Deep Reinforcement Learning for Industrial Insertion Tasks with Visual Inputs and Natural Rewards

Gerrit Schoettler [*1] **Ashvin Nair** [*2] **Jianlan Luo** [2] **Shikhar Bahl** [2]
Juan Aparicio Ojea [1] **Eugen Solowjow** [1] **Sergey Levine** [2]

Abstract

Connector insertion and many other tasks commonly found in modern manufacturing settings involve complex contact dynamics and friction. Since it is difficult to capture related physical effects with first-order modeling, traditional control methodologies often result in brittle and inaccurate controllers, which have to be manually tuned. Reinforcement learning (RL) methods have been demonstrated to be capable of learning controllers in such environments from autonomous interaction with the environment, but running RL algorithms in the real world poses sample efficiency and safety challenges. Moreover, in practical real-world settings we cannot assume access to perfect state information or dense reward signals. In this paper, we consider a variety of difficult industrial insertion tasks with visual inputs and different natural reward specifications, namely sparse rewards and goal images. We show that methods that combine RL with prior information, such as classical controllers or demonstrations , can solve these tasks directly by real-world interaction.

1. Electric Connector Plug Insertion Tasks

In this work, we empirically evaluate learning methods on a set of electric connector assembly tasks, pictured in Fig. 1. Connector plug insertions are difficult for two reasons. First, the robot must be very precise in lining up the plug with its socket. As we show in our experiments, errors as small as ± 1 mm can lead to consistent failure. Second, there is significant friction when the connector plug touches the socket, and the robot must learn to apply sufficient force in order to insert the plug. Image sequences of successful insertions are shown in Fig. 2, where it is also possible to see details of the gripper setup that we used to ensure a fully automated training process. In our experiments, we use a 7 degrees of freedom Sawyer robot with end-effector control, meaning that the action signal u_t can be interpreted as the relative end-effector movement in Cartesian coordinates.

To comprehensively evaluate connector assembly tasks, we repeat our experiments on a variety of connectors. Each connector offers a different challenge in terms of required precision and force to overcome friction. We chose to benchmark the controllers performance on the insertion of a USB connector, a U-Sub connector, and a waterproof Model-E connector manufactured by MISUMI. All the explored use cases were part of the IROS 2017 Robotic Grasping and Manipulation Competition (Falco et al., 2018), included as part of a task board developed by NIST to benchmark the performance of assembly robots.

1.1. Adapters

In the following we describe the used adapters, USB, D-Sub, and Model-E. The observed difficulty of the insertion increases in that order.

1.1.1. USB

The USB connector is a ubiquitous, widely-used connector and offers a challenging insertion task. Because the adapter becomes smoother and therefore easier to insert over time due to wear and tear, we periodically replace the adapter. Of the three tested adapters, the USB adapter is the easiest.

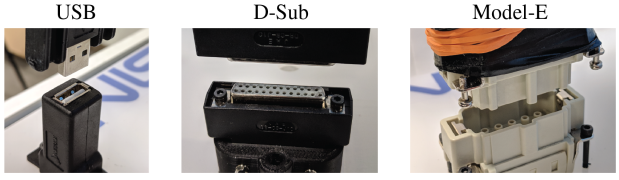

Figure 1. We train an agent directly in the real world to solve connector insertion tasks that involve contacts and tight tolerances from convenient reward signals such as pixel distance to a goal image or a sparse electrical signal.

[*]Equal contribution. [1]Siemens Corporation, Berkeley, USA [2]Berkeley AI Research, University of California, Berkeley, Computer Science. Correspondence to: Ashvin Nair <anair17@berkeley.edu>.

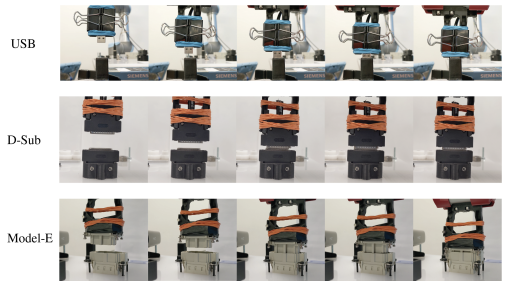

Figure 2. The three image sequences show rollouts from learned policies that successfully complete the insertion tasks.

