VideoDex: Learning Dexterity from Internet Videos

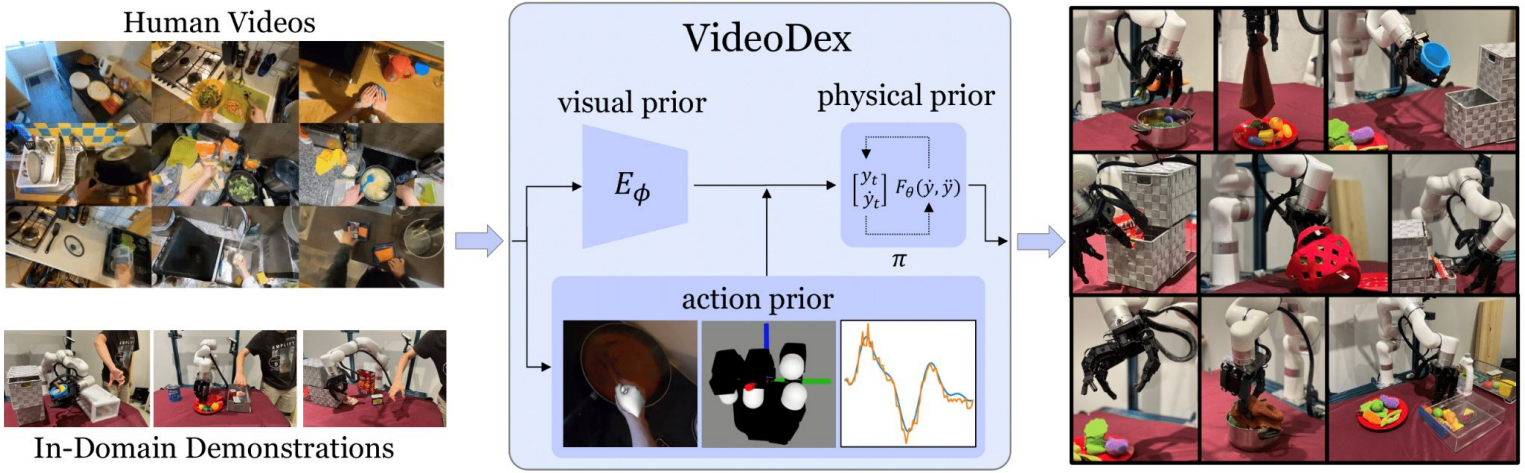
Kenneth Shaw, Shikhar Bahl, and Deepak Pathak

Presentation by Katsumi Ibaraki

Introduction

- Want autonomously performing robots
  - Require successful robotic interaction data
- Use the next best thing: real-world human interaction videos
  - Both visual priors and action priors
  - Combine with physical prior

Introduction



Background

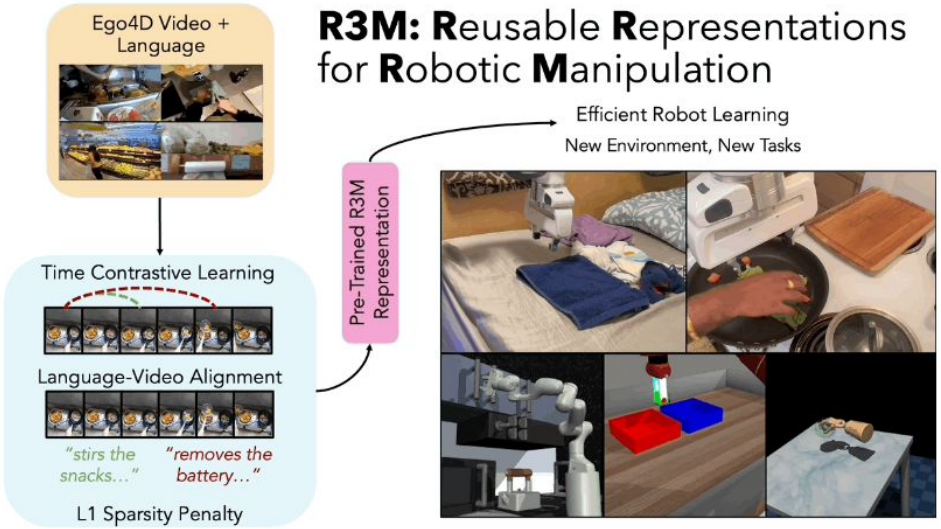
- Neural Dynamic Policies
  - Produce smooth and safe open-loop trajectories
  - $\ddot{y} = \alpha(\beta(g - y) - \dot{y}) + f_w(x, g)$
- Learning from Watching Humans
  - Borrowed from Robotic Telekinesis
  - $E_\pi(\ (\beta_h, \theta_h),\ q\ ) = \sum_{i=1}^{10} ||v_i^h - (c_i \cdot v_i^r)||_2^2$

