BLIND PRECISE ASSEMBLY: PROPRIOCEPTIVE SIM-TO-REAL PEG-IN-HOLE FOR DIVERSE PARTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Every robot already carries high-resolution contact sensors: its own joints. Legged robots and humanoids exploit them to walk, run, and track whole-body motions without vision, but there the contacts are large and forceful. Is joint sensing also enough enough for precise, contact-rich assembly, where the robot must find a millimeter-scale opening through light contact? We study this question by training a blind insertion policy that observes only a short history of joint positions, velocities, and its own actions, without visual, force, torque, or tactile sensing. The poses of both the hole and the peg are unknown to the policy, and the grasp is randomized substantially. The policy must therefore find the opening through contact, sensed exclusively through its own proprioception. We train this behavior with reinforcement learning entirely in simulation over randomized parts, grasps, and contacts, and it transfers zero-shot to a real robot and inserts pegs of different shapes and sizes in 74% of trials. Controlled comparisons with matched visual policies in simulation show that although blind policies are slower, they generalize far better to unseen part shapes and sizes. Training such policies needs thousands of different assemblies, each simulated with its millimeter clearance intact, so we also release an open-source generator that builds them procedurally from exact convex pieces for MuJoCo; it is more accurate and faster than convex decomposition or MuJoCo's signed-distance collision.

1 INTRODUCTION

A force applied anywhere on a robot's body travels through its links to the joints, which then move differently from what their commands alone would produce. The encoders record that difference, so the joints of every arm and leg double as contact sensors. Legged robots exploit this proprioceptive signal to walk, run, and climb stairs blind (Lee et al., 2020; Siekmann et al., 2021a;b; Kumar et al., 2021), and humanoids to walk and track whole-body motions (Radosavovic et al., 2024; Cheng et al., 2024; He et al., 2024). In these settings, however, contacts are large and forceful, and joints often comply meaningfully while recording measurable deviations. Can robots exploit the same signal for precise, contact-rich assembly?

Performing tight insertion with a robotic arm occupies the opposite regime: the robot must find an opening with a millimeter or less of clearance through light contact whose effect on the joints is easy to miss. We study this question by training a *blind* insertion policy that has no information outside of its own joint sensing, and verify that proprioception contains enough information for contact-rich hole localization and precise insertion. In the real world, our policy inserts five pegs, stadiums of four sizes and proportions and a circle it never saw in training, with 0.5–1.25 mm of clearance per side, in 74% of 151 trials, without being given the hole's location, the peg's pose in its grasp, the part's size, or its clearance (figure 1, section 3.3). To our knowledge, this is the first such demonstration.

We train our policy entirely in simulation with reinforcement learning and deploy it in the real world without fine-tuning. Training randomizes part geometry, contact properties, fixture poses, and grasps, which can hold the peg anywhere across the finger pads, tilted by up to 20°. The policy observes a 0.6 s history of joint positions and velocities, the gripper pose and twist computed from them, and its own previous actions. It receives no image and no force signal, so it can perceive contact and localize the hole only through its own motion (section 2).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 1: **Blind insertion.** A policy trained only in simulation inserts pegs of different sizes in the real world using only its joint encoders and previous actions.

Controlled comparisons betw een our proprioceptive policy and matched visual policies revealed two interesting properties of visually blind behaviors:

- **Visually blind yet intelligent.** The learned behavior is not an open-loop pattern: its learned representation encoding its own proprioceptive inputs contains meaningful task-relevant information that the policy itself heavily relies on to adjust its behavior (section 3.1).
- **Proprioception generalizes better.** Because the blind policy depends on the part only through its contacts, it degrades far less than vision on peg shapes and sizes it has not seen during training (section 3.2).

Training for this robustness requires diverse parts: thousands of different assemblies, generated procedurally and simulated in parallel, each with its millimeter clearance intact. In MuJoCo (Todorov et al., 2012), building holes from convex primitives by hand does not scale to random shapes, and the automatic alternatives, convex decomposition and signed-distance-field (SDF) collision, are less accurate and slower at these clearances. We therefore built `mjfit`, which procedurally generates pegs and holes from any planar outline, as well as threads and gears, out of exact convex prisms. To summarize, our contributions are:

1. the first learned policy for precise insertion from proprioception alone, given no hole pose, grasp, or part model: a blind policy, trained purely in simulation, that transfers zero-shot to the real world across part sizes and to a cross-section it never saw in training (section 3.3);
2. a controlled comparison showing that a blind policy searches reactively, acting on the contact state it infers from proprioception, rather than by a fixed pattern and trades speed for robustness to unseen geometry (section 3);
3. `mjfit`, an open-source procedural generator of assemblies for MuJoCo (pegs and holes, threads, and gears) with exact convex collision geometry, which supplies the thousands of different training assemblies, and a benchmark against analytic ground truth in which it is more accurate and faster than convex decomposition and MuJoCo's SDF collision (section 2.2).

2 Learning Blind Insertion

We train a blind insertion policy using auxiliary contact supervision to make learning tractable without providing any privileged information to the actor (section 2.3). This training relies on `mjfit`, our open-source procedural generator for MuJoCo, to supply thousands of different assemblies with exact convex collision geometry that preserves millimeter-scale clearances (section 2.2).

2.1 Task

A 7-DOF robot arm equipped with a parallel jaw gripper must learn to insert pegs across a wide range of initial conditions (figure 2). The pegs are stadium prisms with varying size, aspect ratio, mass, and

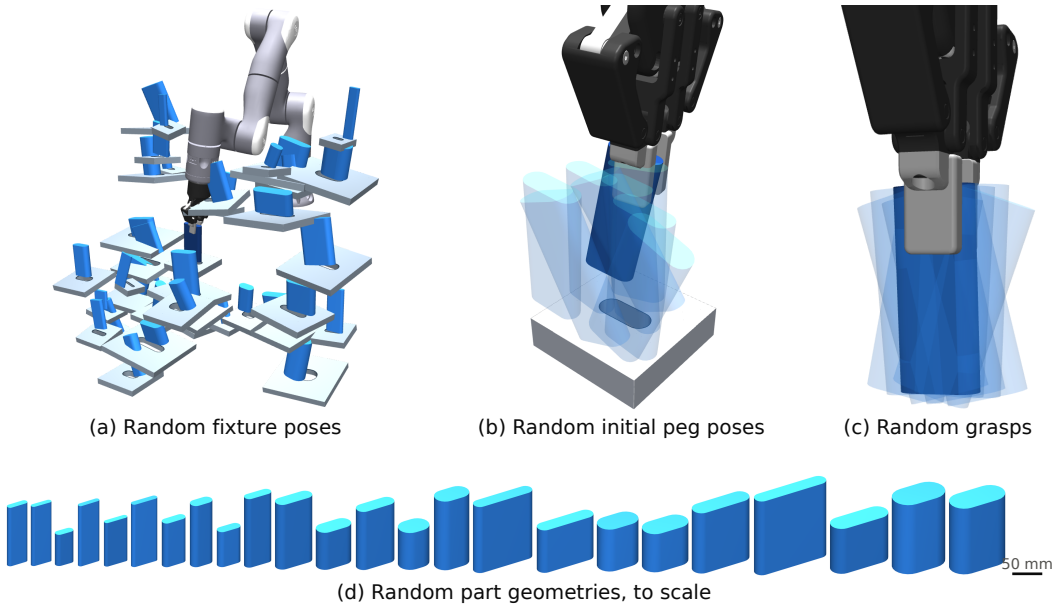

Figure 2: Training distribution of the deployed policy.

