# Zero-Shot Visual Imitation

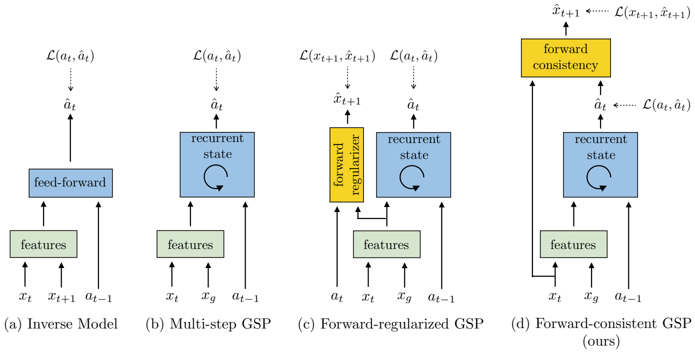
Deepak Pathak*, Parsa Mahmoudieh*, Guanghao Luo*, Pulkit Agrawal*, Dian Chen,
Fred Shentu, Evan Shelhamer, Jitendra Malik, Alexei A. Efros, Trevor Darrell
UC Berkeley
{pathak,parsa.m,michaelluo,pulkitag,dianchen,
yideshentu,shelhamer,malik,efros,trevor}@cs.berkeley.edu

## 1. Introduction

Imitating expert demonstration is a powerful mechanism for learning to perform tasks from raw sensory observations. The current dominant paradigm in learning from demonstration (LfD) [3,16,19,20] requires the expert to either manually move the robot joints (i.e., kinesthetic teaching) or teleoperate the robot to execute the desired task. The expert typically provides multiple demonstrations of a task at training time, and this generates data in the form of observation-action pairs from the agent's point of view. The agent then distills this data into a policy for performing the task of interest. Such a heavily supervised approach, where it is necessary to provide demonstrations by controlling the robot, is incredibly tedious for the human expert. Moreover, for every new task that the robot needs to execute, the expert is required to provide a new set of demonstrations.

Instead of communicating *how* to perform a task via observation-action pairs, a more general formulation allows the expert to communicate only *what* needs to be done by providing the observations of the desired world states via a video or a sparse sequence of images. This way, the agent is required to infer how to perform the task (i.e., actions) by itself. In psychology, this is known as *observational learning* [4]. While this is a harder learning problem, it is a more interesting setting, because the expert can demonstrate multiple tasks quickly and easily.

In this paper, we follow [1, 13, 18] in pursuing an alternative paradigm, where an agent explores the environment without any expert supervision and distills this exploration data into goal-directed skills. These skills can then be used to imitate the visual demonstration provided by the expert [15]. Here, by skill we mean a function that predicts the sequence of actions to take the agent from the current observation to the goal. We call this function a *goal-conditioned skill policy (GSP)*. The GSP is learned in a self-supervised way by re-labeling the states visited during the agent's exploration of the environment as goals and the actions executed by the agent as the prediction targets, similar to [1,2].



(a) Inverse Model (b) Multi-step GSP (c) Forward-regularized GSP (d) Forward-consistent GSP (ours)

Figure 1: The goal-conditioned skill policy (GSP) takes as input the current and goal observations and outputs an action sequence that would lead to that goal. We compare the performance of the following GSP models: (a) Simple inverse model; (b) Mutli-step GSP with previous action history; (c) Mutli-step GSP with previous action history and a forward model as regularizer, but no forward consistency; (d) Mutli-step GSP with forward consistency loss.

One critical challenge in learning the GSP is that, in general, there are multiple possible ways of going from one state to another: that is, the distribution of trajectories between states is multi-modal. We address this issue with our novel *forward consistency loss* based on the intuition that, for most tasks, reaching the goal is more important than how it is reached; details follow in method section. To account for varying number of steps required to reach different goals, we propose to jointly optimize the GSP with a goal recognizer that determines if the current goal has been satisfied. See Figure 1 for a schematic illustration.

We call our method *zero-shot* because the agent never has access to expert actions, neither during training of the GSP nor for task demonstration at inference. In contrast, most recent work on one-shot imitation learning requires full knowledge of actions and a wealth of expert demonstrations during training [6, 7]. In summary, we propose

*Denotes equal contribution. This is a workshop version of the ICLR 2018 paper, available here: https://pathak22.github.io/zeroshot-imitation/

a method that (1) does not require any extrinsic reward or expert supervision during learning, (2) only needs demonstrations during inference, and (3) restricts demonstrations to visual observations alone rather than full state-actions. Instead of learning by imitation, our agent learns to imitate.

