Robotic Visuomotor Control with Unsupervised Forward Model Learned from Videos

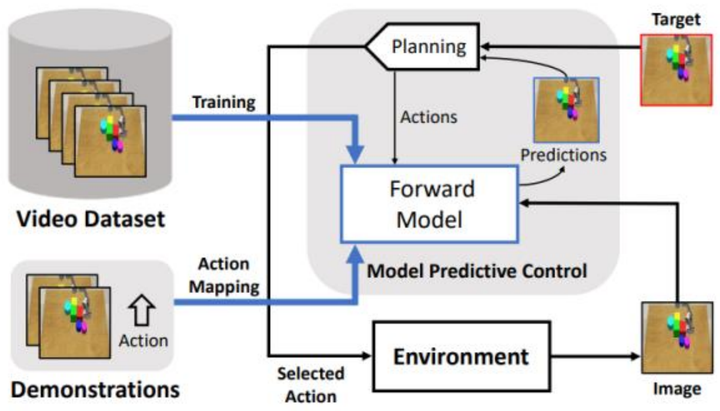
Student: Haoqi Yuan, Ruihai Wu, Andrew Zhao, Haipeng Zhang, Zihan Ding Advisor: Hao Dong

Turing Class (Class of 2017), School of EECS, Peking University

{yhq, wuruihai, 1800012831}@pku.edu.cn, {andrewzhao112, zhding96}@gmail.com & hao.dong@pku.edu.cn

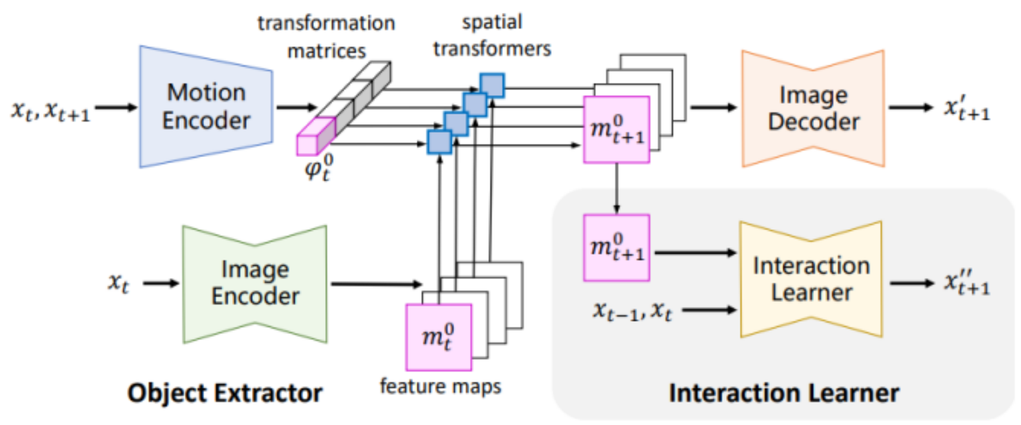
Introduction

Learning an accurate model of the environment is essential for model-based control tasks. Existing methods in robotic visuomotor control usually learn from data with heavily labelled actions, object entities or locations, which can be demanding in many cases. To cope with this limitation, we propose a method that trains a forward model from video data only, via disentangling the motion of controllable agent to model the transition dynamics. An object extractor and an interaction learner are trained in an end-to-end manner without supervision. The agent's motions are explicitly represented using spatial transformation matrices containing physical meanings. Our method achieves superior performance on learning an accurate forward model in a Grid World environment, as well as a more realistic robot control environment in simulation. With the accurate learned forward models, we further demonstrate their usage in model predictive control as an effective approach for robotic manipulations.



Methods

Problem setting: the environment for robotic/object manipulation is a Markov decision process, which can be represented as (S, A, P). State $s \in S$ is consecutive frames reflecting the position and velocity information of objects within the scene. The agent takes an action $a \in A$ based on current state s and change it to the next state s' according to the environment model P. Given a video dataset $\{s\}$ and a few transitions labelled with actions $\{(s, a, s')\}$, our target is to learn the environment model.



Method overview: In the object extractor, the observation x_t is disentangled into a set of object feature maps $\{m_t^i\}$. $\{\varphi_t^i\}$ are transformation matrices extracted from consecutive frames $\{x_t, x_{t+1}\}$, representing objects' motion. The spatial transformer transforms all the feature maps to reconstruct the next frame through the decoder. The interaction learner extracts action information through the transformed map m_{t+1}^0, to predict the environment transition. The training loss is:

$$\mathcal{L} = \|x'_{t+1} - x_{t+1}\|_2^2 + \|x''_{t+1} - x_{t+1}\|_2^2$$

Results

We evaluate our method in Grid World and a robotic simulation environment, comparing with supervised (World Models AE/VAE, E-D CNN, C-SWM) and unsupervised (CLASP) baseline methods and an ablation method.

