Interventional Data Generation for Robust and Data-Efficient Robot Imitation Learning

Ryan Hoque[1,2], **Ajay Mandlekar***[2], **Caelan Garrett***[2], **Ken Goldberg**[1], **Dieter Fox**[2]

[1]UC Berkeley [2]NVIDIA

Abstract: Imitation learning is a promising paradigm for training robot control policies, but these policies can suffer from distribution shift, where the conditions at evaluation time differ from those in the training data. One common real-world source of distribution shift is object pose estimation error, which can cause agents that rely on pose information to fail catastrophically during deployment. A popular approach for increasing policy robustness to distribution shift is interactive imitation learning, in which a human operator provides corrective interventions during policy deployment. However, collecting a sufficient amount of interventions to cover the distribution of policy mistakes can be burdensome for human operators. We propose Interventional MimicGen (I-MG), a novel data generation system that can autonomously generate a large set of corrective interventions with rich coverage of the state space from a small number of human interventions. We apply I-MG to policies deployed under object pose estimation error and show that it can increase policy robustness by up to 39× with only 10 human interventions. Videos and more are available at `https://sites.google.com/view/interventional-mimicgen`.

1 Introduction

Imitation Learning (IL) from human demonstrations is a leading paradigm for training robot policies. One popular approach is to collect a large set of offline task demonstrations via human teleoperation [1, 2] and employ behavior cloning (BC) [3] to train robot policies via supervised learning, where the labels are robot actions. Inspired by the dramatic recent success of large-scale vision and language models [4, 5, 6, 7, 8, 9], there have been recent efforts to scale this approach by collecting thousands of demonstrations using hundreds of human operator hours and training high-capacity neural networks on the large-scale data [10, 11, 12, 13, 14].

However, IL policies can suffer from distribution shift, where the conditions at evaluation time differ from those in the training data [15]. As an example, consider a policy that makes decisions based on object pose observations. A common source of distribution shift in the real world is object pose estimation error, which can occur due to a wide range of factors such as sensor noise, occlusion, network delay, and model misspecification. This can cause inaccuracy in the robot's belief of where critical objects are located in the environment, leading the robot to visit states outside the training distribution that result in poor policy performance.

One approach to addressing distribution shift is to collect a large set of demonstrations under diverse conditions and hope that agents trained on this data can generalize. However, human teleoperation data is notoriously difficult to collect due to the human time, effort, and financial cost required [10, 11, 12, 13, 14]. An alternative approach is interactive IL (i.e., DAgger [15] and variants [16, 17, 18]), where humans can intervene during robot execution and demonstrate *recovery behaviors* to help the robot return to the support of the training distribution. Subsequent

*Equal contribution.

7th Conference on Robot Learning (CoRL 2023), Atlanta, USA.

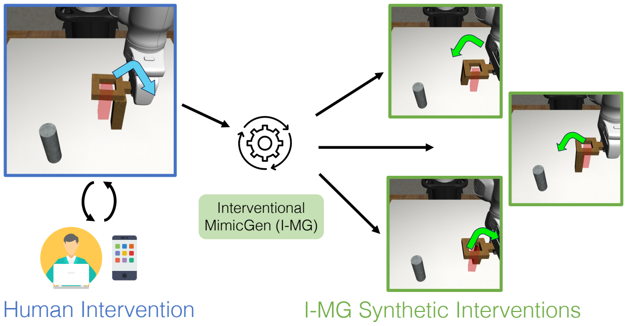

Figure 1: **Overview.** Interventional MimicGen automatically generates corrective interventional data from a handful of human interventions, with coverage across both diverse scene configurations and policy mistake distributions. Here, the robot mistakenly believes the peg is at the position highlighted in red and requires demonstration of recovery behavior toward the true peg position.

training on these corrections can increase policy robustness and performance both theoretically and in practice [15]. However, human-gated interactive IL [16, 17] imposes even more burden on the human supervisors than behavior cloning, as the human must continuously monitor robot task execution and intervene when they see fit, typically over multiple rounds of interleaved data collection and policy training. Moreover, a significant amount of recovery data may be required to adequately cover the distribution of mistakes the policy may make.

We raise the following question: do we actually need to have a human operator collect corrections every single time a policy makes a mistake? MimicGen [19], a recently proposed data generation system, raises an intriguing possibility: a large dataset of synthetically generated demonstrations derived from a small set of human demonstrations (typically 100× smaller or more) can produce performant robot policies. The system's key insight is that similar object-centric manipulation behaviors can be applied in new contexts by appropriately transforming demonstrated behavior to the new object frame. We propose a similar strategy for interventional data (see Fig. 1): with a small set of corrective interventions from a human operator, we can autonomously generate data with significantly higher coverage of the distribution of potential policy mistakes. Naïve application of MimicGen, however, is insufficient for addressing technical challenges in the interventional setting such as variation in not only object poses but also the robot's incorrect estimates of these poses.

This paper makes the following contributions: (1) Interventional MimicGen (I-MG), a system for automatically generating interventional data across diverse scene configurations and broad mistake distributions from only a handful of human interventions. **(2)** Application of I-MG to robustness against 2 sources of object pose estimation error (sensor noise and geometry error) in 5 high-precision 6-DOF manipulation tasks. I-MG dramatically increases policy robustness by up to 39× with only 10 human interventions. **(3)** Experiments demonstrating the utility of I-MG over alternate uses of a human data budget of equivalent or even greater size. A policy trained on synthetic I-MG data from 10 source human interventions can outperform one trained on even 100 human interventions by 24%, with a fraction of the data collection time and effort.