Learning Dexterity

- Use human hand action data as prior robot experience
  - Visual + motion, intent, and interaction (action)
  - Re-target human video data

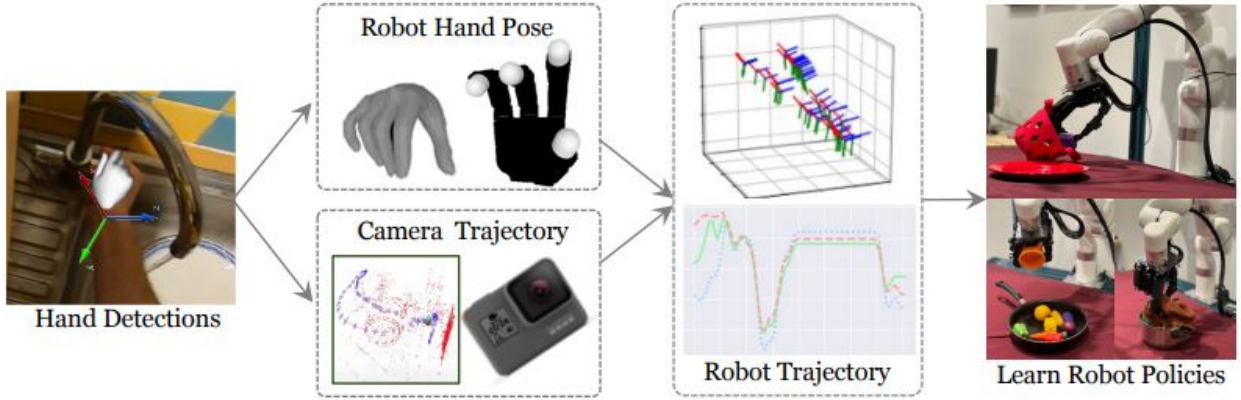
Visual Priors

- R3M (Nair et al.)
  - Visual-language alignmer
  - Encode semantic visual priors
  - Human video frames → visual representations



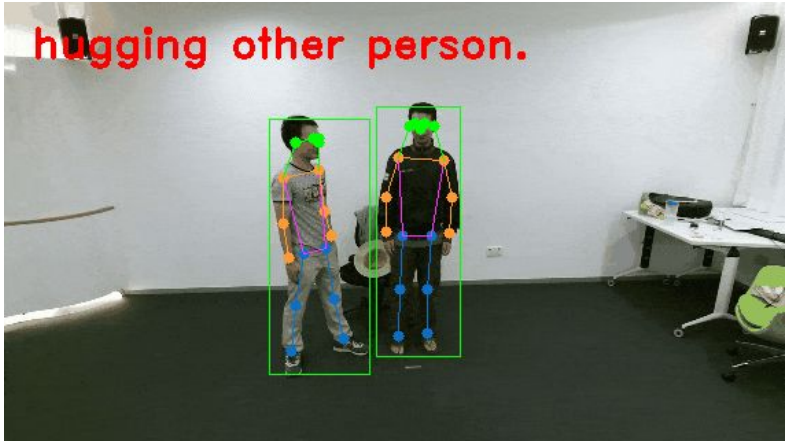
Action Priors

- Re-target human motion
  - Action and hand detection
  - Re-targeting wrist pose
  - Re-targeting hand pose



