FORGE: Supplementary Material

I. Related Work

A. Real World Reinforcement Learning

A large body of work focuses on learning assembly tasks directly on the real-robot. Learning directly on the robot side-steps the *sim-to-real* gap by using data (and contact-interactions) from the same distribution expected at deployment. These works typically address problem of data efficiency by leveraging demonstrations [1], [2], [3], [4], [5] or using model-based approaches [6], [7], [8], [9]. To ensure excessive forces are not exceeded during training, these papers typically use control methods designed to be safe [5], [10], [11].

B. Sim-to-Real Methods

Learning directly in simulation is often preferable for robot safety, increased task variability, and access to privileged state. With advancements in RL and parallelizable simulation [12], [13], [14], [15], there has been much interest in *sim-to-real* transfer for complex control problems. Of note include legged locomotion [16], [17], [18], [19] and in-hand manipulation [20], [21], [22].

Recent advances in contact-rich simulation has enabled efficient simulation of assembly tasks [23], [24], [25], [26]. However, as discussed throughout the paper, the key challenge becomes the sim-to-real gap: how can we *safely* and *successfully* deploy policies that were trained in simulation?

Although *system identification* is a principled approach to minimize the *sim-to-real* gap [27], it is often time-consuming and difficult to apply to contact-rich tasks [28], [29]. Instead, many approaches use *dynamics randomization*: randomizing parameters such as part friction/stiffness [30], [31], [32], [33], [34], controller gains [35], [33], or F/T observation scale [35], [36]. A third class of approaches uses a small amount of real-robot data to finetune the policies learned in simulation [31], [37], [38].

Even with dynamics randomization, excessive forces can occur when deployed on the real-robot. In some work, parts are fixed to the gripper [32] and slip will not occur. When this isn't the case, several methods have been proposed to ensure safe policy deployment. An expert can tune the controller gains at deployment time or design the action space in a way such that all actions are safe [39], [40], [41]. Gains can also be adapted online via optimization [42] or an explicit *gain-tuning* model [36].

Similar to FORGE, other works have proposed to use a force-threshold [31], [43], [32]. These works have a fixed threshold during training which is often very large to primarily prevent damage (e.g., $40N$). However, especially with small parts, slip can occur with much lower contact forces. Most similar to FORGE, [31] introduces a method to specify the *desired* interaction force at deployment time.

Most prior work focus on insertion-style tasks. We show how the combined application of a force-threshold and dynamics randomization can lead to robust sim-to-real transfer for a range of tasks, including the complicated nut-threading task. Prior work on sim-to-real for nut-threading [44] focused on large parts ($M48$ nuts) that were fixed to the gripper. In addition, we show these techniques are applicable for sim-to-real transfer of early termination procedures.

II. Randomization

All randomization ranges are reported in Table I. In addition to the dynamics randomization described in the text, we also randomize the initial state distribution and observation noise.

Initial State Randomization: At the start of an episode, we randomize the position of the fixed part, the relative pose of the hand above the fixed part, and the relative position of the held part in the gripper (where the default position has the top of the held part aligned with the bottom of the gripper).

Observation Randomization: In simulation, the position of the fixed asset is randomized once per episode by adding Gaussian noise. Independent Gaussian noise is added to each observation at every timestep (except velocity, where positional noise is propagated through finite differencing).

III. Reward

A. Keypoint Reward

Here we describe the keypoint reward in more details. Keypoint distance is calculated as: $d_t^{kp}(p_t^{held}, p^{fixed}) = ||k_t^{held} - k^{targ}||$. We use a logistic kernel as in [20] to transform keypoint distances into a bounded reward: $\mathcal{K}_{a,b}(d_{kp}) = (e^{-ax} + b + e^{ax})^{-1}$. The kernel can be tuned to be sensitive to distances at different scales using parameters a and b (see Table I).