1.1.2. D-SUB

Inserting this adapter requires aligning several pins correctly, and is therefore more sensitive than inserting the USB adapter. It also requires more downward force due to a tighter fit.

1.1.3. MODEL-E

This adapter is the most difficult of the three tested connectors as it contains several edges and grooves to align and requires significant downward force to successfully insert the part.

1.2. Experimental Settings

We consider three settings in our experiments in order to evaluate how plausible it is to solve these tasks with more convenient state representations and reward functions and to evaluate the performance of different algorithms changes as the setting is modified.

1.2.1. VISUAL

In this experiment, we evaluate whether the RL algorithms can learn to perform the connector assembly tasks from vision without having access to state information. The state provided to the learned policy is a 32×32 grayscale image, such as shown in Fig. 4. For goal specification, we use a goal image, avoiding the need for state information to compute rewards. The reward is the pixelwise L1 distance to the given goal image. Being able to learn from such a setup is compelling as it does not require any extra state estimation and many tasks can be specified easily by a goal image.

1.2.2. ELECTRICAL (SPARSE)

In this experiment, the reward is obtained by directly measuring whether the connection is alive and transmitting:

$$r = \begin{cases} 1, & \text{if insertion signal detected} \\ 0, & \text{else.} \end{cases} \quad (1)$$

This is the exact true reward for the task of connecting a cable, and can be naturally obtained in many manufacturing systems. As state, the robot is given the Cartesian coordinates of the end-effector x_t and the vertical force f_z that is acting on the end-effector. As we could only automatically detect the USB connection thus far, we only include the USB adapter for the sparse experiments.

1.2.3. DENSE

In this experiment, the robot receives a manually shaped reward based on the distance to the target. We use the reward function

$$r_t = -\alpha \cdot \|x_t - x^*\|_1 - \frac{\beta}{(\|x_t - x^*\|_2 + \varepsilon)} - \varphi \cdot f_z, \quad (2)$$

where $0 < \varepsilon \ll 1$. The hyperparameters are set to $\alpha = 100$, $\beta = 0.002$, and $\varphi = 0.1$. When an insertion is indicated through a distance measurement, the sign of the force term flips, so that $\varphi = -0.1$ when the connector is inserted. This rewards the agent for pressing down after an insertion and showed to improve the learning process.

2. Methods

To solve the connector insertion tasks, we consider and evaluate a variety of reinforcement learning algorithms.

2.1. Preliminaries

In a Markov decision process (MDP), an agent at every time step is at state $s_t \in \mathcal{S}$, takes actions $u_t \in \mathcal{U}$, receives a reward $r_t \in \mathbb{R}$, and the state evolves according to environment transition dynamics $p(s_{t+1}|s_t, u_t)$. The goal of reinforcement learning is to choose actions $u_t \sim \pi(u_t|s_t)$ to maximize the expected returns $\mathbb{E}[\sum_{t=0}^{H} \gamma^t r_t]$ where H is the horizon and γ is a discount factor. The policy $\pi(u_t|s_t)$ is often chosen to be an expressive parametric function approximator, such as a neural network, as we use in this work.

2.2. Efficient Off-Policy Reinforcement Learning

One class of RL methods additionally estimates the expected discounted return after taking action u from state s, the Q-value $Q(s, u)$. Q-values can be recursively defined with the Bellman equation:

$$Q(s_t, u_t) = \mathbb{E}_{s_{t+1}}[r_t + \gamma \max_{u_{t+1}} Q(s_{t+1}, u_{t+1})] \quad (3)$$

and learned from off-policy transitions (s_t, u_t, r_t, s_{t+1}). Because we are interested in sample-efficient real-world learning, we use such RL algorithms that can take advantage of off-policy data.