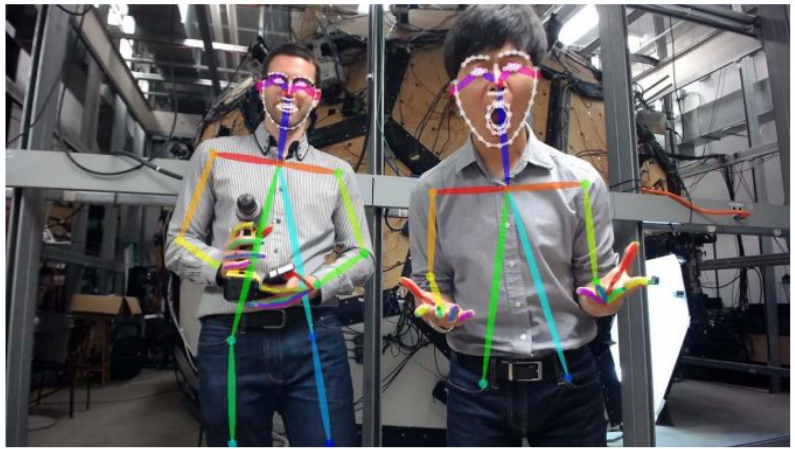
Action Priors: Detection

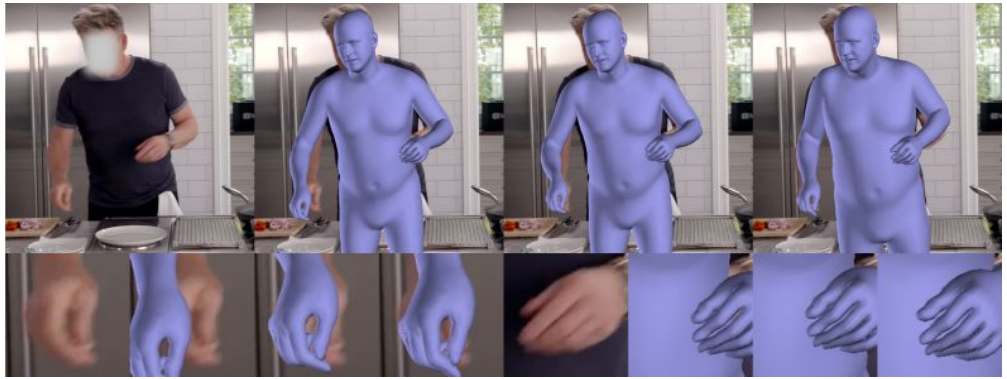
- Detect the right actions
  - EpicKitchens dataset
  - **Action detection network (Openmmlab)**



Action Priors: Detection

- Detect the hand
  - Compute hand with OpenPose
  - FrankMocap to obtain hand shape and pose parameters



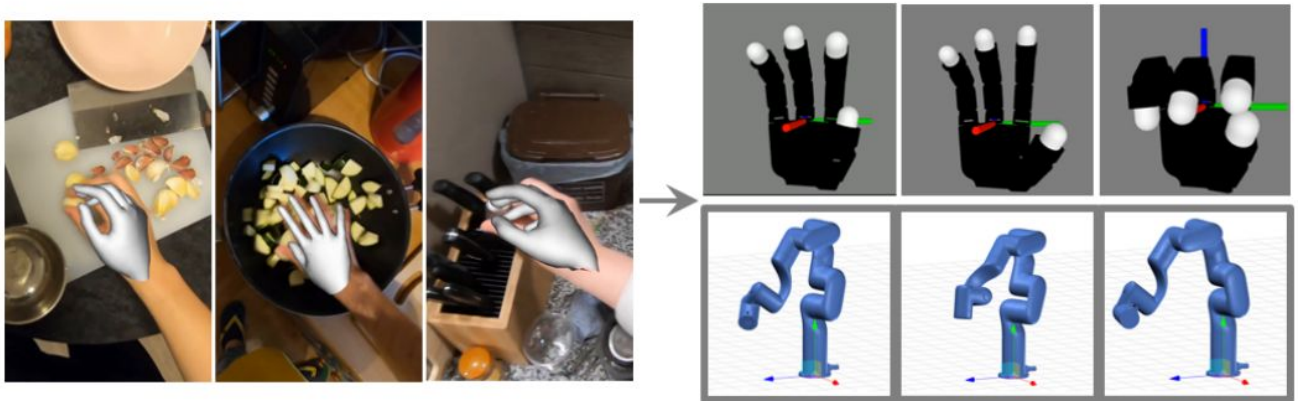


Action Priors: Wrist Pose

- Perspective-n-point algorithm
  - Compute transformation of wrist pose
- Compute transformation between first frame and other frames
- Find vector parallel to gravity in frame
- Rotate human trajectory to fit workspace limits