## 2. Imitation without Expert Supervision

Let $\mathcal{S} : \{x_1, a_1, x_2, a_2, ..., x_T\}$ be the sequence of observations and actions generated by the agent as it explores its environment using the policy $a = \pi_E(s)$. This exploration data is used to learn the goal-conditioned skill policy (GSP). $\pi$ takes as input a pair of observations $(x_i, x_g)$ and outputs sequence of actions $(\vec{a}_\tau : a_1, a_2...a_K)$ required to reach the goal observation $(x_g)$ from the current observation $(x_i)$.

$$\vec{a}_\tau = \pi(x_i, x_g; \theta_\pi) \tag{1}$$

where states $x_i, x_g$ are sampled from the $\mathcal{S}$. The number of actions, $K$, is also inferred by the model. We represent $\pi$ by a deep network with parameters $\theta_\pi$ in order to capture complex mappings from visual observations $(x)$ to actions. $\pi$ can be thought of as a variable-step generalization of the inverse dynamics model [9], or as the policy corresponding to a universal value function [8, 21], with the difference that $x_g$ need not be the end goal of a task but can also be an intermediate sub-goal.

Let the task to be imitated be provided as a sequence of images $\mathcal{D} : \{x^d_1, x^d_2, ..., x^d_N\}$ captured when the expert demonstrates the task. This sequence of images $\mathcal{D}$ could either be temporally dense or sparse. Our agent uses the learned GSP $\pi$ to imitate the sequence of visual observations $\mathcal{D}$ starting from its initial state $x_0$ by following actions predicted by $\pi(x_0, x^d_1; \theta_\pi)$. Let the observation after executing the predicted action be $x'_0$. Since multiple actions might be required to reach close to $x^d_1$, the agent queries a separate *goal recognizer* network to ascertain if the current observation is close to the goal or not. If the answer is negative, the agent executes the action $a = \pi(x'_0, x^d_1; \theta_\pi)$. This process is repeated iteratively until the *goal recognizer* outputs that agent is near the goal, or a maximum number of steps are reached. Let the observation of the agent at this point be $\hat{x}_1$. After reaching close to the first observation $(x^d_1)$ in the demonstration, the agent sets its goal as $(x^d_2)$ and repeats the process. The agent stops when all observations in the demonstrations are processed.

### 2.1. Goal-conditioned Skill Policy (GSP)

We first describe the one-step version of GSP, and describe the multi-step extension and feature space generalization in the main paper [1]. One-step trajectories take the form of $(x_t, a_t, x_{t+1})$.

**Forward Consistency Loss** Instead of penalizing the actions predicted by the GSP to match the ground truth, we propose to learn the parameters of GSP by minimizing the distance between observation $\hat{x}_{t+1}$ resulting by executing the predicted action $\hat{a}_t = \pi(x_t, x_{t+1}; \theta_\pi)$ and the observation $x_{t+1}$, which is the result of executing the ground truth action $a_t$ being used to train the GSP. In this formulation, even if the predicted and ground-truth action are different, the predicted action will not be penalized if it leads to the same next state as the ground-truth action. We call this penalty the *forward consistency loss*.

In this work, we learn the forward dynamics $f$ model from the data, and is defined as $\tilde{x}_{t+1} = f(x_t, a_t; \theta_f)$. Let $\hat{x}_{t+1} = f(x_t, \hat{a}_t; \theta_f)$ be the state prediction for the action predicted by $\pi$. In order to make the outcome of action predicted by the GSP and the ground-truth action to be *consistent* with each other, we include an additional term, $\|x_{t+1} - \hat{x}_{t+1}\|^2_2$ in our loss function and infer the parameters $\theta_f$ by minimizing $\|x_{t+1} - \tilde{x}_{t+1}\|^2_2 + \lambda\|x_{t+1} - \hat{x}_{t+1}\|^2_2$, where $\lambda$ is a scalar hyper-parameter. The first term ensures that the learned forward model explains ground truth transitions $(x_t, a_t, x_{t+1})$ collected by the agent and the second term ensures consistency. The joint objective for training GSP with forward model consistency is:

$$\begin{aligned} \min_{\theta_\pi, \theta_f} \; \|x_{t+1} - \tilde{x}_{t+1}\|^2_2 + \lambda\|x_{t+1} - \hat{x}_{t+1}\|^2_2 \; + \; \mathcal{L}(a_t, \hat{a}_t) \\ \text{s.t.} \quad \tilde{x}_{t+1} = f(x_t, a_t; \theta_f) \\ \hat{x}_{t+1} = f(x_t, \hat{a}_t; \theta_f) \\ \hat{a}_t = \pi(x_t, x_{t+1}; \theta_\pi) \end{aligned}$$