For space considerations we move discussion of related work, preliminaries and assumptions, experiment setup details (including task and baseline descriptions), and some of the experiments (including sim-to-real evaluation) to Appendix 4.1, 4.2, 4.4, and 4.5 respectively.

2 Interventional MimicGen

2.1 Interventional Data Collection

Rather than the full human task demonstrations considered by MimicGen, the input to I-MG consists of interventional demonstrations, in which control alternates between the autonomous robot policy π_θ and human teleoperator π_H. We consider human-gated interventions [16], in which the human

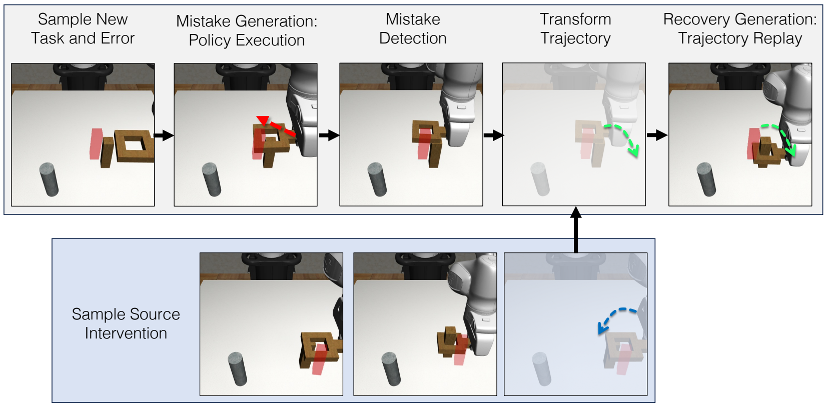
Sample New Task and Error
Mistake Generation: Policy Execution
Mistake Detection
Transform Trajectory
Recovery Generation: Trajectory Replay
Sample Source Intervention

Figure 2: **I-MG Data Generation Example.** We provide an example of how I-MG generates a new intervention. First, a new task instance is sampled with a new configuration (square peg location) and observation corruption (incorrect peg location highlighted in red). We execute the robot policy to generate mistake behavior for the new task instance. When a mistake is detected, we sample a human intervention segment from the source dataset and transform it to adapt to the current scene. Finally, we executed the transformed recovery segment.

monitors the robot policy execution and intermittently takes control to correct policy mistakes. As in DAgger [15], this enables the human to demonstrate corrective recovery behavior from mistakes made by the robot policy that otherwise would not be visited in full human task demonstrations (due to distribution shift). The base robot policy π_θ executed during interventional data collection can come from anywhere, but is typically initialized from behavior cloning on an initial set of offline task demonstrations D [18, 17, 15]. Each collected trajectory can be coarsely divided into robot-generated "mistake" segments and human-generated "recovery" segments.

2.2 Mistake Generation: Closed-Loop Policy Execution

We aim to use the collected human interventions to automatically synthesize interventions for new scene configurations. Recall that MimicGen generates data by decomposing the task into object-centric subtasks, transforming each subtask trajectory with respect to new object poses, and executing the transformed trajectory in an open-loop manner. However, an appealing property of the interventional IL setting is access to the robot policy π_θ that is executed during data collection.

A key insight in this work is that the same robot policy can be used not only during data collection but during the *data generation* process as well. Instead of open-loop replay of a robot mistake trajectory in the source dataset, we can instead execute the policy in the new scene configuration. This has two benefits: (1) rather than assuming the policy will fail in the same manner as the source trajectory, the generated mistake will reflect the genuine behavior of the policy in the new configuration, and (2) it becomes possible to generate new mistake trajectories for new corruptions of the observed object poses. For example, if sensor noise corrupts the object pose during interventional data collection, a new noise corruption can be applied during the data generation process. This allows data diversity in both object poses and the robot's erroneous beliefs about where the objects are (see Fig. 2). However, the use of policy execution during data generation comes with two assumptions: (1) access to a state classifier that determines when to terminate policy execution, and (2) some subset of the source recovery trajectories can be successfully applied to new mistake states. In our experiments, we use MuJoCo contact detection [20] for Assumption (1); a more flexible option could be a learned classifier or robot-gated intervention criteria such as ThriftyDAgger [18].

2.3 Recovery Generation: Open-Loop Trajectory Replay

In each episode of synthetic data generation, once we have completed policy execution and entered a new mistake state, we proceed with generating a recovery trajectory. We select a random source trajectory, segment out the human recovery portion of the trajectory, and adapt the trajectory to the current environment state. As in MimicGen, this adaptation consists of (1) transforming the source

Dataset	Nut Insertion	2-Pc Assembly	Coffee
Base	22%	6%	2%
Source Int	40%	6%	10%
Weighted Src Int [17]	50%	16%	6%
Source Demo	42%	12%	12%
MG Demo [19]	64%	16%	18%
I-MG - Policy (Ours)	86%	52%	42%
I-MG (Ours)	**98%**	**70%**	**80%**

Table 1: Results for 3 high-precision manipulation tasks in MuJoCo with noisy pose estimation.

trajectory to the current object pose, (2) linearly interpolating in end-effector space to the beginning of the transformed trajectory, and (3) executing the transformed trajectory open-loop (see Fig. 2). Note that each object-centric subtask may have zero, one, or multiple instances of mistake and recovery.

