNnWzSU`

- This repeatedly collects data from the current π, to approximate $\min_\theta \mathbb{E}_\pi\{\ell(\pi^*(x_t), \pi_\theta(x_t))\}$

- From Yue's ICML'18 tutorial:



	Direct Policy Learning	Reward Learning	Access to Environment	Interactive Demonstrator	Pre-collected Demonstrations
Behavioral Cloning	Yes	No	No	No	Yes
Direct Policy Learning (Interactive IL)	Yes	No	Yes	Yes	Optional
Inverse Reinforcement Learning	No	Yes	Yes	No	Yes

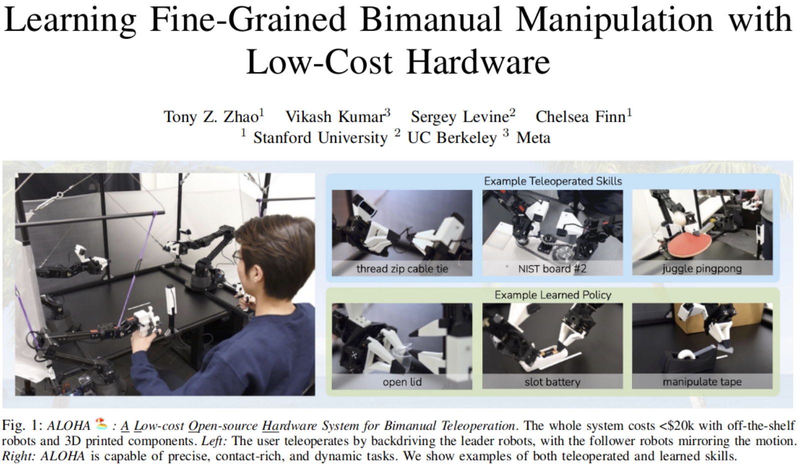
- Crucial point: For DAgger we have a very different setting: Access to the environment (testing rollouts), interactively querying the expert.

Data Collection

Data Collection

- We’ve covered the theoretical aspect concerning distributional shift
- Data source:
  - Tele-Operation
  - Kinesthetic Teaching
  - Human Demonstrations & Motion Capture
  - Videos Only

Tele-Operation: Aloha



Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware

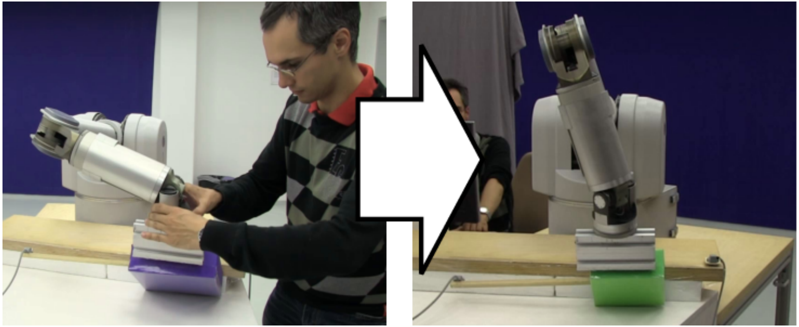
Tony Z. Zhao[1] Vikash Kumar[3] Sergey Levine[2] Chelsea Finn[1]
[1] Stanford University [2] UC Berkeley [3] Meta

Fig. 1: *ALOHA : A Low-cost Open-source Hardware System for Bimanual Teleoperation*. The whole system costs <$20k with off-the-shelf robots and 3D printed components. *Left:* The user teleoperates by backdriving the leader robots, with the follower robots mirroring the motion. *Right: ALOHA* is capable of precise, contact-rich, and dynamic tasks. We show examples of both teleoperated and learned skills.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn, (2023). Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware

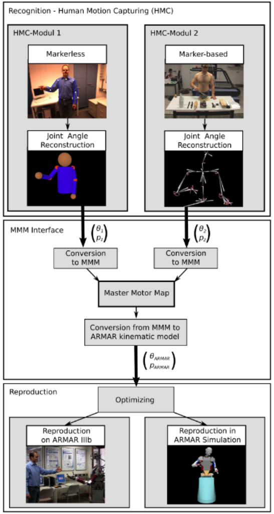
`https://tonyzhaozh.github.io/aloha/`

Kinesthetic Teaching



Learning movement primitives for force interaction tasks (Kober et al'15)