For control with continuous actions, computing the required maximum in the Bellman equation is difficult. Continuous

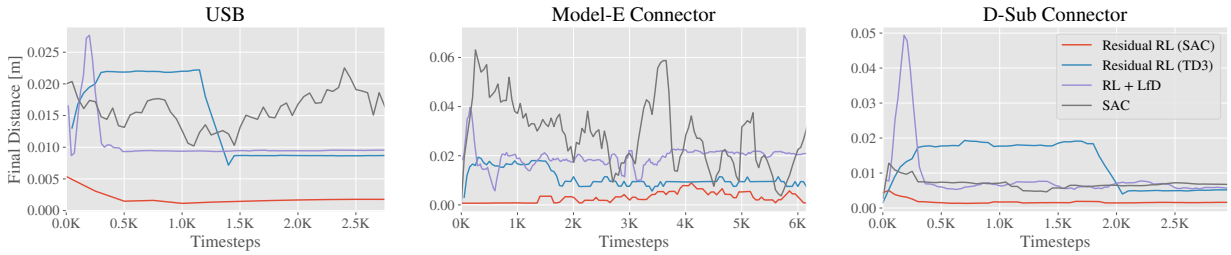

Figure 3. Resulting final mean distance during the vision-based training. The comparison includes residual RL, learning from demonstrations and pure RL, represented by SAC. Only residual RL with SAC manages to deal with the high-dimensional input and consistently solves all tasks after the given amount of training. The deterministic policies learn to move downwards, but often get stuck in the beginning of the insertion and fail to recover from unsuccessful attempts.

control algorithms such as deep deterministic policy gradients (DDPG) (Lillicrap et al., 2016) additionally learn a policy $\pi_\theta(u_t|s_t)$ to approximately choose the maximizing action. In this paper we specifically consider two related reinforcement learning algorithms that lend themselves well to real-world learning as they are sample efficient, stable, and require little hyperparameter tuning.

2.2.1. Twin Delayed Deep Deterministic Policy Gradients (TD3)

Like DDPG, TD3 optimizes a deterministic policy (Fujimoto et al., 2018) but uses two Q-function approximators to reduce value overestimation (Van Hasselt et al., 2016) and delayed policy updates to stabilize training.

2.2.2. Soft Actor Critic (SAC)

SAC is an off-policy value-based reinforcement learning method based on the maximum entropy reinforcement learning framework with a stochastic policy (Haarnoja et al., 2018).

We used the implementation of these RL algorithms publicly available at `rlkit` (Pong et al., 2018).

2.3. Residual Reinforcement Learning

Instead of randomly exploring from scratch, we can inject prior information into an RL algorithm in order to speed up the training process, as well as to minimize unsafe exploration behavior. In residual RL, actions u_t are chosen by additively combining a fixed policy $\pi_\mathrm{H}(s_t)$ with a parametric policy $\pi_\theta(u_t|s_t)$:

$$u_t = \pi_\mathrm{H}(s_t) + \pi_\theta(s_t). \tag{4}$$

The parameters θ can be learned using any RL algorithm. In this work, we evaluate both SAC and TD3, explained in the previous section.

A simple P-controller serves as the hand-designed controller π_H of our experiments. The P-controller operates in Cartesian space and calculates the current control action by

$$\pi_\mathrm{H}(s_t) = -k_p \cdot (x_t - x^*), \tag{5}$$

where x^* denotes the commanded goal location. As control gains we use $k_p = [1,\ 1,\ 0.3]$. This P-controller quickly centers the end-effector above the goal position and reaches the goal after about 10 time steps when starting from the reset positions, which is located about 5cm above the goal.

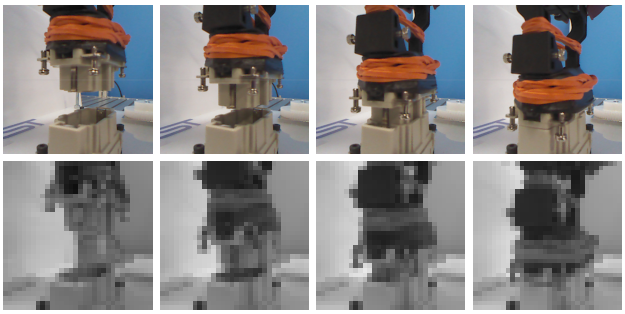

Figure 4. Successful insertion on the Model-E connector task. The 32×32 grayscale images are the only observations that the image-based reinforcement learning algorithm receives.