2.4 Output Filtering and Dataset Aggregation

We only keep the generated demonstration if it successfully completes the task, as in [19]. We also filter out the segment of the synthetic demonstration that corresponds to the human recovery segment; such filtering is used by common algorithms such as DAgger [15] and HG-DAgger [16] and can prevent the imitation of mistakes. Each filtered episode of synthetic data is then aggregated into the base dataset D (used to train the base policy π_θ), and the policy is retrained on the new dataset after data generation is complete. If desired, the entire process of data collection, data generation, and policy training can be iterated. See Appendix 4.3 for the full pseudocode for I-MG.

3 Experiments

In this section we summarize the key takeaways from the comparisons presented in Table 1. Many additional experiments including real robot evalutions are available in Appendix 4.5.

I-MG vastly improves policy robustness under pose estimation error. In Table 1, we observe that I-MG improves policy performance by 3.5×, 10.7×, and 39× over the base policy in Nut Insertion, 2-Piece Assembly, and Coffee respectively, despite only collecting 10 human interventions.

I-MG significantly improves upon naïve uses of an equivalent amount of full human demonstration data. I-MG consistently outperforms human demonstrations collected at test time (Source Demo, Table 1) by 56%-68%. Even if these demonstrations are expanded by 100× with MimicGen (MG Demo), I-MG still outperforms by 34%-62%. Since the human's observability does not match that of the robot, the human can teleoperate toward the true object poses. As a result, the robot does not observe any recovery behavior in the offline data.

I-MG significantly improves upon naïve uses of an equivalent amount of interventional human data. Source Int in Table 1 underperforms I-MG by 58%-70%. While helpful, with only 10 human interventions, the data is insufficient to learn robust recovery under pose error. This remains the case even if the intervention data is weighted higher, in which case the agent overfits to the 10 interventions and underperforms I-MG by 48%-74%. With the same budget of interventional human data, I-MG can generate much richer coverage of the distribution of mistakes under the base policy.

I-MG significantly improves upon naïve uses of MimicGen. We observe a significant 34%-62% improvement over MimicGen on full task demonstrations (MG Demo, Table 1). We also observe that the policy execution component (Section 2.2) boosts performance by 12%-38% respectively over the ablation. While the ablation dataset covers variation in the object pose, it does not cover variation in the error; only the 10 mistake segments in the source dataset are available. This shows that the novel components we introduced in I-MG are crucial for high performance.

I-MG is useful across different environments. While 2-Piece Assembly and Coffee have narrower tolerance regions than Nut Insertion that lower success rates across the board, the relative performance of I-MG remains consistent across environments: I-MG outperforms all baselines by 12%-76% in Nut Insertion, 18%-64% in 2-Pc Assembly, and 38%-78% in Coffee.

References

[1] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei. RoboTurk: A Crowdsourcing Platform for Robotic Skill Learning through Imitation. In *Conference on Robot Learning*, 2018.

[2] A. Mandlekar, J. Booher, M. Spero, A. Tung, A. Gupta, Y. Zhu, A. Garg, S. Savarese, and L. Fei-Fei. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. *arXiv preprint arXiv:1911.04052*, 2019.

[3] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. Touretzky, editor, *Neural Information Processing Systems (NeurIPS)*, volume 1. Morgan-Kaufmann, 1988.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[6] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[9] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[10] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, 2021.

[11] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, et al. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023.

[12] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

[13] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[14] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. K. Baruch, T. Armstrong, and P. R. Florence. Interactive language: Talking to robots in real time. *ArXiv*, abs/2210.06407, 2022.

[15] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 627–635, 2011.

[16] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083, 2018.

[17] A. Mandlekar, D. Xu, R. Martin-Martin, Y. Zhu, L. Fei-Fei, and S. Savarese. Human-in-the-loop imitation learning using remote teleoperation. *ArXiv*, abs/2012.06733, 2020.

[18] R. Hoque, A. Balakrishna, E. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg. ThriftyDAgger: Budget-aware novelty and risk gating for interactive imitation learning. In *Conference on Robot Learning (CoRL)*, 2021.

[19] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning (CoRL)*, 2023.

[20] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5026–5033, 10 2012. ISBN 978-1-4673-1737-5. doi:10.1109/IROS.2012.6386109.

[21] P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. In *International Journal of Robotics Research (IJRR)*, 2017.

[22] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *Robotics and Automation (ICRA), 2016 IEEE Int'l Conference on*. IEEE, 2016.

[23] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2018.

[24] D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

[25] K.-T. Yu, M. Bauza, N. Fazeli, and A. Rodriguez. More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing. In *Int'l Conference on Intelligent Robots and Systems*, 2016.

[26] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.

[27] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[28] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan. VIMA: General robot manipulation with multimodal prompts. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[29] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee. Transporter networks: Rearranging the visual world for robotic manipulation. *Conference on Robot Learning (CoRL)*, 2020.