Using a single kernel parameterization was not sufficient for the nut-threading task due to small geometry. Different phases of the task require motion at different scales. For example, initial placement of the nut on the bolt requires movement ranging from $0 - 2cm$. However, lowering the nut by the final thread changes the position by $< 1mm$. Instead, we propose a *coarse-to-fine* keypoint reward. The final reward is a sum of: (1) A *coarse reward* directing the arm towards the tip of the fixed part and; (2) a *fine reward* incentivizing more detailed motion once the arm is close to

Initial State Randomization			
Parameter	All Tasks		
Fixed: x, y, z	$[0.55, 0.65]m, [-0.05, 0.05]m, [0.0, 0.1]m$		
Hand: x, y (rel)	$[-2, 2]cm, [-2, 2]cm$		
Held: x, y (rel)	$[-3, 3]mm, [0, 0]mm$		
Parameter	$8mm$ Peg	Medium Gear	$M16$ Nut
Hand: z (rel)	$[3.7, 5.7]cm$	$[2.5, 4.5]cm$	$[0.5, 2.5]cm$
Hand: yaw	$[-45, 45]^\circ$	$[-45, 45]^\circ$	$[-120, -90]^\circ$
Held: z (rel)	$[14, 20]mm$	$[12, 15]mm$	$[10, 16]mm$
Observation Randomization			
Parameter	$8mm$ Peg	Medium Gear	$M16$ Nut
Pos-Est Noise	$2.5mm$	$2.5mm$	$2.5mm$
Force Noise	$1N$	$1N$	$1N$
EE-Pos. Noise	$0.25mm$	$0.25mm$	$0.25mm$
Dynamics Randomization			
Parameter	$8mm$ Peg	Medium Gear	$M16$ Nut
Part Friction	$[0.5, 1.0]$	$[0.38, 0.75]$	$[0.1, 0.38]$
Controller Gains	$[400, 800]$	$[400, 800]$	$[400, 800]$
Action Scale: λ	$[1.6, 2.5]cm$	$[1.6, 2.5]cm$	$[1.6, 2.5]cm$
Dead Zone	$[0, 5]N$	$[0, 5]N$	$[0, 5]N$
Force Threshold	$[5, 10]N$	$[5, 10]N$	$[5, 10]N$
Reward Specification			
Parameter	$8mm$ Peg	Medium Gear	$M16$ Nut
Coarse (a^c, b^c)	$(50, 2)$	$(50, 2)$	$(100, 2)$
Fine: (a^f, b^f)	$(100, 0)$	$(100, 0)$	$(500, 0)$
Contact-Pen: β	0.2	0.05	0.05
Success Dist.	$24mm$	$19mm$	$2.5mm$
Place Dist.	$2.5mm$	$2mm$	$2.5mm$
Episode Length	150 ($10s$)	300 ($20s$)	450 ($30s$)

TABLE I

SIMULATION PARAMETERS USED TO TRAIN FORGE POLICIES.

the part. These are implemented using different parameters for the logistic kernel,

$$R_{kp}(p^{fixed}, p_t^{held}) = \mathcal{K}_{a^c,b^c}^{coarse}(d_t^{kp}) + \mathcal{K}_{a^f,b^f}^{fine}(d_t^{kp}). \quad (1)$$

Parameters for each task can be found in Table I.

B. Task Success

Each task defines success based on the relative positions between the held and fixed parts (Table I shows *Success Dist.* as the distance between the top of the fixed part and bottom of the held part when success is achieved):

- *Peg Insertion*: The bottom of the peg is within $1mm$ of the base of the socket (equivalently, $24mm$ below the top of the socket).
- *Gear Meshing*: The bottom of the gear is within $1mm$ of the base of the gear plate (equivalently, $19mm$ below the tip of the gear peg).
- *Nut Threading*: The $M16$ nut is lowered a quarter thread (corresponding to $2.5mm$ below the tip of the bolt, as the first thread is chamfered).

For all tasks, success also requires the parts to be laterally centered.

IV. PLANETARY GEARBOX

For the planetary gearbox, we trained policies for the following tasks: Ring Insertion, Small Gear Meshing, Large Gear Meshing, and M16 Nut Threading.

Gear Tasks: The small and large gear meshing tasks had one abutting gear in simulation. This is similar to deployment for the small gear which achieved a high success rate ($15/15$). However, when the large gear is deployed, it needs to mesh with the three already inserted small gears. This is much harder than how the policy was trained and could be a cause of the performance drop for this task ($3/5$).

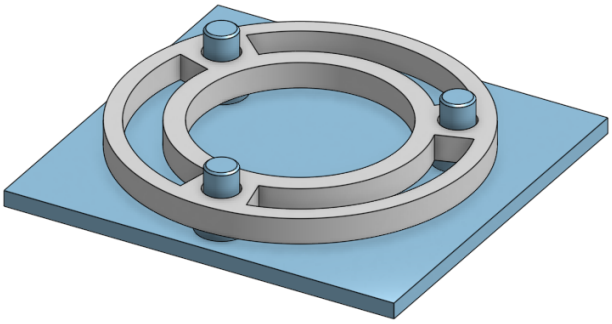

Fig. 1. Simulated assets for the ring insertion task. The ring gear (grey) is inserted onto the gearbox plate (blue).