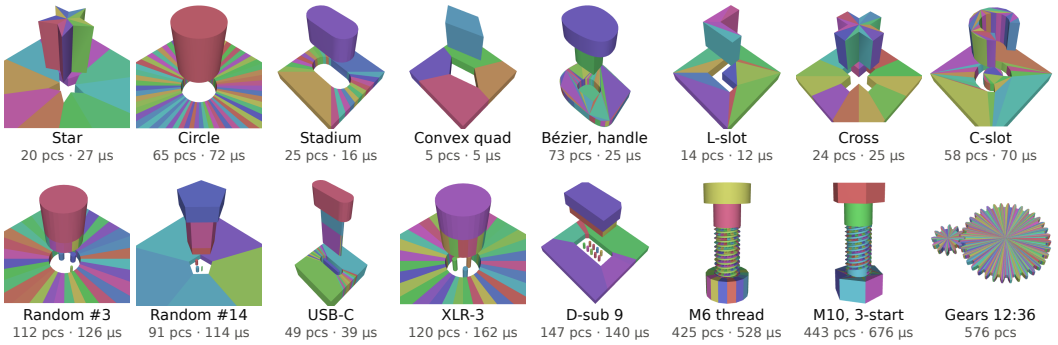

Figure 3: **Assets generated by `mjfit`**, loaded and simulated as emitted, and rendered from their collision geometry, one color per convex piece. Labels give the number of convex pieces (pcs) and the CPU time per physics step with the parts in contact (one thread).

length, and the assembly clearance ranges from 0.5 to 1.5mm. Episodes are initialized with the peg in contact with the fixture surface, typically around 2cm from the hole and tilted far more than the clearance admits, and succeed when the peg is seated in the hole.

2.2 Simulating tight insertion at scale

Training on this distribution means simulating thousands of parallel worlds that each hold a different, randomly generated assembly, with per-side clearances down to 0.5 mm. MuJoCo collides meshes through their convex hulls, so a concave hole must be given an SDF or built from convex pieces, which simulation benchmarks assemble from boxes by hand (Zhu et al., 2020; Tao et al., 2024; Zakka et al., 2025).

Exact convex decomposition. `mjfit` automates the hand-built hole and generalizes it to arbitrary outlines. Given the planar outline of a hole, optionally with islands such as connector pins, it emits a ready-to-simulate MJCF model in which every contact surface is a MuJoCo primitive or a convex prism (figure 3). The plate around the hole is split into convex polygons and extruded; the peg is the hole outline offset inward by the clearance, so every peg edge sits exactly the clearance from its hole

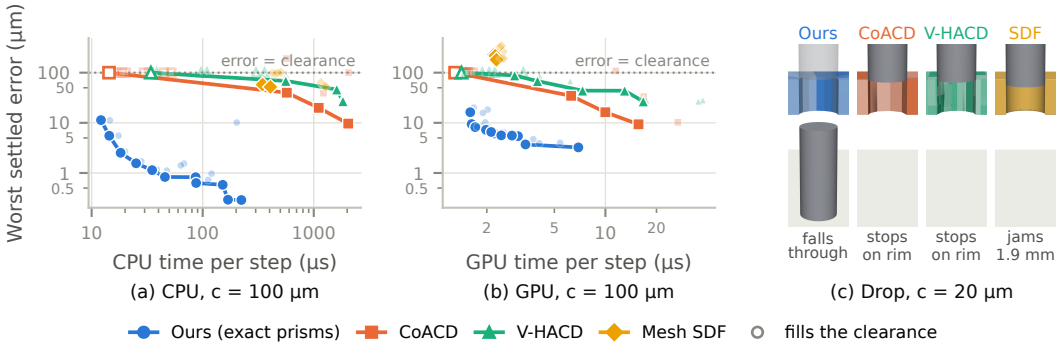

Figure 5: **Speed against accuracy** for a 16 mm round peg with 100 µm clearance, (a) on CPU, one environment on one thread, and (b) on GPU, per environment step across 256 MuJoCo Warp worlds, with the error also measured on Warp. Each line is one method's Pareto front over all its settings; lower left is better, and hollow markers fill the clearance (dotted line). (c) The peg, centered and aligned, dropped from 1 mm above each method's default plate and shown 0.2 s later.

edge, and islands become pockets in a hollow peg. Curves are tessellated by a *circumscribed* polygon, so the wall never intrudes into the true hole, as it does in rings of boxes that inscribe the circle (Zhu et al., 2020); the only error is extra play at the corners, $c\,(1/\cos(\pi/n) - 1)$ for a circle of n sides at radial clearance c. Because the construction knows which region must stay empty, the user specifies the clearance directly rather than tuning a concavity tolerance and checking the result. Companion generators apply the same idea to bolt–nut pairs and spur gears, and a sampler draws random outlines, islands, and dimensions, so one training run can hold thousands of different assemblies.

Benchmark. We compare against the three automatic ways to give MuJoCo a concave hole: CoACD (Wei et al., 2022), V-HACD (Mamou & Ghorbel, 2009), and MuJoCo's native mesh SDF, each swept over its granularity setting. The test is a 16 mm round or square hole with clearance down to 20 µm, where passive probes have exact answers; we measure each method's error against its time per step, on CPU and on GPU with MuJoCo Warp (Google DeepMind and NVIDIA, 2025) (figure 5; Appendix B). Approximate decomposition tolerates concavity far larger than an assembly's clearance, so it leaves wall material inside the hole (figure 4); refining it narrows the error but never closes it (figure 5a). Mesh SDF misses contacts and rounds the rim. These errors change outcomes: a centered peg stops on the rim or jams on every baseline's default plate, but falls cleanly through ours (figure 5c). `mjfit` is also an order of magnitude faster than mesh SDF on CPU at equal or better accuracy (Table 3), and has the best speed–accuracy trade-off on GPU (figure 5b). Each assembly is only a few dozen convex pieces, so every one of our 12,288 parallel training worlds (Zakka et al., 2026) can hold a different one.