[30] M. Dalal, A. Mandlekar, C. Garrett, A. Handa, R. Salakhutdinov, and D. Fox. Imitating task and motion planning with visuomotor transformers. *arXiv preprint arXiv:2305.16309*, 2023.

[31] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023.

[32] M. J. McDonald and D. Hadfield-Menell. Guided imitation of task and motion planning. In *Conference on Robot Learning*, pages 630–640. PMLR, 2022.

[33] A. Tung, J. Wong, A. Mandlekar, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese. Learning multi-arm manipulation through collaborative teleoperation. *arXiv preprint arXiv:2012.06738*, 2020.

[34] J. Wong, A. Tung, A. Kurenkov, A. Mandlekar, L. Fei-Fei, S. Savarese, and R. Martín-Martín. Error-aware imitation learning from teleoperation data for mobile manipulation. In *Conference on Robot Learning*, pages 1367–1378. PMLR, 2022.

[35] A. Mandlekar, C. R. Garrett, D. Xu, and D. Fox. Human-in-the-loop task and motion planning for imitation learning. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=G_FEL3OkiR.

[36] 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. *arXiv preprint arXiv:2103.11512*, 2021.

[37] H. Liu, S. Nasiriany, L. Zhang, Z. Bao, and Y. Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. *arXiv*, abs/2211.08416, 2022.

[38] T. Zhang, Z. McCarthy, O. Jow, D. Lee, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. *arXiv preprint arXiv:1710.04615*, 2017.

[39] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei. Learning to generalize across long-horizon tasks from human demonstrations. *arXiv preprint arXiv:2003.06085*, 2020.

[40] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martin-Martin. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021.

[41] S. Chernova and M. Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34:1–25, 2009.

[42] M. Laskey, S. Staszak, W. Y.-S. Hsieh, J. Mahler, F. T. Pokorny, A. D. Dragan, and K. Goldberg. Shiv: Reducing supervisor burden in dagger using support vectors for efficient learning from demonstrations in high dimensional state spaces. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 462–469, 2016.

[43] B. Packard and S. Ontanón. Policies for active learning from demonstration. In *AAAI Spring Symposium Series*, 2017.

[44] J. Zhang and K. Cho. Query-efficient imitation learning for end-to-end autonomous driving. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[45] K. Menda, K. Driggs-Campbell, and M. J. Kochenderfer. EnsembleDAgger: A Bayesian Approach to Safe Imitation Learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[46] C. Cronrath, E. Jorge, J. Moberg, M. Jirstrand, and B. Lennartson. Bagger: A bayesian algorithm for safe and query-efficient imitation learning. In *Machine Learning in Robot Motion Planning–IROS 2018 Workshop*, 2018.

[47] Y. Cui, D. Isele, S. Niekum, and K. Fujimura. Uncertainty-aware data aggregation for deep imitation learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 761–767. IEEE, 2019.

[48] M. Laskey, C. Chuck, J. Lee, J. Mahler, S. Krishnan, K. Jamieson, A. Dragan, and K. Goldberg. Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations. In *International Conference on Robotics and Automation (ICRA)*, pages 358–365, 2017.

[49] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa. Learning from interventions: Human-robot interaction as both explicit and implicit feedback. In *Robotics: Science and Systems (RSS)*, 2020.

[50] J. DelPreto, J. I. Lipton, L. Sanneman, A. J. Fay, C. Fourie, C. Choi, and D. Rus. Helping robots learn: A human-robot master-apprentice model using demonstrations via virtual reality teleoperation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 10226–10233, 2020.

[51] A. Jevtić, A. Colomé, G. Alenya, and C. Torras. Robot motion adaptation through user intervention and reinforcement learning. *Pattern Recognition Letters*, 105:67–75, 2018.

[52] F. Wang, B. Zhou, K. Chen, T. Fan, X. Zhang, J. Li, H. Tian, and J. Pan. Intervention aided reinforcement learning for safe and practical policy optimization in navigation. In *Conference on Robot Learning*, pages 410–421. PMLR, 2018.

[53] V. G. Goecks, G. M. Gremillion, V. J. Lawhern, J. Valasek, and N. R. Waytowich. Efficiently combining human demonstrations and interventions for safe training of autonomous systems in real-time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2462–2470, 2019.

[54] J. Bi, T. Xiao, Q. Sun, and C. Xu. Navigation by imitation in a pedestrian-rich environment. *arXiv preprint arXiv:1811.00506*, 2018.

[55] T. Ablett, F. Marić, and J. Kelly. Fighting failures with fire: Failure identification to reduce expert burden in intervention-based learning. *arXiv preprint arXiv:2007.00245*, 2020.

[56] R. Hoque, A. Balakrishna, C. Putterman, M. Luo, D. S. Brown, D. Seita, B. Thananjeyan, E. Novoseller, and K. Goldberg. LazyDAgger: Reducing context switching in interactive imitation learning. In *IEEE Conference on Automation Science and Engineering (CASE)*, pages 502–509, 2021.

[57] R. Hoque, L. Y. Chen, S. Sharma, K. Dharmarajan, B. Thananjeyan, P. Abbeel, and K. Goldberg. Fleet-dagger: Interactive robot fleet learning with scalable human supervision. In *Conference on Robot Learning (CoRL)*, 2022.