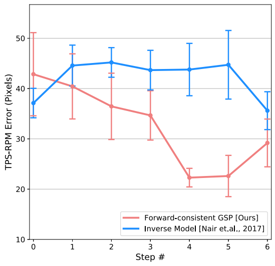
## 3. Experiments

Following methods will be evaluated: (1) Inverse Model: Nair *et al.* [15] leverage vanilla inverse dynamics to follow demonstration in rope manipulation setup. We compare to their method in both visual navigation and manipulation. (2) GSP-NoPrevAction-NoFwdConst is the ablation of our recurrent GSP without previous action history and without forward consistency loss. (3) GSP-NoFwdConst refers to our recurrent GSP with previous action history, but without forward consistency objective. (4) GSP-FwdRegularizer refers to the model where forward prediction is only used to regularize the features of GSP but has no role to play in the loss function of predicted actions. The purpose of this variant is to particularly ablate the benefit of consistency loss function with respect to just having forward model as feature regularizer. (5) GSP refers to our complete method with all the components.

### 3.1. Rope Manipulation

Manipulation of non-rigid and deformable objects, e.g., rope, is a challenging problem in robotics. To test whether

[1] Full ICLR 2018 paper at https://pathak22.github.io/zeroshot-imitation/



(a) TPS-RPM error for ‘S’ shape manipulation

| Method | Success % |
|---|---|
| Inverse Model [Nair et.al. 2017] | 36% ± 9.6% |
| Forward-regularized GSP | 44% ± 9.9% |
| Forward-consistent GSP [Ours] | **60**% ± **9.8**% |

(b) Success rate for Knot-tying

Figure 2: GSP trained using forward consistency loss significantly outperforms the baselines at the task of (a) manipulating rope into ‘S’ shape as measured by TPS-RPM error and (b) knot-tying where we report success rate with bootstrap standard deviation.

our agent could manipulate ropes by simply observing a human, we use the data collected by Nair *et al.* [15], where a Baxter robot manipulated a rope kept on the table in front of it. During exploration, the robot interacts with the rope by using a pick and place primitive that chooses a random point on the rope, and displaces it by a randomly chosen length and direction. This process is repeated number of times to collect about 60K interaction pairs of the form $(x_t, a_t, x_{t+1})$ that are used to train the GSP. During inference, our proposed approach is tasked to follow a visual demonstration provided by a human expert for manipulating the rope into a complex ‘S’ shape and tying a knot.

We compare our approach to the baseline that deploys an inverse model which takes as input a pair of current and goal images to output the desired action to reach goal [15]. We re-implement the baseline and train in our setup for a fair comparison. To further ablate the importance of consistency loss, we compare to a baseline that just uses forward model as a regularizer of features. The results in Figure 2 show that our method significantly outperforms the baseline at task of manipulating the rope in the ‘S’ shape and achieves a success rate of 60% in comparison to 36% for knot tying.

### 3.2. Navigation in Indoor Office Environments

For navigation, both real-world and simulation, we check generalization by testing on a novel building/floor. We used TurtleBot2 which has an onboard RGB camera for indoor office navigation. For learning the GSP, an automated self-supervised scheme for data collection was devised that required no human supervision. The robot collected number of trajectories that contain 230K interactions data, i.e. $(x_t, a_t, x_{t+1})$, from two floors of a academic building. We then deployed the learned model on a separate floor of a building with substantially different textures and layout for visual imitation at test time.