2.4. Learning from Demonstrations

Another method to incorporate prior information is to use demonstrations from an expert policy to guide exploration during RL. We first collected demonstrations with a keyboard controller. Then, we add a behavior cloning loss while performing RL that pushes the policy towards the demonstrator actions, as previously considered in (Nair et al., 2018). Instead of DDPG, the underlying algorithm RL algorithm used is TD3.

D-Sub connector		Perfect Goal	Noisy Goal
Human Controller		100%	44%
SAC		16%	0%
Residual RL	Dense	100%	60%
	Images	100%	64%

Model-E Connector		Perfect Goal	Noisy Goal
Human Controller		52%	24%
SAC		0%	0%
Residual RL	Dense	100%	76%
	Images	100%	76%

Figure 5. Performance evaluation on the D-Sub connector. We report the average success rate out of 25 rollouts of the trained policies.

3. Experiments

We evaluate the industrial applicability of the residual RL approach on a variety of connector insertion tasks that are performed on a real robot, using easy-to-obtain reward signals. In this section, we consider two types of natural rewards which are intuitive to humans: an image directly specifying a goal and a binary sparse reward indicating success. For both cases, we report success rates on tasks they solve. We aim to answer the following questions: (1) Can such trained policies provide comparable performance to policies that are trained with densely-shaped rewards? (2) Are these trained policies robust to light variations and noise?

3.1. Vision-based Learning

For the vision-based learning experiments, we use only raw image observations and L1 distance in image space as the goal. Sample images that the robot received are shown in Fig. 4. We evaluate this type of reward on all three tasks. In our experiments, we use grayscale images converted from RGB ones, this simplification reduces input space; but is good enough for our experiments.

3.2. Learning from Sparse Rewards

The applicability of a sparse reward function is explored on an insertion of the USB connector. The binary insertion signal is used as the metric for success. This experiment is most applicable to electronic manufacturing settings where the electrical connection between connectors can be directly measured.

3.3. Robustness

For safe and reliable future usage, it is required that the insertion controller is robust against small measurement or calibration errors that can occur when disassembling and reassembling a mechanical system. In this experiment, small goal perturbations are introduced in order to uncover the required setup precision of our algorithms.

3.4. Exploration Comparison

One advantage of using reinforcement learning is the exploratory behavior that allows the controller to adapt from new experiences, unlike a deterministic control law. The two RL algorithms we consider in this paper, SAC and TD3, explore differently. SAC maintains a stochastic policy, and the algorithm also adapts the stochasticity through training. TD3 has a deterministic policy, but uses another noise process (in our case Gaussian) to inject exploratory behavior during training time. We compare the two algorithms, as well as when they are used in conjunction with residual RL, in order to evaluate the different exploration schemes.

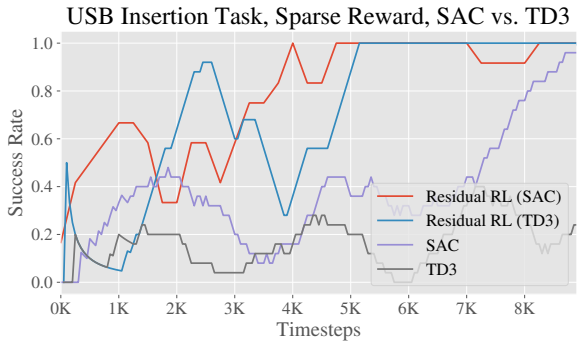

Figure 6. Comparison of two RL algorithms, SAC and TD3, on the USB insertion task with sparse rewards. Using residual RL, both algorithms can solve the task. Without residual RL, TD3 struggles to solve the task while SAC eventually does solve it to the same level of performance as the residual RL methods. We believe the difference may be due to the adaptive noise in SAC.

4. Results

In order to evaluate our experiments with dense and vision-based rewards, we analyze the achieved final distance to the goal throughout the training process. Policies trained with sparse rewards are compared based on their success rate because their training objective does not include the minimization of the distance to the goal. We report the success rate of all final policies and compare their robustness towards measurement noise in the goal location.

4.1. Vision-based Learning

The results of the vision-based experiment are shown in Fig. 3. Our experiments show that a successful and consistent vision-based insertion policy can be learned from relatively few samples using residual RL with SAC. This

result suggests that goal-specification through images is a practical way to solve these types of industrial tasks.