[58] B. Wen, W. Lian, K. E. Bekris, and S. Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. In *Robotics: Science and Systems (RSS)*, 2022.

[59] E. Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4613–4619, 2021. URL https://api.semanticscholar.org/CorpusID:234482766.

[60] M. Laskey, J. Lee, R. Fox, A. Dragan, and K. Goldberg. Dart: Noise injection for robust imitation learning. *arXiv preprint arXiv:1703.09327*, 2017.

[61] D. Brandfonbrener, S. Tu, A. Singh, S. Welker, C. Boodoo, N. Matni, and J. Varley. Visual backtracking teleoperation: A data collection protocol for offline image-based reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11336–11342. IEEE, 2023.

[62] H. Nguyen, A. Baisero, D. Wang, C. Amato, and R. Platt. Leveraging fully observable policies for learning under partial observability. *arXiv preprint arXiv:2211.01991*, 2022.

[63] S. Choudhury, A. Kapoor, G. Ranade, and D. Dey. Learning to gather information via imitation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 908–915. IEEE, 2017.

[64] G. Cideron, B. Tabanpour, S. Curi, S. Girgin, L. Hussenot, G. Dulac-Arnold, M. Geist, O. Pietquin, and R. Dadashi. Get back here: Robust imitation by return-to-distribution planning. *arXiv preprint arXiv:2305.01400*, 2023.

[65] S. Haldar, J. Pari, A. Rai, and L. Pinto. Teach a robot to fish: Versatile imitation from one minute of demonstrations. *arXiv preprint arXiv:2303.01497*, 2023.

[66] A. Peng, A. Netanyahu, M. K. Ho, T. Shu, A. Bobu, J. Shah, and P. Agrawal. Diagnosis, feedback, adaptation: A human-in-the-loop framework for test-time policy adaptation. 2023.

[67] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[68] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

[69] A. Mandlekar, Y. Zhu, A. Garg, L. Fei-Fei, and S. Savarese. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3932–3939. IEEE, 2017.

[70] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3):4915–4922, 2021.

[71] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

[72] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

4 Appendix

4.1 Related Work

Data Collection Approaches for Robot Learning. Many prior works address the need for large-scale data in robotics. Some use self-supervised data collection [21, 22, 23, 24, 25, 26], but the data can have low signal-to-noise ratio due to the trial-and-error process. Other works collect large datasets using experts that operate on privileged information available in simulation [27, 28, 29, 30, 31, 32]. Still, designing such experts can require significant engineering. One popular approach is to collect demonstrations by having human operators teleoperate robot arms [1, 2, 17, 33, 34, 11, 12, 10, 13, 14, 35]; however, this can require hundreds of hours of human operator time. Some systems also allow for collecting interventions to help correct policy mistakes [36, 17, 37]. In this work, we make effective use of a handful of interventional corrections provided by a single human operator to autonomously generate large-scale interventional data, substantially reducing the operator burden.

Imitation Learning from Human Demonstrations. Behavioral Cloning (BC) [3] on demonstrations collected using robot teleoperation with human operators has shown remarkable performance in solving real-world robot manipulation tasks [38, 39, 40, 11, 10, 13]. However, scaling this paradigm can be costly due to the need for large amounts of data, requiring many hours of human operator time [11, 10, 14]. Furthermore, policies trained via IL are often brittle and can fail when deployment conditions change from the training data [15].

Interactive Imitation Learning. Interactive IL allows demonstrators to provide corrective supervision in situations where policies require assistance. Some approaches require an expert to relabel states encountered by the agent with actions that the expert would have taken [15, 41, 42, 43, 44, 45, 46, 47], but it can be difficult for human supervisors to relabel robot actions in hindsight [48]. An alternative is to cede control of the system to a human supervisor for short corrective trajectories (termed *interventions*) in states where the robot policy needs assistance. Interventional data collection can either be human-gated [16, 36, 17, 37, 49, 50, 51, 52, 53, 54], where the human monitors the policy and decides when to provide interventions, or robot-gated [55, 56, 18, 57], where the robot decides when the human should provide interventions. However, these approaches impose the burden of collecting a sufficient number of human interventions for the robot to learn robust recovery. In this work, we develop a novel data generation mechanism based on replay-based imitation [19, 58, 59] in order to alleviate this burden.

Policy Adaptation under Domain Shift. There are other approaches besides interactive IL for increasing policy robustness. These include injecting noise during demonstration collection [60], having human operators intentionally introduce mistakes and corrections during data collection [61], and enabling policies to deal with partial observability [62, 63]. Other approaches include employing a planner to return to states that the agent has seen before [34, 64], using Reinforcement Learning (RL) with learned rewards to help an agent adapt to new object distributions [65], and using counterfactual data augmentation to identify irrelevant concepts and ensure agent behavior will not be affected by them [66]. There are also approaches to make policies trained with RL more robust, such as domain randomization [67, 68], using adversarial perturbations [69], and training agents to recover from unsafe situations [70]. One interpretation of I-MG is that it is a procedure analogous to domain randomization for sim-to-real transfer of policies trained with IL rather than RL.