Video prediction:
In Fig 6, qualitatively, our proposed method based on STN achieves moderate prediction errors in recursive forecasts without losing the track of objects and the agent, while other methods either have more and more obscure image prediction results, like WM AE, or have accurate object location prediction but obscure agent prediction, like in CLASP and EDCNN.
Table I shows the quantitative evaluation results, where MSE means the mean squared error in image space, $Pos\ err$ means the estimated objects position error. Compared to ours-No STN, our method with spatial transformers shows significant advantages in both environments. Even though our method cannot outperform all the baselines when models are trained on 100% action supervision, it can still have better performance than all baselines when they are trained on the reduced training set. Note that, even with 10% of the action labels, the baselines still use more supervision than ours.

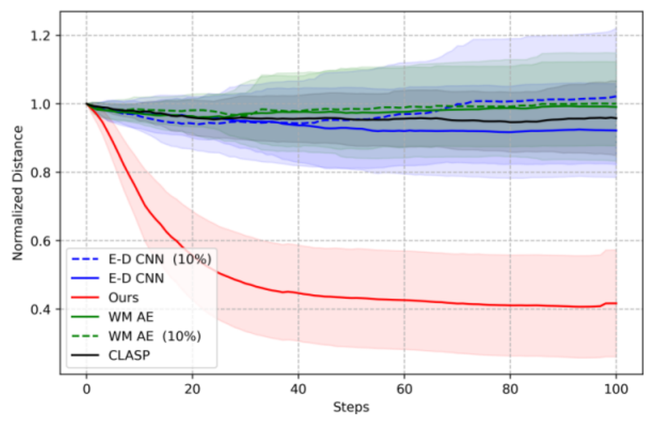


Fig. 5. Results of visuomotor control for Robot Pushing object manipulation task. The horizontal axis is the time step. The vertical axis is the average normalised distance between current and desired object locations, with the shaded regions indicating the standard deviations. Dotted lines show results from baselines trained with 10% of labelled data.

Model predictive control:
Based on the forward model with motion disentangling, we are able to solve planning and control tasks using model predictive control methods. We combine our model with cross entropy method (CEM) for optimal control.
In Fig 5, the forward model learned in our method with CEM is capable of achieving a much smaller average distance value (0.417) to the target positions, compared against all other methods (best 0.922).

Model	Grid World		Robot Pushing	
	MSE	*Pos err.*	*MSE*	*Pos err.*
WM AE	151±473	3.19±17.8	202±88.81	1.52±2.94
WM VAE	263±642	6.26±24.3	170.76±98.52	1.29±2.7
E-D CNN	27.4±139	**0.278**±4.85	87.37±98.63	0.51±0.81
C-SWM	18.2±94.9	**0.251**±3.56	552.18±99.32	4.46±6.65
WM AE(10%)	1169±775	19.9±38.4	276.82±118.59	2.11±3.88
WM VAE(10%)	960±724	20.2±40.1	256.03±126.79	1.94±3.82
E-D CNN(10%)	60.7±230	1.76±13.8	96±99.73	0.55±0.92
C-SWM(10%)	122±338	2.13±13.1	593.03±152.99	6.09±6.65
CLASP	714.7±355.5	4.8±1.7	**81.77**±93.94	0.45±0.78
Ours(No STN)	371±537	3.52±15.9	164.69±136.94	0.75±1.86
Ours	**14.3**±99.8	**0.480**±7.62	**86.78**±132.47	**0.38**±0.69

TABLE I
QUANTITATIVE EVALUATION RESULTS. A LOWER SCORE MEANS BETTER PERFORMANCE. BASELINE MODELS MASKED BY 10% USE 10% OF ACTION LABELS IN THE TRAINING SET.

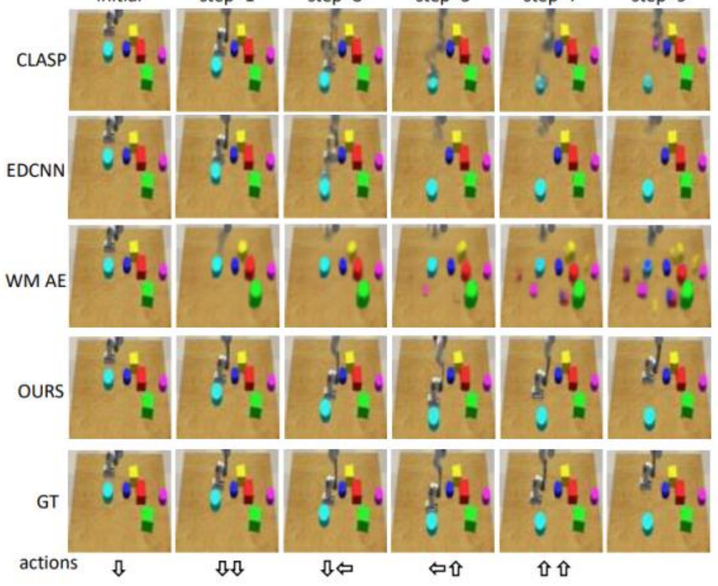


Fig. 6. Visualisation of longterm forecasting in the Robot Pushing environment. Action labels are provided at each timestep and we recursively generate the next timestep. Notice how the baselines fail to pinpoint and disentangle the exact location of the agent after a few timesteps.