Ring Insertion: The outer ring gear must be inserted onto the three bolts of the gearbox base. We designed simulation assets for the corresponding parts (see Fig. 1) and trained a policy using the FORGE framework. We assume there is small orientation error on the ring ($< 5^\circ$) during training. Success is defined as having the ring gear placed close to the gearbox base ($< 2mm$ displacement) and all three bolt holes aligned.

Gearbox Design: Note, we also designed a "lock" for the gear carrier which is removed by the robot after the small gears are inserted. This ensures a fixed base during the small gear insertions (see video).

Policy Selection: The M16 policy was chosen as the best policy from our main evaluation (FORGE No Force). All other policies were trained using the FORGE framework including force observations. We trained one policy per task without any additional checkpoint selection procedure. For the gearbox experiments only, we selected high control stiffness for the roll and pitch dimensions of the impedance controller, as the policy does not generate actions for these degrees of freedom.

Task Execution: To pick up the held parts, we assume a known grasp location which was predetermined (with small noise from placement error). However, the location of the corresponding fixed parts were estimated from the *IndustReal* perception system [41]. Grasping and movement to the initial state for policy execution was performed with a standard position controller. No additional artificial noise or initial-state randomization was added for the gearbox experiments.

REFERENCES

[1] F. J. Abu-Dakka, L. Rozo, and D. G. Caldwell, "Force-based learning of variable impedance skills for robotic manipulation," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9.

[2] T. Davchev, K. S. Luck, M. Burke, F. Meier, S. Schaal, and S. Ramamoorthy, "Residual Learning From Demonstration: Adapting DMPs for Contact-Rich Manipulation," *IEEE RA-L*, 2022.

[3] J. Luo, O. Sushkov, R. Pevceviciute, W. Lian, C. Su, M. Vecerik, N. Ye, S. Schaal, and J. Scholz, "Robust Multi-Modal Policies for Industrial Assembly via Reinforcement Learning and Demonstrations: A Large-Scale Study," in *RSS*, 2021.

[4] M. Vecerik, O. Sushkov, D. Barker, T. Rothörl, T. Hester, and J. Scholz, "A Practical Approach to Insertion with Variable Socket Position Using Deep Reinforcement Learning," in *ICRA*. IEEE, 2019.

[5] J. Luo, Z. Hu, C. Xu, Y. L. Tan, J. Berg, A. Sharma, S. Schaal, C. Finn, A. Gupta, and S. Levine, "SERL: A Software Suite for Sample-Efficient Robotic Reinforcement Learning," *arXiv:2401.16013*, 2024.

[6] J. Luo *et al.*, "Reinforcement Learning on Variable Impedance Controller for High-Precision Robotic Assembly," in *ICRA*. IEEE, 2019.

[7] Y. Fan, J. Luo, and M. Tomizuka, "A Learning Framework for High Precision Industrial Assembly," in *ICRA*. IEEE, 2019.

[8] M. A. Lee, C. Florensa, J. Tremblay, N. Ratliff, A. Garg, F. Ramos, and D. Fox, "Guided Uncertainty-Aware Policy Optimization: Combining Learning and Model-Based Strategies for Sample-Efficient Policy Learning," in *ICRA*. IEEE, 2020.

[9] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, and A. M. Agogino, "Deep Reinforcement Learning for Robotic Assembly of Mixed Deformable and Rigid Objects," in *IROS*. IEEE, 2018.

[10] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, and R. Tachibana, "Deep reinforcement learning for high precision assembly tasks," in *IROS*. IEEE, 2017.

[11] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making Sense of Vision and Touch: Learning Multimodal Representations for Contact-Rich Tasks," *IEEE T-RO*, 2020.

[12] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[13] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[14] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019. [Online]. Available: https://drake.mit.edu

[15] V. Makoviychuk *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv:2108.10470*, 2021.

[16] J. Hwangbo *et al.*, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, 2019.

[17] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged Locomotion in Challenging Terrains using Egocentric Vision," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2022.

[18] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid Locomotion via Reinforcement Learning," in *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

[19] N. Rudin, D. Hoeller, M. Hutter, and P. Reist, "Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2021.