4.2 Preliminaries

Problem Statement. We model the task environment as a Partially Observable Markov Decision Process (POMDP) with state space S, observation space O, and action space A. The robot does not have access to the transition dynamics or reward function but has a dataset of samples $D = \{(o,a)\}_{i=1}^{N}$ from an expert human policy $\pi_H : O \rightarrow A$. We assume that while the human observes observation o, the robot's observation is corrupted by some function z, yielding $z(o) = o' \in O$ (e.g., due to sensor noise or network delay). In this work we train policies on demonstration datasets D using supervised learning with the objective $\arg\min_{\theta} \mathbb{E}_{(o,a)\sim D}[-\log \pi_{\theta}(a|o)]$.

Algorithm 1 Interventional MimicGen

```
Declare: Initial state distribution $p_0$
Declare: Base dataset $D$
Declare: Number of iterations $k$
Declare: Number of human intervention episodes $m$
Declare: Number of synthesized trajectories $n$
procedure I-MG($p_0, D; k, m, n$)
    for $i \in [1, ..., k]$ do                                  ▷ One or more iterations
        $\pi_\theta \leftarrow$ TRAIN-POLICY($D$)
        $\mathscr{D} = \emptyset$
        for $j \in [1, ..., m]$ do                              ▷ Data Collection
            $s_0 \sim p_0$                                      ▷ Sample initial state
            $\tau \leftarrow$ EXECUTE-POLICY($s_0, \pi_\theta$)
            INTERVENE($\tau$)                                   ▷ Human intervention
            $\mathscr{D} \leftarrow \mathscr{D} \cup \tau$
        for $j \in [1, ..., n]$ do                              ▷ Data Generation
            $s_0 \sim p_0$
            $\xi \leftarrow$ EXECUTE-POLICY($s_0, \pi_\theta$)
            $t \leftarrow$ TERMINATE-POLICY($\xi$)
            $\tau \sim \mathscr{D}$                             ▷ Sample source demonstration
            $\tau \leftarrow \tau$[human]                        ▷ Filter intervention
            $\tau' \leftarrow$ ADAPT($\xi, \tau$)                ▷ Transform trajectory
            $\xi \leftarrow \xi \oplus$ REPLAY($\tau'$)
            if SATISFIES-GOAL($\xi[-1]$) then
                $D \leftarrow D \cup \xi[t:]$                    ▷ Filter intervention
    return $D$
```

Assumptions. Since we build on MimicGen [19], we inherit its assumptions: **(Assumption 1)** the action space consists of delta-pose commands in Cartesian end effector space; **(Assumption 2)** the task is a known sequence of object-centric subtasks; **(Assumption 3)** object poses can be observed at the beginning of each subtask during data collection (but not deployment). **(Assumption 4)** We also assume that demonstrated recovery behavior can be explained by some component of the robot's observations $\{o'_1, o'_2, \dots\}$ during a human intervention despite corruption by z. Without this assumption, it would not be possible for the robot to learn a policy that maps o' to $\pi_H(o)$. This information can be provided, for instance, in additional observation modalities such as force-torque sensing or tactile sensing that provide a coarse signal about an object's pose. Some settings may not require any additional information: for example, a fully closed gripper can inform the robot it must recover from a missed grasp.

MimicGen Data Generation System. MimicGen [19] takes a small set of source human demonstrations D_{src} and uses it to automatically generate a large dataset D in a target environment. It first divides each source trajectory $\tau \in D_{src}$ into object-centric manipulation segments $\{\tau_i\}_{i=1}^{M}$, each of which corresponds to an object-centric subtask (Assumption 2 above). Each segment is a sequence of end effector poses. Then, to generate a demonstration in a new scene, it uses the pose of the object corresponding to the current subtask, and transforms the poses in a source human segment τ_i (with an SE(3) transform) such that the relative poses between the end effector and the object frame are preserved between the source demonstration and the new scene. It also adds an interpolation segment between the robot's current configuration and the start of the transformed segment. Then, the sequence of poses in the interpolation segment and transformed segment are executed by the robot end effector controller open-loop until the current subtask is complete, at which point the process repeats for the next subtask.

4.3 Algorithm Pseudocode

See Algorithm 1 for the full pseudocode.

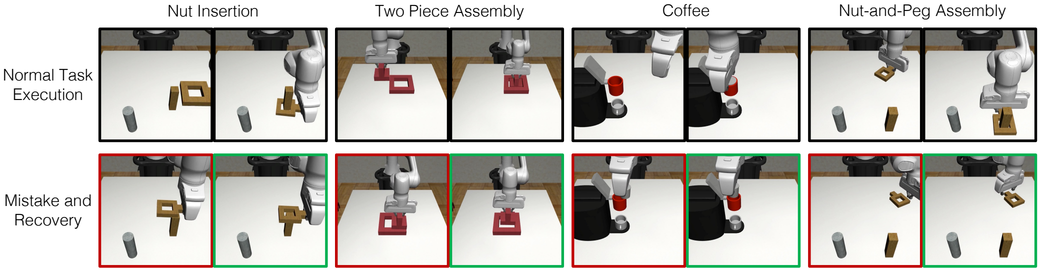

Figure 3: **Tasks.** We evaluate I-MG in several contact-rich, high-precision tasks. The top row shows normal task execution for each task while the bottom row shows typical mistakes encountered by the agent when using inaccurate object poses (or object geometry for Nut-and-Peg Assembly) and associated recovery behaviors.