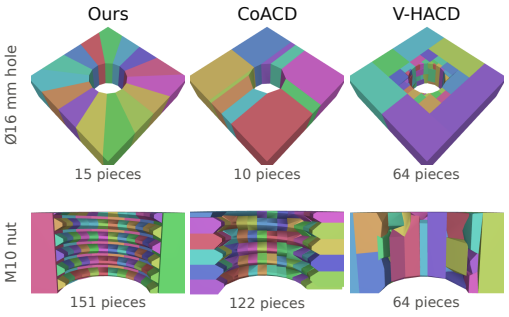

Figure 4: **Same parts, three decompositions**, one color per convex piece (Ø16 mm plate; M10 nut cut open). Ours errs only toward extra play; CoACD and V-HACD put material inside the hole and fill the nut's 0.1 mm thread clearance.

2.3 Policy and Training

Policy. At each step, the actor (figure 6) observes its joint positions and velocities, the gripper pose and twist computed from them by forward kinematics, and its previous action. It never receives the hole pose, the part geometry, a wrench estimate, or an image. A temporal convolutional encoder compresses the last 0.6 s of observations into a latent, and an MLP maps the latent and the current observation to joint-position increments. A joint impedance controller tracks them on an arm model whose armature, damping, and friction were identified from real-world measurements. Additional training details in Appendix A.

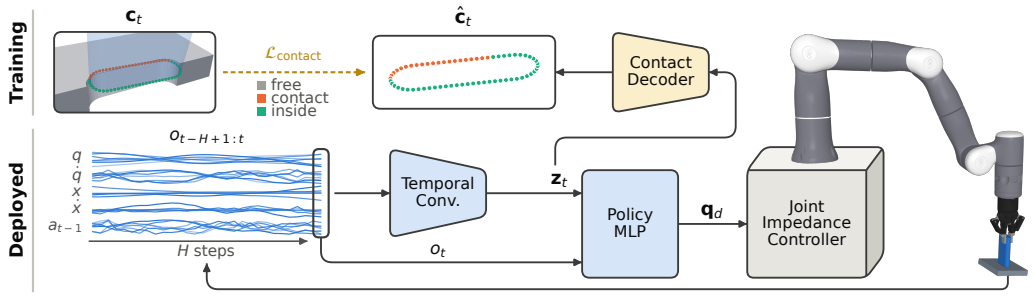

Figure 6: **Policy Architecture**. A temporal convolution compresses the last $H = 30$ proprioceptive observations o_t (joint positions q and velocities $\dot{q}$, gripper pose x and twist $\dot{x}$, and the previous action a_{t-1}) into a latent z_t, and an MLP maps z_t and the current observation o_t to joint targets q_d for a joint impedance controller. During training, we provide contact supervision from privileged simulator state. A contact decoder (amber) predicts the contact state ($\hat{c}_t$) from z_t. Its cross-entropy $\mathcal{L}_{\text{contact}}$ with the simulator's labels c_t (left, near half of the fixture cut away) trains the encoder, but $\hat{c}_t$ never reaches the policy.

Training. We use PPO (Schulman et al., 2017) with a sparse reward: one bonus when the peg is first captured by the opening and one on success. Two training-only signals make the sparse reward tractable without giving the actor privileged information: an asymmetric critic (Pinto et al., 2018) that also sees the peg's pose relative to the hole and the true contact labels, and the contact supervision described next. Randomizing dynamics and geometry (Tobin et al., 2017; Peng et al., 2018) covers the ranges of Table 1.

Contact supervision. A blind policy has to infer contact from how the arm moves. To give that inference a dense training signal, we supervise the actor's latent with the contact state it cannot observe. At every step, the simulator labels points along the peg's bottom edge as free, in contact with the rim or top face, or inside the opening (figure 6, top), geometrically from the peg and hole poses. A small contact decoder predicts these labels from the latent z_t, and its class-weighted cross-entropy is added to the PPO loss (Jaderberg et al., 2017). The gradient of this loss trains the encoder, but the prediction never enters the policy, so nothing privileged reaches the actor and the decoder is dropped at deployment. After training, the decoder recovers the labels from proprioception well above chance, most reliably for points inside the opening.

Visual policies. To isolate the sensing modality, we train visual policies with the same task, resets, randomization, reward, actions, and success check. A camera rigidly mounted on the wrist renders a 64×64 RGB image of the peg and opening. The visual policy sees only images, over the same 0.6 s history, through a compact CNN; its critic also sees only images.

3 What does blindness cost, and what does it buy?

We compare the blind policy with a visual policy trained on the same task using wrist-camera observations, with scripted spiral search serving as a geometric baseline. We first evaluate in-distribution success rates, insertion times, and lateral travel, and analyze the blind policy's history representation and actions to assess whether its search responds to contact (section 3.1). We then compare both learned policies on part geometries outside the training distribution (section 3.2) and evaluate zero-shot deployment of the blind policy on a real robot without fine-tuning (section 3.3).

3.1 Blind but smart: between scripted search and vision

Proprioception is sufficient, but slower. In distribution, both policies achieve near-perfect insertion success rates (figure 9, first column), but the blind policy takes more time to succeed (figure 7). Across matched successful starts, the blind policy's median lateral travel exceeds that of the visual policy but is substantially shorter than that of an optimistic scripted spiral baseline (c), indicating that proprioceptive feedback supports more efficient search than a fixed geometric pattern.

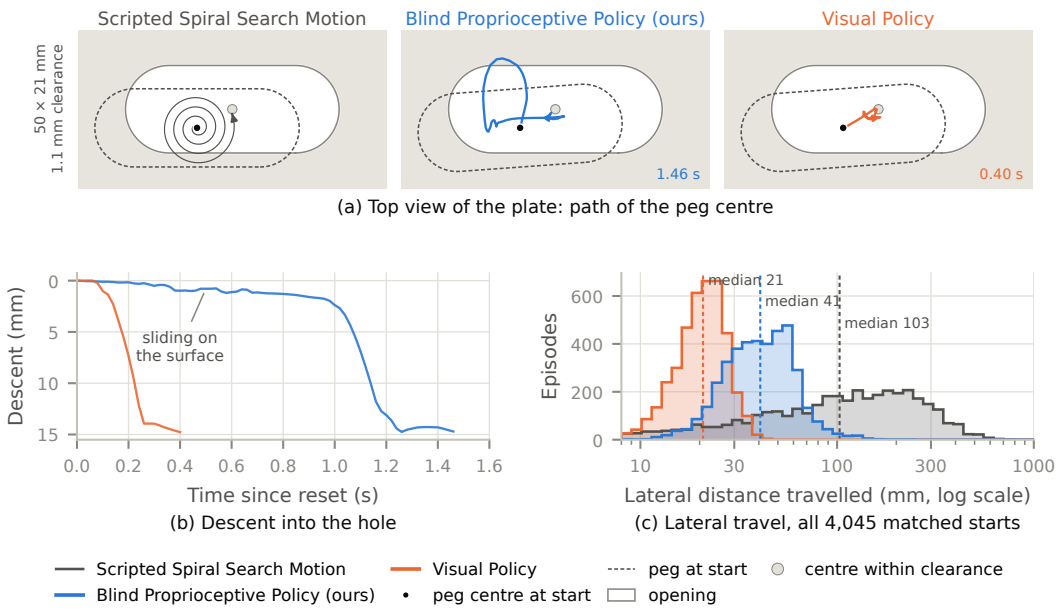

Figure 7: **Proprioception guides efficient search without vision.** Because the blind policy must search through contact, it follows a less direct path and seats later than the visual poilcy (a, b). Nonetheless, it travels far less (c) than an optimistic scripted spiral search baseline (Chhatpar & Branicky, 2001).