Interestingly, during training with pure RL, the policy would sometimes learn to "hack" the reward signal by moving down in the image in front of or behind the socket. In contrast, the stabilizing human-engineered controller in residual RL provides sufficient horizontal control to prevent this and it also transforms the 3-dimensional task into a quasi 1-dimensional problem for the reinforcement learning algorithm, which explains the very good results obtained with residual RL in conjunction with vision-based rewards.

4.2. Learning From Sparse Rewards

In this experiment, we compare several methods on the USB insertion task with sparse rewards. The results are reported in Fig. 7.

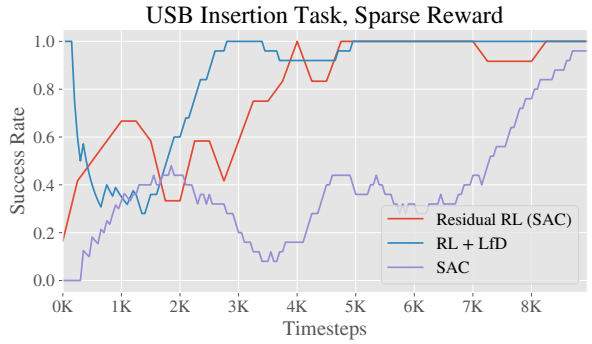

Figure 7. Learning curves for solving the USB insertion task with a sparse reward are shown. In this experiment, ground truth state is given as observations to the agent. Residual RL and RL with learning from demonstrations both solve the task relatively quickly, while RL alone (SAC) takes about twice as long to solve the task at the same level of performance.

4.3. Robustness

In previous set of experiments, the goal locations were known exactly. In this case, the hand-engineered controller performs well. However, once noise is induced to the goal location, the deterministic P-controller does not solve the task anymore. After training on perfect observations, a goal perturbation is created artificially and the controllers are tested under this condition. All results of our robustness evaluations are listed in Tab. 1, Tab. **??**, and Tab. **??**.

4.4. Exploration Comparison

A comparison of TD3 and SAC is made in Fig. 6. When combined with residual RL, they perform comparably. When considering RL alone, TD3 learns the task faster than SAC. However, TD3 is significantly less robust, as shown in Tab. 1. Furthermore, we found that the outputted action of TD3 approaches the extreme values at the edge of the allowed action space. This suggests it finds a local minimum, which performs well, but may not be robust and TD3 cannot improve beyond that policy.

Table 1. Test-time performance on the USB insertion task. Noised is added in form of $\pm 1\,\mathrm{mm}$ perturbations of the goal location. We report the average success rate out of 25 rollouts.

USB		Perfect Goal	Noisy Goal
Human Controller		100%	60%
SAC		16%	8%
RL + LfD		100%	32%
Residual RL	Dense	100%	84%
	Sparse, SAC	88%	84%
	Sparse, TD3	100%	36%
	Images	100%	80%

5. Discussion and Future Work

In this paper we studied residual RL with natural rewards and demonstrated that this approach can solve complex industrial assembly tasks with tight tolerances, e. g. connector plug insertions. We introduced vision inputs to the residual RL formulation, which increases the algorithm's usefulness for a wide range of industrial applications. Compared to previous work (Johannink et al., 2019), which uses dense reward signals, we showed that we can learn insertion policies only from sparse binary rewards or even purely from goal images. We conducted a series of experiments for various connector type assemblies and could demonstrate the feasibility of our method, even under challenging conditions such as noisy goals and complex connector geometries. Our study motivates the application of residual RL to industrial automation tasks, where reward shaping is not feasible, but sparse rewards or image goals can often be provided.

Future work will include more complex environments focusing on multi-stage assembly tasks through vision. This would pose a challenge to the goal-based policies as the background would be visually more complex. Moreover, multi-step tasks involve adapting to previous mistakes or inaccuracies, which could be difficult, however, in theory should be able to be handled by RL. Extending the presented approach to multi-stage assembly tasks will pave the road to a higher robot autonomy in flexible manufacturing.