[20] A. Allshire *et al.*, "Transferring dexterous manipulation from gpu simulation to a remote real-world trifinger," in *IROS*. IEEE, 2022.

[21] I. Akkaya *et al.*, "Solving rubik's cube with a robot hand," *arXiv:1910.07113*, 2019.

[22] A. Handa *et al.*, "DeXtreme: Transfer of Agile In-hand Manipulation from Simulation to Reality," in *ICRA*. IEEE, 2023.

[23] L. Lan, D. M. Kaufman, M. Li, C. Jiang, and Y. Yang, "Affine body dynamics: Fast, stable & intersection-free simulation of stiff materials," *arXiv:2201.10022*, 2022.

[24] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and Z. Corse, "Local optimization for robust signed distance field collision," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 3, no. 1, pp. 1–17, 2020.

[25] Y. Narang *et al.*, "Factory: Fast Contact for Robotic Assembly," in *RSS*, 2022.

[26] J. Yoon, M. Lee, D. Son, and D. Lee, "Fast and Accurate Data-Driven Simulation Framework for Contact-Intensive Tight-Tolerance Robotic Assembly Tasks," *arXiv:2202.13098*, 2022.

[27] L. Ljung, "System identification," in *Signal analysis and prediction*. Springer, 1998, pp. 163–173.

[28] B. Acosta, W. Yang, and M. Posa, "Validating robotics simulators on real-world impacts," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6471–6478, 2022.

[29] M. Guo, Y. Jiang, A. E. Spielberg, J. Wu, and K. Liu, "Benchmarking Rigid Body Contact Models," in *LDCC*, 2023.

[30] A. Apolinarska *et al.*, "Robotic assembly of timber joints using reinforcement learning," *Automation in Construction*, 2021.

[31] C. Beltran-Hernandez, D. Petit, I. Ramirez-Alpizar, and K. Harada, "Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach," *Applied Sciences*, 2020.

[32] M. Hebecker, J. Lambrecht, and M. Schmitz, "Towards Real-World Force-Sensitive Robotic Assembly through Deep Reinforcement Learning in Simulations," in *AIM*. IEEE, 2021.

[33] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-Real Transfer of Robotic Control with Dynamics Randomization," in *ICRA*. IEEE, 2018.

[34] O. Spector and M. Zacksenhouse, "Learning Contact-Rich Assembly Skills Using Residual Admittance Policy," in *IROS*. IEEE, 2021.

[35] S. Jin, X. Zhu, C. Wang, and M. Tomizuka, "Contact Pose Identification for Peg-in-Hole Assembly under Uncertainties," in *ACC*. IEEE, 2021.

[36] X. Zhang, M. Tomizuka, and H. Li, "Bridging the Sim-to-Real Gap with Dynamic Compliance Tuning for Industrial Insertion," in *ICRA*. IEEE, 2024.

[37] G. Schoettler and et al., "Meta-reinforcement learning for robotic industrial insertion tasks," in *IROS*. IEEE, 2020.

[38] Z. Wu, Y. Xie, W. Lian, C. Wang, Y. Guo, J. Chen, S. Schaal, and M. Tomizuka, "Zero-shot policy transfer with disentangled task representation of meta-reinforcement learning," in *ICRA*. IEEE, 2023.

[39] N. Vuong, H. Pham, and Q. Pham, "Learning Sequences of Manipulation Primitives for Robotic Assembly," in *ICRA*. IEEE, 2021.

[40] K. Zhang, M. Sharma, J. Liang, and O. Kroemer, "A modular robotic arm control stack for research," *arXiv:2011.02398*, 2020.

[41] B. Tang *et al.*, "IndustReal: Transferring Contact-Rich Assembly Tasks from Simulation to Reality," in *RSS*, 2023.

[42] X. Zhang, C. Wang, L. Sun, Z. Wu, X. Zhu, and M. Tomizuka, "Efficient Sim-to-real Transfer of Contact-Rich Manipulation Skills with Online Admittance Residual Learning," in *CORL*, 2023.

[43] R. Martín-Martín, M. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg, "Variable impedance control in end-effector space: An action space for reinforcement learning," in *IROS*. IEEE, 2019.

[44] D. Son, H. Yang, and D. Lee, "Sim-to-Real Transfer of Bolting Tasks with Tight Tolerance," in *IROS*. IEEE, 2020.