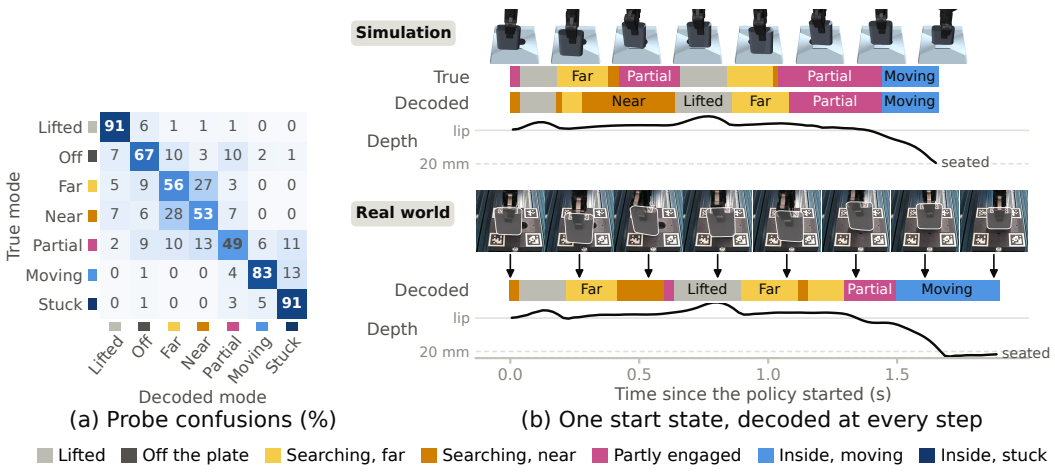

Figure 8: **The policy's joint history encodes the insertion mode, in simulation and in the real world.** A linear probe fitted only in simulation reads the mode from the policy's history latent. (a) It cleanly separates modes that call for different actions (lifted, moving, stuck), but not near from far search. (b) On the robot, it decodes the same sequence of modes as in simulation. Camera frames are for illustration only; the policy never sees them.

Visually blind but intelligent. Is the blind policy's search a fixed pattern that happens to work, or does it respond to what its joints reveal about contact? Everything the actor knows about the past passes through its history latent z_t (figure 6). A policy that reacts to contact must track there where it is in the task, such as whether the peg is still searching or has caught the opening, and act on it; an open-loop pattern needs neither. We label every simulated state with one of seven insertion modes from the privileged peg pose: lifted, beside the plate, searching far from the opening or near it, partly engaged, and fully inside, either moving or stuck, and ask what a linear probe can read from z_t (figure 8a; Appendix C). The latent, far more than the current observation alone, keeps what

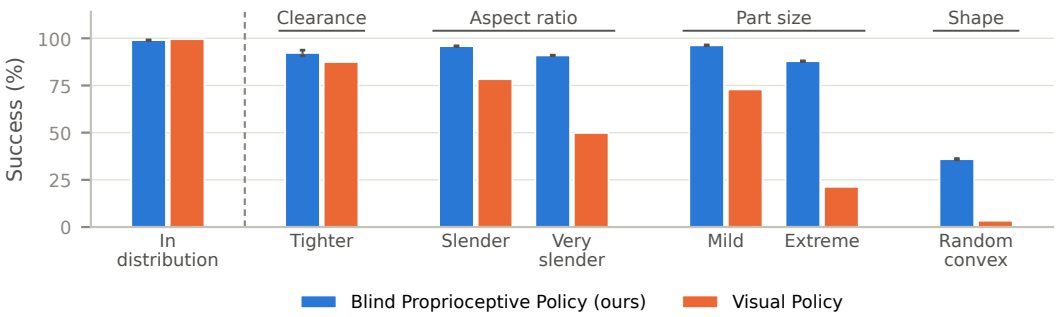

Figure 9: **Proprioceptive policies remain more robust to geometric variation.** Success in distribution (left of the dashed line) and on unseen parts, each widening one object property beyond its training range while the others stay within theirs. The proprioceptive policy retains its performance better across all variations.

touch can reveal and the task needs: the contact events that should change the policy's action, not the distance to a hole it cannot yet feel. Its actions follow the same split (figure 12). The policy acts on whether it has found the hole, not on where the hole is; its search responds to contact rather than following a script.

3.2 Generalization to unseen parts

As figure 9 shows, the two policies are equally reliable on the parts they were trained on, but on parts they have not seen, the visual policy is the less reliable of the two (exact values in Appendix C). A tighter fit barely affects either policy, since clearance matters only in the final alignment. But as parts grow more elongated or leave the training sizes, the blind policy degrades gently while the visual policy falls steeply, and on a new cross-section the blind policy still inserts about a third of the pegs while the visual policy almost never does. The reason is what each policy senses. A camera sees a part's size and shape directly, so the visual policy learns what the training parts look like, and that does not extrapolate. Contact changes far less: the rim of a larger or narrower hole resists a push just as a training hole's rim does, so what the blind policy learned through its joints carries over.

3.3 Zero-shot transfer to the real world

Setup. We deploy the frozen blind policy on a Flexiv Rizon 4S with a Robotiq 2F-85 at 50 Hz, with no real-world data or fine-tuning. It receives only encoder-derived joint and gripper state; no camera, force/torque sensor, or hole pose enters the policy. We test 3D-printed pegs (figure 10): stadiums of four sizes and proportions, whose cross-section, size, and clearance lie inside the training distribution, and a circular peg, a cross-section the policy never saw in training. Each trial starts from a start state drawn from the simulated reset distribution, with the peg grasped at a randomized position and tilt and touching the fixture near the hole. An operator marks a trial as successful once the peg is seated.

Results. One policy, with no parameter changes, inserts every peg in most trials (figure 10), pressing onto the surface, sweeping, and dropping in once the opening is under the peg, as in simulation (figure 1). Success barely changes with stadium size and drops only modestly for the circle, a cross-section unseen in training. The hardest peg is the most elongated and tightest stadium, at the edge of both training ranges. The circle succeeds more often in the real world than in simulation; each simulated rate averages over many randomized worlds, and the real world is one particular world in which the policy may do better than on average.