Appendix

A. State-Based Training

A.1. Dense Reward Connector Plug Insertion

After evaluating the tasks in the above settings, we further evaluate with full state information with a dense and carefully shaped reward signal, given in Eq. 2, that incorporates distance to the goal and force information. Evaluating in this setting gives us an "oracle" that can be compared to the previous experiments in order to understand how much of a challenge sparse or image rewards pose for various algorithms.

A.2. Dense Reward Connector Plug Insertion

The results of the experiment with dense rewards are shown in Fig. 8.

Here, the same conclusions of residual RL outperforming pure RL hold. Due to the shaped reward, pure RL makes more initial progress, but cannot overcome the friction required to fully insert the plugs. It appears that the hand-designed reward function does not incentivise the full insertion enough, as we were able to obtain better results on the USB insertion with sparse rewards.

B. Related Work

Learning has been applied previously in a variety of robotics contexts. Different forms of learning have enabled autonomous driving (Pomerleau, 1989), biped locomotion (Nakanishi et al., 2004), block stacking (Deisenroth et al., 2011), grasping (Pinto & Gupta, 2016), and navigation (Giusti et al., 2015; Pathak et al., 2018). Among these methods, many involve reinforcement learning, where an agent learns to perform a task by maximizing a reward signal. Reinforcement learning algorithms have been developed and applied to teach robots to perform tasks such as balancing a robot (Deisenroth & Rasmussen, 2011), playing ping-pong (Peters et al., 2010) and baseball (Peters & Schaal, 2008). The use of large function approximators, such as neural networks, in RL has further broadened the generality of RL (Mnih et al., 2013). Such techniques, called "deep" RL, have further allowed robots to perform fine-grained manipulation tasks from vision (Levine et al., 2016), open doors (Gu et al., 2016), score a hockey puck (Chebotar et al., 2017), and grasp objects (Kalashnikov et al., 2018). In this work, we further explore solving real-world robotics tasks using RL.

Many RL algorithms introduce prior information about the specific task to be solved through various means such as reward shaping (Ng et al., 1999), incorporating a trajectory planner (Thomas et al., 2018; Eruhimov & Meeussen, 2011; Mayton et al., 2010), learning classifiers between goals and non-goals (Ho & Ermon, 2016; Pinto & Gupta, 2016; Levine et al., 2017). These methods require access to various goal states to build a robust classifier, which might be difficult to collect in assembly as there is often only one goal image possible. Reward shaping can become arbitrarily difficult as the complexity of the task increases. For complex assembly tasks, trajectory planners require a host of information about objects and geometries which can be difficult to provide.

Mainly, previous work on incorporating prior information has focused on using demonstrations either to initialize a policy (Peters & Schaal, 2008; Kober & Peter, 2008), infer reward functions using inverse reinforcement learning (Finn et al., 2016; Abbeel & Ng, 2004; Ziebart et al., 2008; Rhinehart & Kitani, 2017; Fu et al., 2018) or to improve the policy throughout the learning procedure (Hester et al., 2018; Nair et al., 2018; Rajeswaran et al., 2018; Večerík et al., 2017). These methods require multiple demonstrations, which can be difficult to collect, especially for assembly tasks. More recently, manually specifying a policy and learning the residual task has been proposed (Johannink et al., 2019; Silver et al., 2018). In this work we evaluate both residual RL and combining RL with learning from demonstrations (LfD).

Previous work has also tackled high precision assembly tasks, especially insertion-type tasks. One line of work focuses on obtaining high dimensional observations, including geometry, forces, joint positions and velocities (Li et al., 2014; Tamar et al., 2017; Inoue et al., 2017; Luo et al., 2019), but this information is not easily procured, increasing complexity of the experiments and the supervision required to collect the data. Other work relies on external trajectory planning or very high precision control (Inoue et al., 2017; Tamar et al., 2017), but this can be brittle to error in other components of the system, such as perception. We show how our method not only solves insertion tasks with much less information about the environment, it also does so under noisy conditions.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 1, 2004.