4.4 Experiment Setup

4.4.1 Tasks

We consider tasks in the robosuite simulation environment [71] powered by MuJoCo [20] (see Fig. 3). We implement and evaluate the following 6-DOF continuous control manipulation tasks:

Nut Insertion: The robot must place a square nut (held in-hand) onto a square peg. The peg position is sampled in a 10 cm x 10 cm region at the start of each episode.

2-Piece Assembly: The robot must place an object into a square receptacle with a narrow affordance region. The receptacle position is sampled in a 10 cm x 10 cm region at the start of each episode.

Coffee: The robot must place and release a coffee pod into a coffee machine pod holder with a narrow affordance region. The coffee machine position is sampled in a 10 cm x 10 cm region at the start of each episode.

Block Grasp: The robot must reach a block and grasp it. The block position is sampled in a 20 cm x 30 cm region at the start of each episode and the gripper orientation is fixed to top-down.

Nut-and-Peg Assembly: [71, 40] A multi-stage task consisting of (1) grasping a nut with a varying initial position and orientation and (2) placing it on a peg in a fixed target location. The nut is placed in a 0.5 cm x 11.5 cm region with a random top-down rotation at the start of each episode.

Sources of Observation Error. In the first three environments, the source of observation error is *sensor noise*: at test time, uniform random noise is applied to the observed position of the peg (±4 cm in each dimension, with at least 2 cm in one dimension), receptacle (±4 cm in each dimension, with at least 1 cm in one dimension), coffee machine (radial noise between 2 cm and 4 cm), and block (±1 cm in x and ±7 cm in y, with at least 2.5 cm in y) respectively. In the fourth environment, the source of observation error is *object geometry*: for an identical observed nut pose, the nut handle may exist on either of two sides of the nut. This setting corresponds to object model misspecification during pose registration.

4.4.2 Setup

Data Collection. For interventional data collection, we use the remote teleoperation system proposed by Mandlekar et al. [17]. The observation space consists of robot proprioception (6DOF end effector pose and gripper finger width) and object poses, while the action space consists of 6DOF pose deltas and a binary gripper open/close command. For the base policy π_θ used in each task, we (1) collect 10 full human task demonstrations in each environment *without* observation corruption (i.e., ground truth poses), (2) synthesize 1000 demonstrations with MimicGen [19], and (3) train an off-the-shelf BC-RNN policy with default hyperparameters using the robomimic framework [40], with the exception of an increased learning rate of 0.001 [19].

Data Generation. We then deploy π_θ in the test environment *with* observation corruption (i.e., object pose error) and collect 10 human-gated interventions. These interventions are expanded to 1000 synthetic interventions with I-MG and aggregated with the 1000 demonstrations used to train the base policy. Finally, we train a new BC-RNN policy on the aggregated dataset. We report policy performance as the success rate over 50 trials for the highest performing checkpoint during training (where training takes 2000 epochs with evaluation every 50 epochs), as in [40, 19].

Observability. In order for demonstrated recovery behavior to be learnable (Section 4.2), I-MG and all baselines can access additional observation information in Nut Insertion, Two-Piece Assembly, and Coffee upon contact between (1) the nut and peg, (2) object and receptacle, and (3) pod and pod holder, respectively. We study both the idealized case of full observability (i.e., ground truth pose) upon contact in Section 3 and partially improved observability (e.g., position of contact) in Appendix 4.5. These are intended to be surrogates for sensor modalities such as force-torque sensing that can help inform the robot about the object pose when its belief is wrong. For Nut-and-Peg Assembly, we do not add additional information, as a closed gripper state is sufficient for the policy to map a missed grasp to learned recovery.

Real Robot Setup. To evaluate sim-to-real transfer, we set up a real-world counterpart to the Block Grasp task. We use a Franka Research 3 robot arm and a red cube with a side length of 5 cm. We use an Intel RealSense D415 depth camera and Iterative Closest Point [72] for cube pose estimation. The deployed policies output continuous control delta-pose actions at 20 Hz and are trained entirely in simulation without any real-world data or fine-tuning. See Figure 4 for images.

4.4.3 Baselines

We implement and evaluate the following baselines. Each baseline corresponds to a *different dataset* used to train the agent (all agents are trained with BC-RNN [40]):

Base: Deploy the base policy in the test environment without any additional data or fine-tuning.

Source Interventions (Source Int): Deploy the base policy π_θ, collect 10 human interventions when the policy makes mistakes, and add them to the base dataset.

Weighted Source Interventions (Weighted Src Int) [17]: Same as Source Interventions, but weight the intervention data higher so that it is sampled as frequently as the base data despite its smaller quantity.

Source Demonstrations (Source Demo): Collect 10 full human task demonstrations in the test environment.

MimicGen Demonstrations (MG Demo) [19]: Same as Source Demonstrations, but use (regular) MimicGen to generate 1000 synthetic demonstrations from the initial 10.

Policy Execution Ablation (I-MG - Policy): Augment the 10 source interventions to 1000 I-MG interventions, but do not use policy execution to generate new mistake states.