4 Related work

Robotic insertion. Classical peg-in-hole assembly combines compliance with search patterns such as spirals, rasters, and tilting (Whitney, 1982; Chhatpar & Branicky, 2001; Marvel et al., 2018), in some cases without force feedback (Park et al., 2017); these strategies assume a known grasp and a hole within the pattern and are tuned for each part. Learned insertion policies usually add force/torque

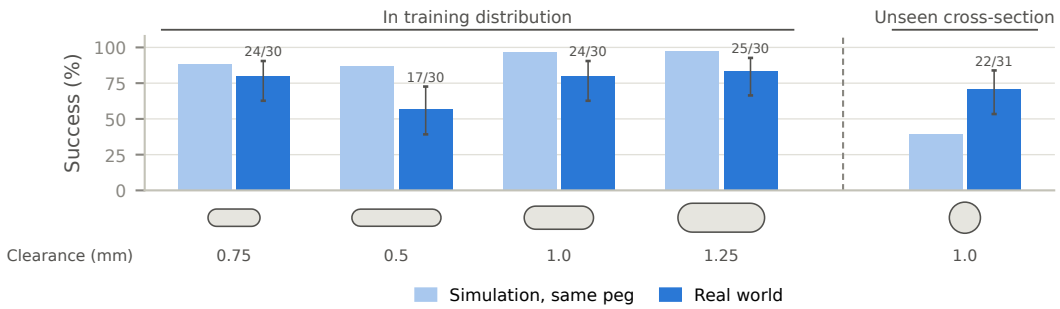

Figure 10: **Real-world results** of the blind policy, with no fine-tuning, beside the same policy in simulation on the same peg. Below each pair are the peg's cross-section, drawn to scale, and its clearance per side. Error bars are 95% Wilson intervals. Details in Appendix C.

sensing (Inoue et al., 2017; Beltran-Hernandez et al., 2020), wrist vision (Schoettler et al., 2020; Bogunowicz et al., 2020; Spector & Di Castro, 2021; Haugaard et al., 2021; Seo et al., 2025), or both (Lee et al., 2019; Luo et al., 2021; Watson et al., 2020). Wrist force/torque sensing combined with a soft wrist has allowed policies to tolerate 10 mm of hole-position error and an unseen peg shape (Fuchioka et al., 2024), while a blind dexterous policy relying on fingertip tactile sensing (Pan et al.) was demonstrated for precise insertion. Others (Tang et al., 2023; 2024; Noseworthy et al., 2025; Narang et al., 2022) observe joint state and a direct goal pose from perception or calibration with a few millimeters of error. Our policy instead uses proprioception alone: it is given no hole pose, part model, or force signal, and must locate the hole through contact while the peg's pose in its grasp is unknown.

Proprioceptive policies. Legged robots traverse rough terrain with policies built on histories of onboard state (Hwangbo et al., 2019; Lee et al., 2020; Siekmann et al., 2021b; Kumar et al., 2021; Radosavovic et al., 2024). In manipulation, proprioceptive policies reorient unseen objects in hand (Chen et al., 2021; Qi et al., 2022), and *Blind Dexterity* treats joint and action histories as a whole-body tactile channel (Bhatt et al., 2026). These tasks involve forceful contact or centimeter tolerances; insertion needs neither. In imitation learning, robot state can hurt (Team et al., 2024) by suppressing visual learning (Lu et al., 2026).

Simulating tight contact. Approximate convex decomposition (Mamou & Ghorbel, 2009; Wei et al., 2022) tolerates concavity relative to the mesh size, which can exceed an assembly's clearance. Exact polygon partitions are classical (Keil, 1985), and simulation benchmarks build holes from primitives by hand (Zhu et al., 2020; Tao et al., 2024; Zakka et al., 2025); `mjfit` automates this for arbitrary outlines and generates assemblies in bulk. SDFs handle concavity directly and underlie GPU assembly simulation in Factory and its successors (Narang et al., 2022; Macklin et al., 2020; Tang et al., 2023), whose assets obtain their clearance approximately, by displacing mesh vertices or eroding voxels (Tang et al., 2024; Wang et al., 2025); SDF accuracy depends on field resolution and contact search. Newton combines SDFs with hydroelastic contact (The Newton developers, 2025), and barrier methods simulate threads and gear trains without interpenetration (Ferguson et al., 2021; Lan et al., 2022). Our benchmark (section 2.2) follows the simulator comparisons of Erez et al. (2015) and Le Lidec et al. (2024). CRISP (Lee et al., 2026) recently compared engines on tight insertion and threading; we instead keep MuJoCo's solver fixed and vary only the collision geometry, using passive probes with analytic answers and speed–accuracy Pareto fronts.

5 Limitations and Conclusion

We trained a policy that inserts pegs from proprioception alone, with no hole pose, grasp, or part model, on thousands of assemblies that `mjfit` generates with exact collision geometry. It transfers zero-shot to the real world and, in simulation, generalizes to unseen parts far better than a matched visual policy. It is, however, a last-mile manipulation skill, and its limits are those of one. First, it assumes it starts in contact: every training episode begins with the peg already touching the fixture a

few centimeters from the hole, so another motion, scripted or visual, must bring it there. Second, its behavior is local. It searches the surface around where it starts and corrects the misalignments it met in training, but it cannot handle events that call for more than a local correction, such as a peg that slides out from the grasp and lands far from the hole.

Reproducibility statement

We will release `mjfit`, the collision benchmark with its raw results, as well as the policy training and evaluation code.

AI use statement

Generative AI assisted with prose editing, figure scripting, and literature discovery. The authors verified the citations, technical descriptions, and empirical claims and are responsible for the final paper.

References

Cristian C Beltran-Hernandez, Damien Petit, Ixchel G Ramirez-Alpizar, and Kensuke Harada. Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach. *Applied Sciences*, 10(19):6923, 2020.

Aditya Bhatt, Oleg Kaidanov, Puze Liu, and Jan Peters. Blind dexterity: Whole-body humanoid manipulation via pure proprioception, 2026.

Damian Bogunowicz, Aleksandr Rybnikov, Komal Vendidandi, and Fedor Chervinskii. Sim2real for peg-hole insertion with eye-in-hand camera, 2020. URL `https://arxiv.org/abs/2005.14401`.

Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *Conference on Robot Learning*, 2021.

Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. 2024.

Siddharth R Chhatpar and Michael S Branicky. Search strategies for peg-in-hole assemblies with position uncertainty. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, volume 3, pp. 1465–1470. IEEE, 2001.

Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX. In *IEEE International Conference on Robotics and Automation*, pp. 4397–4404, 2015. doi: 10.1109/ICRA.2015.7139807.

Zachary Ferguson, Minchen Li, Teseo Schneider, Francisca Gil-Ureta, Timothy Langlois, Chenfanfu Jiang, Denis Zorin, Danny M. Kaufman, and Daniele Panozzo. Intersection-free rigid body dynamics. *ACM Transactions on Graphics*, 40(4), 2021. doi: 10.1145/3450626.3459802.

Yuni Fuchioka, Cristian C. Beltran-Hernandez, Hai Nguyen, and Masashi Hamaya. Robotic object insertion with a soft wrist through sim-to-real privileged training. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 9159–9166, 2024. doi: 10.1109/IROS58592.2024.10801575.

Google DeepMind and NVIDIA. MuJoCo Warp: GPU-accelerated MuJoCo. `https://github.com/google-deepmind/mujoco_warp`, 2025.

Rasmus Haugaard, Jeppe Langaa, Christoffer Sloth, and Anders Buch. Fast robust peg-in-hole insertion with continuous visual servoing. In *Conference on Robot Learning*, pp. 1696–1705. PMLR, 2021.

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. 2024.

Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science robotics*, 4(26):eaau5872, 2019.

Tadanobu Inoue, Giovanni De Magistris, Asim Munawar, Tsuyoshi Yokoya, and Ryuki Tachibana. Deep reinforcement learning for high precision assembly tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 819–825, 2017. doi: 10.1109/IROS.2017.8202244.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2017.

J. Mark Keil. Decomposing a polygon into simpler components. *SIAM Journal on Computing*, 14(4): 799–817, 1985. doi: 10.1137/0214056.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. 2021.

Lei Lan, Danny M. Kaufman, Minchen Li, Chenfanfu Jiang, and Yin Yang. Affine body dynamics: Fast, stable and intersection-free simulation of stiff materials. *ACM Transactions on Graphics*, 41 (4), 2022. doi: 10.1145/3528223.3530064.

Quentin Le Lidec, Wilson Jallet, Louis Montaut, Ivan Laptev, Cordelia Schmid, and Justin Carpentier. Contact models in robotics: a comparative analysis. *IEEE Transactions on Robotics*, 40:3716–3733, 2024.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020. doi: 10.1126/scirobotics.abc5986.

Michelle A Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *arXiv preprint arXiv:1907.13098*, 2019.

Somang Lee, Sunkyung Park, Jinhee Yun, Seoki An, and Dongjun Lee. CRISP: Contact-rich robotic simulation platform with extensive geometries and contact solvers, 2026. URL `https://arxiv.org/abs/2609.21761`.

Jingxian Lu, Wenke Xia, Yuxuan Wu, Zhiwu Lu, and Di Hu. When would vision-proprioception policies fail in robotic manipulation? 2026.

Jianlan Luo, Oleg Sushkov, Rugile Pevceviciute, Wenzhao Lian, Chang Su, Mel Vecerik, Ning Ye, Stefan Schaal, and Jon Scholz. Robust multi-modal policies for industrial assembly via reinforcement learning and demonstrations: A large-scale study. In *Robotics: Science and Systems*, 2021.

Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Zach Corse. Local optimization for robust signed distance field collision. In *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, volume 3, 2020. doi: 10.1145/3384538.

Khaled Mamou and Faouzi Ghorbel. A simple and efficient approach for 3D mesh approximate convex decomposition. In *IEEE International Conference on Image Processing*, pp. 3501–3504, 2009. doi: 10.1109/ICIP.2009.5414068.

Jeremy A Marvel, Roger Bostelman, and Joe Falco. Multi-robot assembly strategies and metrics. *ACM Computing Surveys (CSUR)*, 51(1):1–32, 2018.

Yashraj Narang, Kier Storey, Iretiayo Akinola, Miles Macklin, Philipp Reist, Lukasz Wawrzyniak, Yunrong Guo, Adam Moravanszky, Gavriel State, Michelle Lu, et al. Factory: Fast contact for robotic assembly. 2022.

Michael Noseworthy, Bingjie Tang, Bowen Wen, Ankur Handa, Chad Kessens, Nicholas Roy, Dieter Fox, Fabio Ramos, Yashraj Narang, and Iretiayo Akinola. Forge: Force-guided exploration for robust contact-rich manipulation under uncertainty, 2025.

Jiahe Pan, Stelian Coros, Jitendra Malik, and Toru Lin. Beyond binary: Sim-to-real dexterous manipulation with physics-aware contact representation.

Hyeonjun Park, Jaeheung Park, Dong-Hyuk Lee, Jae-Han Park, Moon-Hong Baeg, and Ji-Hun Bae. Compliance-based robotic peg-in-hole assembly strategy without force feedback. *IEEE Transactions on Industrial Electronics*, 64(8):6299–6309, 2017. doi: 10.1109/TIE.2017.2682002.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation*, 2018.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *Robotics: Science and Systems*, 2018.

Haozhi Qi, Ashish Kumar, Roberto Calandra, Yi Ma, and Jitendra Malik. In-hand object rotation via rapid motor adaptation. In *Conference on Robot Learning*, 2022.

Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 9(89):eadi9579, 2024.

Gerrit Schoettler, Ashvin Nair, Jianlan Luo, Shikhar Bahl, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Joohwan Seo, Arvind Kruthiventy, Soomi Lee, Megan Teng, Seoyeon Choi, Xiang Zhang, Jongeun Choi, and Roberto Horowitz. Equicontact: A hierarchical se (3) vision-to-force equivariant policy for spatially generalizable contact-rich tasks, 2025.

Jonah Siekmann, Yesh Godse, Alan Fern, and Jonathan Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE international conference on robotics and automation (ICRA)*, pp. 7309–7315. IEEE, 2021a.

Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. 2021b.

Oren Spector and Dotan Di Castro. InsertionNet: A scalable solution for insertion. *IEEE Robotics and Automation Letters*, 6(3):5509–5516, 2021.

Bingjie Tang, Michael A Lin, Iretiayo Akinola, Ankur Handa, Gaurav S Sukhatme, Fabio Ramos, Dieter Fox, and Yashraj Narang. Industreal: Transferring contact-rich assembly tasks from simulation to reality. 2023.

Bingjie Tang, Iretiayo Akinola, Jie Xu, Bowen Wen, Ankur Handa, Karl Van Wyk, Dieter Fox, Gaurav S. Sukhatme, Fabio Ramos, and Yashraj Narang. AutoMate: Specialist and generalist assembly policies over diverse geometries. In *Robotics: Science and Systems*, 2024.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. ManiSkill3: GPU parallelized robotics simulation and rendering for generalizable embodied AI, 2024. URL https://arxiv.org/abs/2410.00425.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. 2024.

The Newton developers. Newton: GPU-accelerated physics simulation for robotics. https://github.com/newton-physics/newton, 2025.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Yian Wang, Bingjie Tang, Chuang Gan, Dieter Fox, Kaichun Mo, Yashraj Narang, and Iretiayo Akinola. MatchMaker: Automated asset generation for robotic assembly. In *IEEE International Conference on Robotics and Automation*, 2025.

James Watson, Austin Miller, and Nikolaus Correll. Autonomous industrial assembly using force, torque, and rgb-d sensing. *Advanced Robotics*, 34(7-8):546–559, 2020.

Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3D meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics*, 41(4), 2022. doi: 10.1145/3528223.3530103.

Daniel E Whitney. Quasi-static assembly of compliantly supported rigid parts. 1982.

Kevin Zakka, Baruch Tabanpour, Qiayuan Liao, Mustafa Haiderbhai, Samuel Holt, Jing Yuan Luo, Arthur Allshire, Erik Frey, Koushil Sreenath, Lueder A Kahrs, et al. Mujoco playground, 2025.

Kevin Zakka, Qiayuan Liao, Brent Yi, Louis Le Lay, Koushil Sreenath, and Pieter Abbeel. mjlab: A lightweight framework for gpu-accelerated robot learning, 2026.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning, 2020. URL https://arxiv.org/abs/2009.12293.

A TRAINING DETAILS

Table 1: **Randomization ranges.** The controlled study (section 3) trains its policies on the first column; the deployed policy (section 3.3) trains on the second, whose scale, aspect-ratio, and clearance limits also bound the study's unseen-geometry tests. Ranges marked (log) are sampled log-uniformly and the rest uniformly. Geometry is drawn once per world; poses and contact parameters at every reset. Clearance is per side, independent of scale. No observation noise is added.

Parameter	Study policies Training	Deployed Training
Part and fixture		
Assembly scale (relative to base stadium)	1.35–2.45	0.8–3.0
Stadium aspect ratio, area-preserving (log)	1.25–2.5	1.25–5.0
Opening, long × short side	38–91 × 20–54 mm	22–156 × 8–67 mm
Per-side clearance	1.0–1.5 mm	0.5–1.5 mm
Peg length	50–60 mm × scale	55–150 mm
Hole depth	10 mm × scale	20 mm
Plate side	80 mm × scale	80 mm × scale
Peg mass	0.35–0.675 kg	0.35–1.0 kg
Contact		
Plate sliding friction (log)	0.5–2.0	0.25–4.0
Plate contact time constant (log)	3.5–7.1 ms	2.5–10 ms
Finger friction scale (log)	1.41–2.0	1.0–2.0
Finger contact time-constant scale (log)	0.78–1.34	0.6–1.75
Gripper force cap scale (log)	1.41–2.83	1.0–4.0
Poses		
Fixture center from robot base, x / y / z	0.39–0.83 / -0.46–0.46 / -0.47–0.49 m, all policies	
Fixture roll, pitch, yaw	$\pm 5^\circ$	$\pm 10^\circ$
Gripper rotation, per axis	$\pm 5^\circ$	$\pm 10^\circ$
Initial peg position, per axis	± 12.5 mm	± 25 mm
Grasp position along the peg	± 5.5 mm	± 11 mm
In-hand tilt about the finger axis	$\pm 10^\circ$	$\pm 20^\circ$
Control		
Joint impedance gains k_p / k_d; action scale	80 / 20; 0.02 rad	40 / 20; 0.04 rad
Joint speed and torque penalty	none	staged

Task. Physics runs at 1 kHz under a 50 Hz policy. Starts lie a median 21 mm from the hole center, with the peg axis tilted a median 14° from the hole's. Success requires all 64 sampled points on the peg's bottom edge to be inside the opening and within 1 mm of the hole floor. Training geometries are drawn from three banks of 4,096.

Architecture. The actor observes 33 values per step: seven joint positions, seven joint velocities, the gripper pose and twist (twelve), and its previous action (seven). A temporal convolutional encoder maps the last 30 observations (0.6 s) to a 32-dimensional latent; an MLP (256, 256, 128 units) maps the latent and the current observation to seven joint-position increments, scaled by the action scale of Table 1 and added to the measured joint positions. The critic has 240 inputs. The reward is $+1$ on first capture and $+1$ on success. The deployed policy's safety penalty starts once success passes 75% and plateaus, penalizing joint speeds and commanded torques above one third, and later one sixth, of the hardware limits. Contact labels mark each of the 64 points as free (more than 1 mm above the fixture's top face), in contact (on the rim or top face, within 1 mm), or inside the opening. In 512 fresh episodes, the deployed policy's decoder recovers them with 76% balanced accuracy (chance 33%), and 92% of the points inside the opening.

PPO. 12,288 worlds, 32-step rollouts, 5 epochs of 20 minibatches, learning rate 3×10^{-4} (fixed), clip 0.2, $\gamma = 0.98$, $\lambda = 0.95$, entropy coefficient 0.003, value-loss coefficient 1 with clipping, gradient norm clipped at 1, target KL 0.01, and initial action standard deviation 0.5. The contact decoder has two hidden layers of 128 units and class weights 0.25 (free), 3.0 (contact), and 2.0 (inside). An episode ends on success, when the peg moves far from the fixture, or after 5 s.

B COLLISION BENCHMARK DETAILS

Table 2: **Passive probes.** Errors are in micrometers of wall position; negative values mean that phantom material stopped the peg early.

Probe	Setup	Exact answer
Wall offset	Rays from the hole axis at 720 angles and 11 heights; no solver	Wall on the analytic outline
S0 contact onset	Aligned peg placed at known gaps; collision only	Contact appears at gap 0
S1 drop	centered free peg dropped from 1 mm; c from $+1$ mm to -500 µm	Passes with zero wall contact iff $c > 0$
S2 lateral play	Planar slides; 0.1 N push in 24 directions	Support of the feasible translation set
S3 tilt play	Slides and one hinge; 1.5 mN m torque	Maximum tilt at which both faces fit
S4 yaw play	Slides and a vertical hinge (square hole)	$\arcsin(a/(s\sqrt{2})) - \pi/4$

Table 3: **Real-time factor** at a 1 ms time step with the peg pressed against the wall; ranges span each method's settings, best of multi-point collision on and off. GPU figures sum over 512 MuJoCo Warp worlds on one RTX 5090. The last column gives the worst settled error at a 20 µm radial clearance.

Representation	CPU, 1 thread	GPU, 512 worlds	Error at $c = 20$ µm
`mjfit`, square hole	93×	970×	0.2 µm
`mjfit`, circle, 7–32 sides	82–29×	1120–1070×	≤2.6 µm
`mjfit`, circle, 63–281 sides	16–4.9×	875–442×	≤0.4 µm
Mesh SDF, octree depth 5–8	2.9–2.2×	590–840×	fills, or 23–79 µm
CoACD, threshold 0.01 (839 hulls)	1.8×	254×	fills
Other CoACD and V-HACD settings	3.3–70×	680–1900×	fills

All scenes use a 1 ms time step, the implicit-fast integrator, the Newton solver, $\text{solref} = (2\,\text{ms}, 1)$, $\text{solimp} = (0.99, 0.999, 0.001)$, $\mu = 1$, and zero margin. Baselines receive an exact, watertight plate mesh (the circle tessellated at 1,024 segments, 0.04 µm sagitta), and ground truth uses the analytic circle, so `mjfit` is charged for its own chord error. The peg is always an exact MuJoCo primitive, so every difference comes from the hole. Timings come from one idle machine (Intel i9-13900KS, NVIDIA RTX 5090). Warp contact and constraint buffers are calibrated per representation, and every GPU run is checked for buffer overflow, which would silently drop contacts. At a 2 ms contact time constant, frictionless drops let pegs up to 20 µm oversize slip in for every method, including exact geometry, so S1 cannot resolve geometric errors below about 20 µm; the play probes can.

Setup. The round and square holes are 16 mm wide in a 15 mm plate, with radial clearance c from 20 to 200 µm; at $c = 20$ µm, $2c/D = 0.0025$. The M10×1.5 bolt and nut have flank clearance c of 20 and 100 µm and are probed for axial, radial, and tilt play and for screwing under load. CPU times use one environment on one thread; GPU times use 512 MuJoCo Warp worlds, except in figure 5b, which uses 256 and measures the error on Warp as well.

C ADDITIONAL RESULTS



Figure 11: **Study peg cross-sections, to scale.** The study policies train on stadiums; the unseen random convex pegs used only for evaluation have one vertex on each side of a matching stadium's bounding box and the same clearance.

Table 4: **In-distribution behaviour** (section 3.1), from 8,192 episodes (success, time, rate) and from the 4,045 of 4,096 matched starts where both policies succeed (travel, engagement, tilt). Medians unless stated; insertions per minute count every attempt, with failures at their full 5 s. Starts lie a median 10 mm from the hole center.

	Blind proprioceptive (ours)	Visual
Success (%)	99.0	99.5
Time to seat (s)	1.05	0.36
Insertions per minute	51	137
Lateral travel before seating (mm)	41	21
Time partly engaged (s)	0.54	0.16
Tilt when partly engaged / seated (°)	7.9 / 1.2	6.2 / 1.2

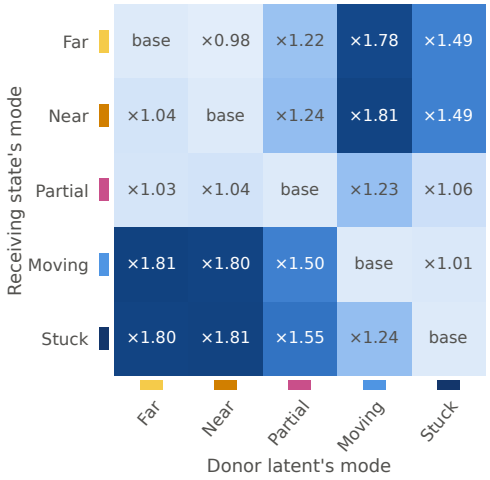

Figure 12: **Latent-swap test** for the deployed policy in simulation. Each state keeps its own current observation but receives the history latent z_t of a state in another mode; each cell is the mean change in action relative to a swap within the receiving state's own mode.

Behaviour. The blind policy travels farther than the visual policy from 98% of matched starts (Table 4). On unseen random convex pegs, the blind policy succeeds in 35.9% of episodes and the visual policy in 3.3%; these pegs have a median 53% of the matching stadium's area.

Probe details. The search modes are split at 5 mm from the opening, and an inside peg is stuck when it progresses less than 0.5 mm in 0.2 s. We roll out the deployed policy in 1,536 worlds of its training distribution (1.15M states), train the probes on even-numbered worlds, and test on odd ones. Balanced accuracy over the seven modes is 70% (chance 14%) from the history latent, against 57% from the current observation alone and 36% from a randomly initialized encoder. The history latent recognizes lifted, moving, and stuck pegs with 83–91% recall; near and far are confused 27–28% of the time in each direction. Swapping latents between a searching and an inside state changes the action 1.5–1.8× as much as swapping between two states of the same mode; swapping between near and far changes it 0.98–1.04× as much (figure 12). On the robot, the policy's actions recomputed from the logged observations match the logged ones to within 10^{-5}. At the end of successful trials, the peg pose estimated from the joint encoders and planned grasp is a median 4.2 mm (10th–90th percentile 1.7–11.8 mm) from the seated pose and places only 1% of their states fully inside the hole. Scored states have every point of the peg's bottom edge at least 3 mm above the plate (lifted) or at least 3 mm inside the opening (inside). Over the 90 stadium trials, the probe labels 94% of these lifted and 91% of these inside states correctly, against 99.7% and 99.1% in simulation from the same start states, and labels the final state of 68 of the 73 successful trials as inside. In figure 8b, depth is the mean depth of the peg's bottom edge below the plate surface, estimated in the real world from the joint encoders and the planned grasp; in simulation, the start state shown is near the median of the 90 in agreement between true and decoded modes.

Unseen sizes, aspect ratios, and clearances. The unseen parts of figure 9 widen one property at a time: a per-side clearance of 0.5–0.625 mm (tighter; training 1.0–1.5 mm); an aspect ratio of 3.54–4.20 (slender) or 4.20–5.0 (very slender; training 1.25–2.5); or a scale of 1.075–1.35 or 2.45–2.725 times the base stadium (mild) or 0.8–1.075 or 2.725–3.0 (extreme; training 1.35–2.45). Each widened bar pools about 1,000 episodes (2,000 for size). In these single-axis evaluations, both policies succeed in 99.0–99.6% of episodes whose widened value lies inside its training range (not shown), matching their in-distribution success. The blind and visual policies seat 95.8% and 78.3% of slender and 90.9% and 49.8% of very slender parts; 92.2% and 87.4% of the tighter fits; and 96.3% and 72.9% of mildly and 87.8% and 21.3% of extremely resized ones. Split by side, the visual policy seats 57.8% of parts just smaller than training and 88.4% of those just larger, and 0.8% of the smallest and 42.8% of the largest, against 95.0%, 97.5%, 82.9%, and 93.1% for the blind policy. The aspect ratios and clearances between the shown bars and the training range (2.5–3.54 and 0.625–1.0 mm) cost either policy at most 6 points.

Real-world pegs. The stadiums in figure 10 have openings of 60×21, 80×28 (weighted), 100×35, and 101×20 mm, and the circle is Ø37 mm. The stadiums' cross-section, size, and clearance lie inside the deployed policy's training ranges (Table 1); every real peg weighs less than the lightest simulated peg (0.35 kg). Each simulated rate is the same policy over 2,048 episodes on that peg's cross-section, size, clearance, length, and mass. Real success is over operator-labelled trials; pooled over the five pegs, 112 of 151 succeed. One further circle trial ended when the robot's state stream timed out and is excluded.

Checkpoints. Blind policy: PPO updates 657 and 662; visual policy: 733. A proprioceptive variant without joint velocities (20 features: joint positions, gripper pose, previous action; updates 1661 and 1511) reaches 96.5% in distribution and 21.5% on random convex pegs.