Action Priors: Hand Pose

- Use 16 DOF LEAP hand
- Given human pose, match to robot’s embodiment



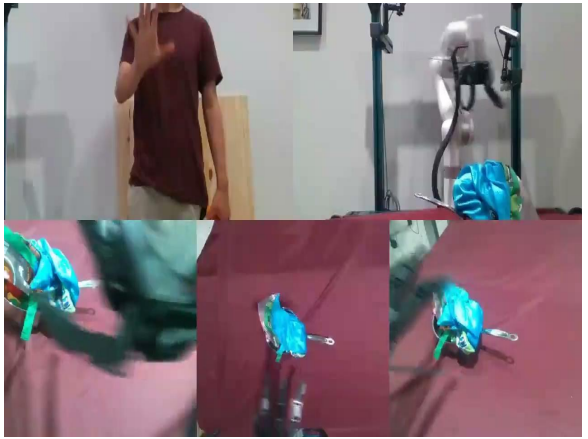
Learning

- Design an open-loop policy
  - Learn from re-targeted human trajectories (video)
  - Learn from real robot trajectories (teleoperation)
- ResNet-based encoder + physical prior

$$\mathcal{L} = \sum_{k} \text{Loss}_{L1}(\tau_R - [f_{\text{hand}}(E_\phi(I_k)), f_{\text{wrist}}(E_\phi(I_k))])$$

Training

- 500-3000 video clips
  - Retargeted to robot domain
- Final policy trained on teleoperated demonstrations
- 3-layer MLP