Chebotar, Y., Hausman, K., Zhang, M., Sukhatme, G., Schaal, S., and Levine, S. Combining Model-Based and

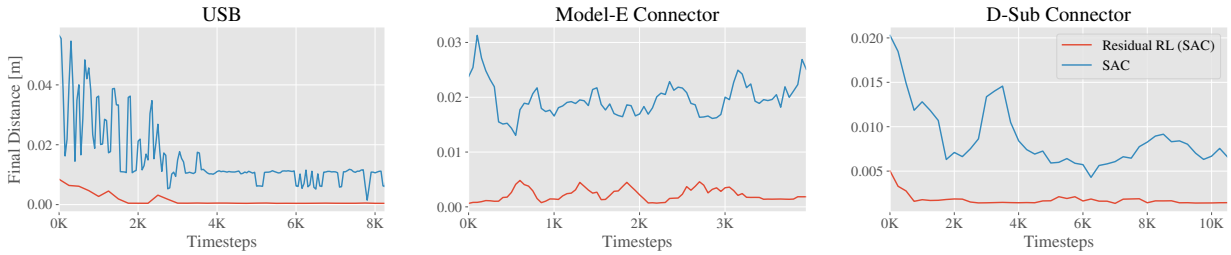

Figure 8. Plots of the final mean distance to the goal during the state-based training. Final distances greater than $0.01\,\mathrm{m}$ indicate unsuccessful insertions. Here, the residual RL approach performs noticeably better than pure RL and is often able to solve the task during the exploration in the early stages of the training.

Model-Free Updates for Trajectory-Centric Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2017.

Deisenroth, M. P. and Rasmussen, C. E. PILCO: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning (ICML)*, pp. 465–472, 2011.

Deisenroth, M. P., Rasmussen, C. E., and Fox, D. Learning to Control a Low-Cost Manipulator using Data-Efficient Reinforcement Learning. *Robotics: Science and Systems (RSS)*, VII:57–64, 2011.

Eruhimov, V. and Meeussen, W. Outlet detection and pose estimation for robot continuous operation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2941–2946, 2011.

Falco, J., Sun, Y., and Roa, M. Robotic grasping and manipulation competition: Competitor feedback and lessons learned. In Sun, Y. and Falco, J. (eds.), *Robotic Grasping and Manipulation*, pp. 180–189, Cham, 2018. Springer International Publishing. ISBN 978-3-319-94568-2.

Finn, C., Levine, S., and Abbeel, P. Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization. In *International Conference on Machine Learning (ICML)*, 2016.

Fu, J., Singh, A., Ghosh, D., Yang, L., and Levine, S. Variational inverse control with events: A general framework for data-driven reward definition. In *Advances in Neural Information Processing Systems 31*, pp. 8538–8547, 2018.

Fujimoto, S., van Hoof, H., and Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. *arXiv preprint arXiv:1802.09477*, 2018.

Giusti, A., Guzzi, J. J., Cirean, D. C., He, F.-L., Rodríguez, J. P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J. J., Caro, G. D., Scaramuzza, D., Gambardella, L. M., Ciresan, D. C., He, F.-L., Rodriguez, J. P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J. J., Caro, G. D., Scaramuzza, D., and Gambardella, L. M. A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots. In *IEEE Robotics and Automation Letters.*, volume 1, pp. 2377–3766, 2015. ISBN 9781467380256. doi: 10.1109/LRA.2015.2509024.

Gu, S., Lillicrap, T., Sutskever, I., and Levine, S. Continuous Deep Q-Learning with Model-based Acceleration. In *International Conference on Machine Learning (ICML)*, 2016. ISBN 3405062780. doi: 10.3390/robotics2030122.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning*, 2018.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Dulac-Arnold, G., Osband, I., Agapiou, J., Leibo, J. Z., and Gruslys, A. Learning from Demonstrations for Real World Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, 2018.

Ho, J. and Ermon, S. Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Inoue, T., De Magistris, G., Munawar, A., Yokoya, T., and Tachibana, R. Deep reinforcement learning for high precision assembly tasks. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pp. 819–825. IEEE, 2017.

Johannink, T., Bahl, S., Nair, A., Luo, J., Kumar, A., Loskyll, M., Aparicio Ojea, J., Solowjow, E., and Levine, S. Residual Reinforcement Learning for Robot Control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M.,

Vanhoucke, V., and Levine, S. QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. In *Conference on Robot Learning (CoRL)*, 2018.