We tested if the GSP learned by the TurtleBot can enable it to find its way to a goal that is within the same room from just a *single image* of the goal. To test an extrapolative generalization, we keep the Turtlebot approximately 20-30 steps away from the target location in a way that current and goal observation has no overlap. We judge the robot to be successful if it stops close to the goal and failure if it crashed into furniture or does not reach the goal within 200 steps. Since the initial and goal images have no overlap, classical techniques such as structure from motion that rely on feature matching cannot be used to infer the robot’s action. In order to reach the goal, the robot must explore its surroundings. We find that our GSP model outperforms the baseline models in reaching the target location. Our model learns the exploratory behavior of rotating at its location until it encounters overlap between its current image and goal image. Results are shown in Table 1 and videos are available at the website [2].

| Model Name | Success Rate |
|---|---|
| Random Search | 0/8 |
| Inverse Model [Nair et. al. 2017] | 0/8 |
| GSP-NoPrevAction-NoFwdConst | 2/8 |
| GSP-NoFwdConst | 4/8 |
| GSP (Ours) | 6/8 |

Table 1: Quantitative evaluation of various methods on the task of navigating using a *single image* of goal in an unseen environment. Our full GSP model outperforms the baselines significantly.

## 4. Related Work

Nair *et al.* [15] observe a sequence of images from the expert demonstration for performing rope manipulations. Sermanet *et al.* [22] imitate humans with robots by self-supervised learning but require expert supervision at training time. Third person imitation learning [23] and the concurrent work of imitation-from-observation [14] learn to translate expert observations into agent observations such that they can do policy optimization to minimize the distance between the agent trajectory and the translated demonstration, but they require demonstrations for learning.Visual servoing is a standard problem in robotics [5, 10–12, 24, 26] that seeks to take actions that align the agent’s observation with carefully-designed visual features or raw pixel intensities. The works of Jordan *et al.* [9]; Wolpert *et al.* [25]; Agrawal *et al.* [1]; Pathak *et al.* [17] jointly learn forward and inverse dynamics model but do not optimize for consistency between the forward and inverse dynamics. We empirically show that learning models by our forward consistency loss significantly improves task performance.

[2] https://pathak22.github.io/zeroshot-imitation/

## References

[1] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine. Learning to poke by poking: Experiential learning of intuitive physics. *NIPS*, 2016. 1, 3

[2] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. In *NIPS*, 2017. 1

[3] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 2009. 1

[4] A. Bandura and R. H. Walters. *Social learning theory*, volume 1. Prentice-hall Englewood Cliffs, NJ, 1977. 1

[5] G. Caron, E. Marchand, and E. M. Mouaddib. Photometric visual servoing for omnidirectional cameras. *Autonomous Robots*, 35(2-3):177–193, 2013. 3

[6] Y. Duan, M. Andrychowicz, B. Stadie, O. J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba. One-shot imitation learning. In *NIPS*, 2017. 1

[7] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine. One-shot visual imitation learning via meta-learning. *CoRL*, 2017. 1

[8] D. Foster and P. Dayan. Structure in the space of value functions. *Machine Learning*, 2002. 2

[9] M. I. Jordan and D. E. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive science*, 1992. 2, 3

[10] H. Koichi and H. Tom. *Visual servoing: real-time control of robot manipulators based on visual sensory feedback*, volume 7. World scientific, 1993. 3

[11] T. Lampe and M. Riedmiller. Acquiring visual servoing reaching and grasping skills using neural reinforcement learning. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8. IEEE, 2013. 3

[12] A. X. Lee, S. Levine, and P. Abbeel. Learning visual servoing with deep features and fitted q-iteration. *arXiv preprint arXiv:1703.11000*, 2017. 3

[13] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen. Learning hand-eye coordination for robotic grasping with large-scale data collection. In *ISER*, 2016. 1

[14] Y. Liu, A. Gupta, P. Abbeel, and S. Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. *ICRA*, 2018. 3

[15] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. *ICRA*, 2017. 1, 2, 3

[16] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *ICML*, pages 663–670, 2000. 1

[17] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017. 3

[18] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. *ICRA*, 2016. 1

[19] D. A. Pomerleau. ALVINN: An autonomous land vehicle in a neural network. In *NIPS*, 1989. 1

[20] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 1999. 1

[21] T. Schaul, D. Horgan, K. Gregor, and D. Silver. Universal value function approximators. In *ICML*, 2015. 2

[22] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, 2018. 3

[23] B. C. Stadie, P. Abbeel, and I. Sutskever. Third-person imitation learning. In *ICLR*, 2017. 3

[24] W. J. Wilson, C. W. Hulls, and G. S. Bell. Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5):684–696, 1996. 3

[25] D. M. Wolpert, Z. Ghahramani, and M. I. Jordan. An internal model for sensorimotor integration. *Science-AAAS-Weekly Paper Edition*, 1995. 3

[26] B. H. Yoshimi and P. K. Allen. Active, uncalibrated visual servoing. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 156–161. IEEE, 1994. 3