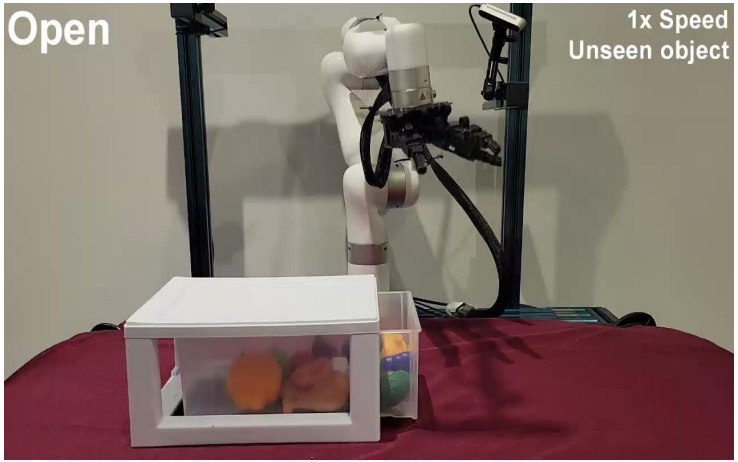


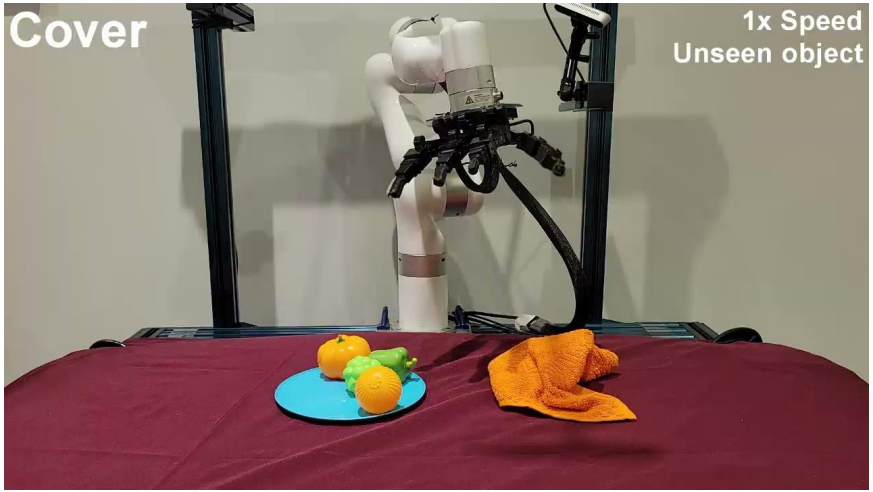
Setup

- Pretrain the action priors on seven tasks
- 120-175 demonstrations per task

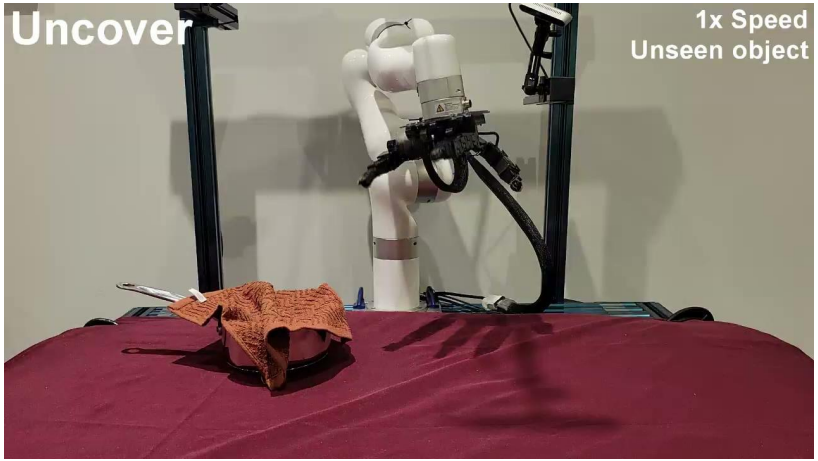
Results

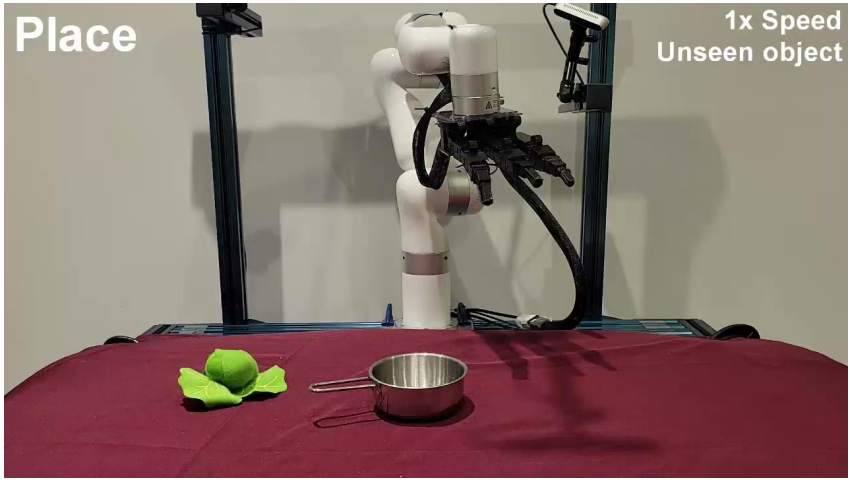
Results



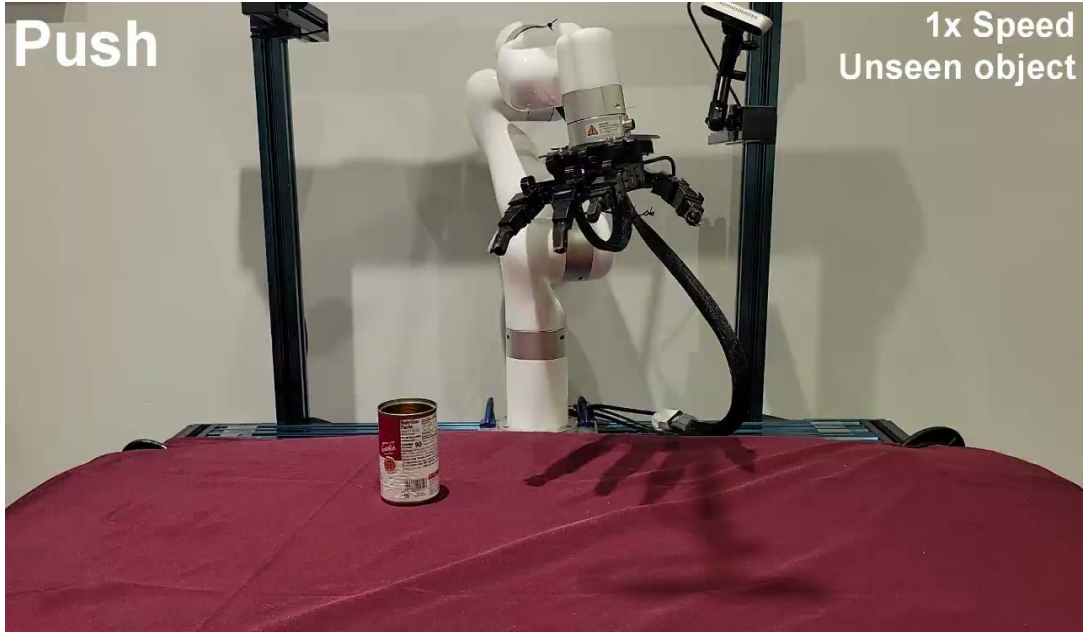


Results


Uncover
1x Speed
Unseen object


Place
1x Speed
Unseen object

Results



Results

	Pick		Rotate		Open		Cover		Uncover		Place		Push	
	train	test	train	test	train	test	train	test	train	test	train	test	train	test
BC-NDP [14]	0.64	0.38	**0.94**	0.56	**0.90**	0.60	**0.78**	0.58	0.88	0.82	0.70	0.35	1.00	0.71
BC-Open[51]	0.50	0.44	0.72	0.38	0.80	0.40	0.44	0.58	**1.00**	**0.91**	0.40	0.25	1.00	0.93
BC-RNN [51]	0.56	0.31	0.78	0.50	**0.90**	0.50	0.56	0.42	0.88	0.75	0.70	0.50	1.00	**1.00**
VideoDex	**0.83**	**0.77**	0.85	**0.71**	**0.80**	**0.80**	**0.75**	**0.63**	**0.96**	0.92	**0.89**	**0.80**	1.00	**1.00**

Table 1: We present the results of train objects and test objects for Videodex and baselines.

Results

- Action priors
  - Outperform or similar performance
- Hand vs 2-finger
  - Still improves
- Initial pose

	Place	Open	Pick
1-DOF BC-Open[51]	0.62	0.69	0.71
1-DOF VideoDex	**0.69**	**0.82**	**0.77**

	Place	Cover	Uncover
VideoDex-Fixed	0.55	0.50	0.77
VideoDex-Random	0.45	0.63	0.85
VideoDex-IMU	0.70	**0.67**	0.90
VideoDex	**0.80**	0.63	**0.92**

Results

- Generalization
  - Less teleoperated demonstrations
  - Still a 30% success rate for unseen objects
- Visual Priors
  - Visual priors help, but action priors more impactful

Visual Prior Ablation:		
VideoDex-VGG	0.20	0.20
VideoDex-MVP	0.40	0.20
Constrained Data:		
VideoDex-Const-5	0.80	0.60
VideoDex-Const-10	0.50	0.30
VideoDex **(ours)**	**0.90**	**0.70**