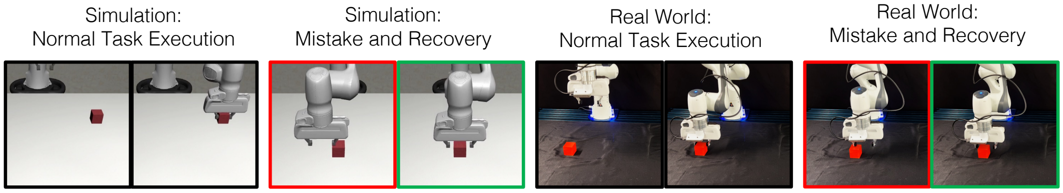

Figure 4: **Sim-to-Real.** We evaluate sim-to-real transfer for a block grasping task with a Franka Panda robot. Similar to Figure 3 we show normal task execution, typical mistakes due to inaccurate object poses, and associated recovery for the simulation and real world environments.

Dataset	Geometry 1	Geometry 2	Mixture
Base	**100%**	0%	50%
Source Int	**100%**	6%	53%
MG Demo [19]	0%	**100%**	50%
Base + MG Demo	64%	60%	62%
I-MG	92%	88%	**90%**

Table 2: Results in the Nut-and-Peg Assembly experiment. While baselines typically overfit to one geometry or struggle with disambiguating the two, I-MG attains high performance on the mixture of geometries.

Dataset	Nut Insertion	2-Pc Assembly
Base	26%	6%
Source Int	40%	6%
MG Demo [19]	46%	22%
I-MG - Policy	68%	42%
I-MG	**90%**	**66%**

Table 3: Additional evaluation in two domains with partially improved (rather than full) observability upon contact.

Dataset	Simulation	Real
Base	6%	0%
Source Int	26%	10%
MG Demo [19]	42%	50%
I-MG - Policy	86%	60%
I-MG	**100%**	**90%**

Table 4: Results for the block grasping task in simulation (50 trials) and zero-shot evaluation of the same policies in the real world (10 trials).

4.5 Additional Experiments and Analysis

We present additional results and further analysis on the properties of I-MG in this section.

I-MG is useful across different sources of observation error. Results for the Nut-and-Peg Assembly task with object geometry error (Section 4.4.1) are in Table 2. We evaluate each policy with 50 evaluations of each of the two possible geometries. Base and Source Int attain perfect performance on the original geometry but struggle with the alternate geometry (0%-6% performance). MG Demo has the opposite issue: since it consists of test-time demonstrations with the alternate geometry, it can attain perfect performance on the alternate but 0% on the original. A mixture of full demonstrations on both geometries (Base + MG Demo) attains an even 60% and 64%; since it does not observe recovery behavior it must guess between the two object geometries and has difficulty performing much higher than the 50% expected value of random chance. Finally, I-MG maintains 92% performance on the original geometry but also learns to recover when missing its grasp due to the alternate geometry (88%), leading to a 28%-40% improvement in the average case over baselines. See the website for videos.

I-MG can facilitate sim-to-real transfer of learned control policies. In Table 4 we observe that state-based policies for the Block Grasp task deployed zero-shot on the physical system perform similarly to simulation. I-MG outperforms baselines by 14%-94% in simulation and 30%-90% in real world trials, suggesting learned recovery behaviors can transfer to real. The policy is also robust to physical perturbations, dynamic pose changes, and visual distractors; see the website for videos.

How is agent performance affected as observability decreases? For Nut Insertion, we replace true pose information upon contact with the mean position of the first contact between the nut and peg; for 2-Piece Assembly, we provide the unit vector in the direction of the true pose at the first point of contact. Table 3 in comparison with Table 1 shows that, as expected, a degradation in observability results in a degradation in agent performance. However, I-MG performance falls by only 4%-8%, indicating partial observability can be sufficient to ground recovery behavior. An important direction for future work is investigating raw real-world perception signals such as force-torque sensing.

How does action noise play a role in data generation? Noise injection in executed actions during the MimicGen process can significantly increase downstream policy performance [19, 60]; consequently, we used the default setting of additive unit Gaussian noise with 0.05 scale [19]. However, we found that I-MG can be less sensitive to the inclusion of action noise: in the Nut Assembly environment, with a 10× reduced magnitude of action noise, the ablation's performance falls from 86% to 66%, while I-MG performance remains at 98%. This could be due to the broad coverage of the mistake distribution derived through policy execution (Section 2.2).

How does performance vary across training seeds? I-MG in the (full observability) Nut Assembly task attains 98%, 100%, and 98% for 3 training seeds, indicating stability across runs. More evidence is available on the supplemental website.

How does synthetic Interventional MimicGen data compare to an equal amount of human data? In 2-Piece Assembly, 100 I-MG interventions (from 10 human interventions) attain 24% while 100 human interventions attain 46%. Both improve upon 10 human interventions, which only attains 6% (Table 1). However, 1000 I-MG interventions from 10 human interventions (70%) can outperform 100 human interventions, and 100 human interventions take significantly more human time and effort to collect than 10 human interventions (29.9 minutes instead of 3.6 minutes).

How does performance scale with the amount of synthetically generated interventions? With the same 10 human source interventions in 2-Piece Assembly, an agent trained on 200 synthetic I-MG interventions attains 34%, 1000 interventions attains 70% (Table 1), and 5000 interventions attains 88%. This suggests performance scales with dataset size, at the cost of additional data generation time.