Kober, J. and Peter, J. Policy search for motor primitives in robotics. In *Advances in Neural Information Processing Systems (NIPS)*, volume 97, pp. 83–117, 2008. ISBN 1099401052236. doi: 10.1007/978-3-319-03194-1_4.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-End Training of Deep Visuomotor Policies. *Journal of Machine Learning Research (JMLR)*, 17(1):1334–1373, 2016. ISSN 15337928. doi: 10.1007/s13398-014-0173-7.2.

Levine, S., Pastor, P., Krizhevsky, A., and Quillen, D. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection. *International Journal of Robotics Research*, 2017.

Li, R., Platt, R., Yuan, W., Ten Pas, A., Roscup, N., Srinivasan, M. A., and Adelson, E. Localization and Manipulation of Small Parts Using GelSight Tactile Sensing. In *International Conference on Intelligent Robots and Systems (IROS)*, 2014.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2016. ISBN 0-7803-3213-X. doi: 10.1613/jair.301.

Luo, J., Solowjow, E., Wen, C., Aparicio Ojea, J., Agogino, A., Tamar, A., and P, A. Reinforcement learning on variable impedance controller for high-precision robotic assembly. In *Robotics and Automation (ICRA), 2019 IEEE International Conference on*. IEEE, 2019.

Mayton, B., LeGrand, L., and Smith, J. R. Robot, feed thyself: Plugging in to unmodified electrical outlets by sensing emitted ac electric fields. pp. 715–722, 2010.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing Atari with Deep Reinforcement Learning. In *NIPS Workshop on Deep Learning*, pp. 1–9, 2013. ISBN 1476-4687 (Electronic) 0028-0836 (Linking). doi: 10.1038/nature14236.

Nair, A., Mcgrew, B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Overcoming Exploration in Reinforcement Learning with Demonstrations. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., and Kawato, M. Learning from demonstration and adaptation of biped locomotion. In *Robotics and Autonomous Systems*, volume 47, pp. 79–91, 2004. ISBN 0921-8890. doi: 10.1016/j.robot.2004.03.003.

Ng, A. Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*, 1999.

Pathak, D., Mahmoudieh, P., Luo, G., Agrawal, P., Chen, D., Shentu, Y., Shelhamer, E., Malik, J., Efros, A. A., and Darrell, T. Zero-Shot Visual Imitation. In *International Conference on Learning Representations (ICLR)*, 2018.

Peters, J. and Schaal, S. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697, 2008. ISSN 08936080. doi: 10.1016/j.neunet.2008.02.003.

Peters, J., Mülling, K., and Altün, Y. Relative Entropy Policy Search. In *AAAI Conference on Artificial Intelligence*, pp. 1607–1612, 2010.

Pinto, L. and Gupta, A. Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours. *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

Pomerleau, D. A. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 305–313, 1989. ISBN 1-558-60015-9.

Pong, V., Gu, S., Dalal, M., and Levine, S. Temporal Difference Models: Model-Free Deep RL For Model-Based Control. In *International Conference on Learning Representations (ICLR)*, 2018.

Rajeswaran, A., Kumar, V., Gupta, A., Schulman, J., Todorov, E., and Levine, S. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Robotics: Science and Systems*, 2018.

Rhinehart, N. and Kitani, K. M. First-person activity forecasting with online inverse reinforcement learning. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Silver, T., Allen, K., Tenenbaum, J., and Kaelbling, L. Residual Policy Learning. dec 2018.

Tamar, A., Thomas, G., Zhang, T., Levine, S., and Abbeel, P. Learning from the hindsight plan episodic mpc improvement. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 336–343, 2017.

Thomas, G., Chien, M., Tamar, A., Ojea, J. A., and Abbeel, P. Learning Robotic Assembly from CAD. In *IEEE*

International Conference on Robotics and Automation (ICRA), 2018.

Van Hasselt, H., Guez, A., and Silver, D. Deep Reinforcement Learning with Double Q-learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2016. ISBN 1509.06461v3.

Večerík, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., and Riedmiller, M. Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards. *CoRR*, abs/1707.0, 2017.

Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K. Maximum Entropy Inverse Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, pp. 1433–1438, 2008. ISBN 9781577353683 (ISBN).