Future Work

- Use internet videos
- Better human hand detection modules
- React to changes in environment

Discussion

@96_f3

This type of work may in some ways be more immediately useful than the "match human infant performance by doing what babies do"-type work.

One of the advantages of LLMs (e.g. ChatGPT) is while they don't have a great understanding of the physical world, the amount of training data available on the internet is so incredibly large it is possible to learn useful structures from this data alone.

In the same vein: it may be less efficient (compute-wise) to learn how to perform motor tasks mostly from watching billions of hours of video (compared to spending years doing real physical exploration), but we may see much faster results at a "useful" level because this setup is suitable for gigantic training clusters, no hardware needed.

Discussion

@96_f3

In what domains might this paper’s approach be better than trying to emulate an infant’s development?

If training the robot directly in the real-world was feasible, would there still be benefits to this approach?

Discussion

@96_f2

The use of internet video data for robot learning is indeed well-motivated, as it provides access to a large, diverse dataset of human demonstrations that can be used to train robots to perform dexterous manipulation tasks. However there may be limitations to the extent to which we can explore and learn from more sophisticated problems. While the use of internet video data is a promising approach, there may be challenges in effectively pre-processing and extracting meaningful information from large and diverse datasets.

Discussion

@96_f2

What are possible limitations/challenges to this approach?