Human Demonstrations & Motion Capture


Recognition - Human Motion Capturing (HMC)
HMC-Modul 1
Markerless
Joint Angle Reconstruction
HMC-Modul 2
Marker-based
Joint Angle Reconstruction
MMM Interface
$\begin{pmatrix}\theta_1\\ p_1\end{pmatrix}$
$\begin{pmatrix}\theta_2\\ p_2\end{pmatrix}$
Conversion to MMM
Conversion to MMM
Master Motor Map
Conversion from MMM to ARMAR kinematic model
$\begin{pmatrix}\theta_{ARMAR}\\ p_{ARMAR}\end{pmatrix}$
Reproduction
Optimizing
Reproduction on ARMAR IIIb
Reproduction in ARMAR Simulation

Martin Do, Pedram Azad, Tamim Asfour, and Rudiger Dillmann, (2008). Imitation of human motion on a humanoid robot using non-linear optimization.

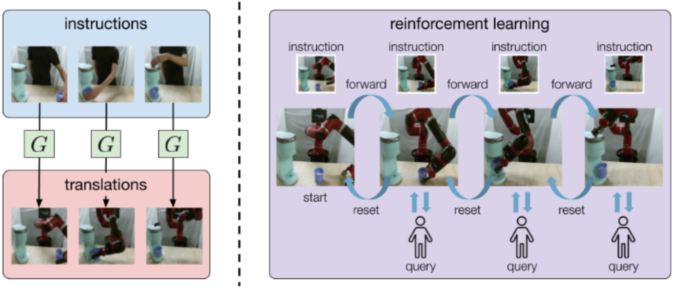
In *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*, pages 545–

552

Human Demonstrations From Video Only

AVID: Learning Multi-Stage Tasks via Pixel-Level Translation of Human Videos

Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine
Berkeley Artificial Intelligence Research, Berkeley, CA, 94720
Email: smithlaura@berkeley.edu



Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine, (2020). AVID: Learning Multi-Stage Tasks via Pixel-Level Translation of Human Videos

- This whole lecture talked about states! Same for observations y_t only!
  - History-input policies (analogous to autoregressive dynamics)
  - Recursive (RNN) policies (analogous to recursive dynamics)
  - Transformer policies (sequence models)

[1] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning, (2009).
A survey of robot learning from demonstration.
Robotics and autonomous systems, 57(5):469–483.

[2] Christopher G. Atkeson and Stefan Schaal, (1997).
Robot learning from demonstration.
In *ICML*, volume 97, pages 12–20.

[3] Sylvain Calinon and Aude Billard, (2007).
Incremental learning of gestures by imitation in a humanoid robot.
In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, pages 255–262.

[4] Martin Do, Pedram Azad, Tamim Asfour, and Rudiger Dillmann, (2008).
Imitation of human motion on a humanoid robot using non-linear optimization.
In *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*, pages 545–552.

[5] Jung-Su Ha, Danny Driess, and Marc Toussaint, (2022).
Deep visual constraints: Neural implicit models for manipulation planning from visual input.
IEEE Robotics and Automation Letters, 7(4):10857–10864.

[6] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake, (2022).
KPAM: KeyPoint Affordances for Category-Level Robotic Manipulation.
In Tamim Asfour, Eiichi Yoshida, Jaeheung Park, Henrik Christensen, and Oussama Khatib, editors, *Robotics Research*, volume 20, pages 132–157.

[7] Alexandros Paraschos, Christian Daniel, Jan R. Peters, and Gerhard Neumann, (2013).
Probabilistic movement primitives.
Advances in neural information processing systems, 26.

[8] Dean A. Pomerleau, (1988).
Alvinn: An autonomous land vehicle in a neural network.

Advances in neural information processing systems, 1.

[9] Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell, (2011).
A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning.

[10] Stefan Schaal, Auke Ijspeert, and Aude Billard, (2003).
Computational approaches to motor learning by imitation.
Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences, 358(1431):537–547.

[11] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B. Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann, (2022).
Neural descriptor fields: Se (3)-equivariant object representations for manipulation.
In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400.

[12] Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine, (2020).
AVID: Learning Multi-Stage Tasks via Pixel-Level Translation of Human Videos.

[13] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn, (2023).
Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware.