JACK: Joint Axis Correction from Kinesthetic Feedback

Arthur Jakobsson[1,*] Eugene Lee[1] Bardienus Pieter Duisterhof[1] Huang Huang[2]
Jeff Tan[2] Romina Mir[1,3] Jonathan Francis[1,3] Jean Oh[1] Jeffrey Ichnowski[1]

[1]Robotics Institute, Carnegie Mellon University [2]Stanford University
[3]Bosch Center for Artificial Intelligence

[*]Corresponding author: ajakobss@cs.cmu.edu

Abstract: What should a robot's world model of an articulated object contain? We argue for an explicit, revisable belief over the object's kinematic structure: for each joint, a type, an axis direction and a pivot point, held as a mixture of candidate explanations of the motion the robot has caused, each with a calibrated angular and positional spread. Because the belief predicts how each part will move, even a single interaction can disprove or improve our model's understanding of it. We describe a formulation in which the robot proposes the object's joints from its first point cloud, tests them by pushing parts and tracking points on their surfaces, and chooses the next action for the robot that it believes will maximize information on the object's structure. This approach seeks to evaluate problems that are otherwise difficult for articulation estimation methods, such as joints glued shut, mislabeled or deceptive joint types, and joint chains (e.g. a toy tractor's arm). We discuss the formulation and the representation that we choose to enable such exploration.

Keywords: Articulated objects, Explicit world models, Interactive perception

1 Introduction

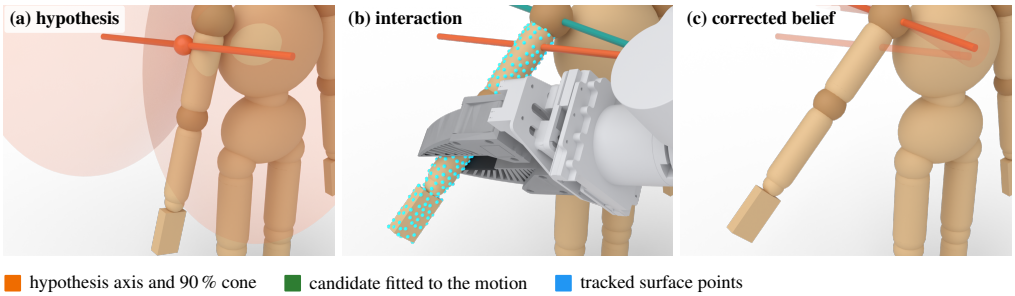


Figure 1: **JACK.** (a) A joint estimated from a single view is a hypothesis with a wide 90 % cone around its axis. (b) The robot moves the part and tracks points on its surface. (c) The belief weighs the hypothesis against axes fitted to the tracked motion and returns a corrected joint with a narrower, calibrated cone.

Cabinet doors, drawers, laptop lids and desk lamps are rigid parts connected by joints, and a robot must know each joint's type, axis and pivot before it can open or move them safely. Recent methods estimate joints from point clouds [1, 2], vision-language models [3], or RGB-D video [4], but on unfamiliar or ambiguous objects these estimates can be wrong, and a pull along a misestimated axis presses the part against its hinge or rail. We therefore treat an estimated joint as a hypothesis that the robot tests by moving the part and corrects from the motion it observes.

To act on an articulated object, a robot must predict how each part will move when pushed. We argue that the world model for this task should be an explicit, revisable belief over kinematic structure, not a predictor of future pixels or latent states rolled forward in time. The model has four elements: a hypothesis for each joint, the tracked motion that tests it, a confidence stating how far the hypothesis can be trusted, and a rule for choosing the next action so that error and uncertainty both fall over

Do Robots Need World Models? Workshop, 10th Conference on Robot Learning (CoRL 2026).

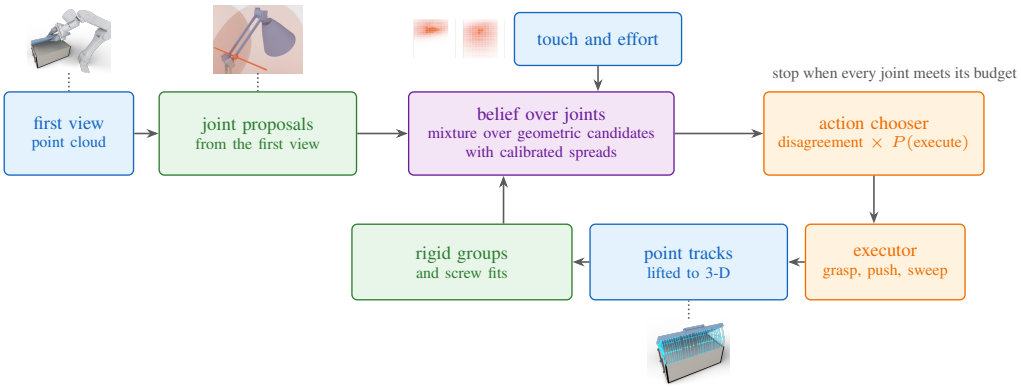


Figure 2: **The belief as the world model.** From the first point cloud, the robot proposes the object's joints and holds them as unordered slots, each a mixture of candidate axes with calibrated spreads. The chooser selects the action whose outcome the belief's samples disagree about most, weighted by the probability that it executes. Tracks from the action are lifted to 3-D, grouped into rigid parts and fitted with screw motions, and the fits return to the belief as new candidates.

a sequence of actions. Because the belief is stated as joint types, axes and pivots, one push can confirm or refute it and any planner can use it. Representing the joints is the easier part of such a model; choosing actions from the belief is the harder part.

Section 3 describes the formulation, Section 4 lists considerations for training it as an explicit world model, and Section 6 lists open problems.

2 Related work

Articulation estimation from vision alone includes regression from depth [5, 6], point clouds before and after an interaction [7], reconstruction with joint refinement [8, 9], language models [10, 3], and RGB-D video [4, 11]. All of these estimate from observations recorded before the robot acts, and we use their output as an optional prior. Interactive perception [12] uses the robot's actions to generate the observations it needs. Classical methods keep an explicit belief over structure: Sturm et al. [13] select among kinematic models of tracked motion, Martín-Martín and Brock [14] fuse vision, proprioception and force in coupled recursive estimators, and Hausman et al. [15] choose actions by information gain over a particle filter of mechanisms [see also 16, 17]. Learned policies act to reveal articulation [18, 19, 20, 21, 22, 23, 24], and online estimators combine a learned prior with interaction [25]. Learned world models for manipulation predict future pixels or latent states [26, 27] or particle dynamics [28] and roll them forward to plan. Our belief instead predicts how each part moves under the robot's action, is updated after each action, and is tested by the next. One push can refute a belief of this kind and a planner can use it directly; a latent predictor needs a decoder to offer either. ForceTwin [29] builds physics-informed twins from a person probing objects with an instrumented gripper. For deformable objects, PhysCoRe [30] corrects an analytic physical model with a learned residual and a per-part confidence; this motivates our approach of correcting a joint hypothesis with a residual whose confidence improves with each action.

3 Belief Based World Model

Initial hypothesis. From the first point cloud of the object and the number of joints, a network proposes a set of candidate joints, each with a type (revolute or prismatic), an axis direction and a pivot point. The belief holds one slot per joint, initialised from these proposals. The slots are unordered and are matched to the true joints only inside the training loss, so at inference no slot is tied to a particular part. A joint estimate from an external method, such as Flowbot++ [1], can be added as an optional prior. A proposal and an external estimate enter the belief the same way, as one mixture component that later evidence can confirm, adjust or outweigh. The proposer sees only geometry, so the belief learns which part a joint moves only once the robot has moved that part.

Motion evidence. The belief is updated from the motion of the object's surface, not from its appearance. A push moves a part by a few millimetres while the arm itself moves through the same images, so we compute that motion explicitly instead of asking a network to find it in images, touch and proprioception. Point tracks from CoTracker3 [31], fine-tuned on simulated ground truth, are lifted to 3-D and pooled across actions. Points that move rigidly together are grouped, and each group's motion is fitted with a closed-form screw fit and robust refinement, giving a rotation axis or a translation direction in the world frame and relative to the parent part. The fit does not use the hypothesis, so it checks the hypothesis independently. Like Sturm et al. [13], we fit joints to tracked motion; unlike them, each fit is a candidate that the learned belief weighs against the hypothesis and the other fits, not the final estimate.

Belief update. Each joint's belief is a mixture of the hypothesis plus a learned residual, each rigid-motion fit in the world frame and relative to the parent part, the belief from before the latest action, and a uniform outlier term. Each component is a Gaussian in the tangent plane of its axis: the axis error is an angle and the pivot error is the distance to the true axis line. The point estimate is the highest-weight component. A transformer takes tokens for the keyframes and the geometric candidates and updates the joint slots. Tactile and joint-effort readings enter as further tokens, since they can detect contact, joint limits and resistance that tracked motion cannot; in our experiments the geometric candidates carry nearly all of the information, and we make no claim that touch improves the estimate. The belief is trained on the likelihood of the true joint under the mixture; for the axis, with θ_k the sign-free angle to component k and u the outlier density,

$$\mathcal{L}_{\text{axis}} = -\ln\Big(\sum_k w_k \,\tfrac{1}{2\pi\sigma_k^2}\, e^{-\theta_k^2/2\sigma_k^2} + w_{\text{out}}\, u\Big), \tag{1}$$

and for the pivot with the same form on the distance to the true axis line. A spread that is too narrow is penalised by the exponent and one that is too wide by the normaliser, so the loss is lowest when the spread matches the model's error on similar inputs. We reweight the loss so that the estimate keeps improving after the spread has widened to cover its error [32]. The robot stops when every joint's axis, pivot and type meet a confidence budget.

Action selection. Each candidate action is a choice of joint, contact point and push direction. Samples drawn from the belief each predict how the contacted part would move under the action. An action for which all samples predict the same motion cannot change the belief; one for which they predict different motions can. The chooser scores each action by this disagreement multiplied by the predicted probability that the action executes, which discounts actions that would be informative but are likely to fail, for example because the grasp would slip or the contact point is off the part. The same belief thus explains past motion and predicts the outcome of the next action. Training data come from a scripted teacher whose grasps, pushes and sweeps are defined relative to the current hypothesis, in simulated worlds with randomised placement, joint physics and hypothesis error.

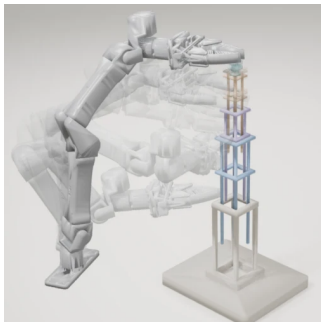


Figure 3: A series of actions on an articulated satellite mast.

The formulation handles revolute and prismatic joints, including chains in which a joint's parent also moves, such as the two arms of a desk lamp, through the fits relative to the parent part. It assumes rigid parts, a robot that can reach and push them, and a known number of joints.

4 Training considerations

Leaky actions. If the teacher that generates training actions uses the true joint, or a hypothesis close to it, the action itself carries the answer: a push along the true axis reveals the axis before any motion is observed. A model trained on such data learns to read the action instead of the outcome and fails once actions come from its own, possibly wrong, belief. Training actions, like training inputs, should be computed only from what the robot has at test time; the same holds for simulator quantities such as part labels and canonical axes.

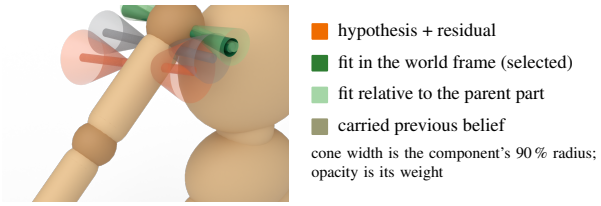


Figure 4: **One joint's belief.** The elbow of the mannequin's raised arm, after the upper arm has moved. Each candidate axis has its own spread (cone width) and weight (opacity); the point estimate is the highest-weight candidate.

Considering Training Data with Planning. A learned action chooser is worth its cost only when actions differ in what they reveal. For a single drawer, almost any pull on the handle identifies the joint, and a scripted action does as well as a planned one. Planning matters when a contact can miss the moving part, when one joint's motion hides another's, as in a chain, or when each action is slow or risky. A chooser should be compared against scripted and random actions on the same objects before gains are attributed to it.

Starting errors. The starting errors seen in training set what the model can correct. If training hypotheses all lie near the truth, the model learns small corrections and keeps a grossly wrong hypothesis, such as a hinge on the wrong edge of a door or a joint assigned to the wrong part. Training should include far-wrong starts and wrong assignments.

Failed actions. A push that moves nothing shows that the part does not move in that direction from that contact. Blocked and slipping actions belong in the training data and should update the belief; otherwise the chooser proposes the same action again.

5 Discussion: Do Robots Need World Models?

We loop back to address the driving question: *Do Robots Need World Models?* We argue that for complex interactions the answer is undeniably yes, and we believe that the best world models should be able to adapt using information gathered from the robot's interactions. We introduce an early form of a method to explicitly reason about the kinematic structure of articulated objects with a robot in the loop to perform probing actions and improve our world model. We believe that this is a step in the right direction to enable robots to reason about the world and adapt to new situations. One future direction is passive information gathering, which uses the same backbone but extracts information from actions already performed; another is mixed passive–active policies that combine acting to observe with acting to achieve a goal.

6 Open problems

The next step for explicit models is explicit structure: a parent–child graph over joints, so that a lamp's forearm joint is explained relative to its upper arm, and per-joint part masks on the current geometry instead of the first frame, so that the belief records which surface each joint moves. Further open problems are consistently providing informative actions with a single arm; a potential future direction is bimanual interaction to improve object stability.

Conclusion. For articulated objects, an explicit belief over joint types, axes and pivots is a useful world model because a single push can test it and a planner can use it directly. The harder problem is choosing which push to make.

Acknowledgments

This work used computational resources at Pittsburgh Supercomputing Center through allocation CIS251423 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296 [33].

References

[1] H. Zhang, B. Eisner, and D. Held. FlowBot++: Learning generalized articulated objects manipulation via articulation projection. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1222–1241. PMLR, 2023. URL https://proceedings.mlr.press/v229/zhang23c.html.

[2] Y. Li, W. H. Leng, Y. Fang, B. Eisner, and D. Held. FlowBotHD: History-aware diffuser handling ambiguities in articulated objects manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

[3] L. Le, J. Xie, W. Liang, H.-J. Wang, Y. Yang, Y. J. Ma, K. Vedder, A. Krishna, D. Jayaraman, and E. Eaton. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=s3FTX4Ay55.

[4] A. Werby, M. Büchner, A. Röfer, C. Huang, W. Burgard, and A. Valada. Articulated object estimation in the wild. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 3828–3849. PMLR, 2025. URL https://proceedings.mlr.press/v305/werby25a.html.

[5] X. Li, H. Wang, L. Yi, L. Guibas, A. L. Abbott, and S. Song. Category-level articulated object pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[6] A. Jain, R. Lioutikov, C. Chuck, and S. Niekum. ScrewNet: Category-independent articulation model estimation from depth images using screw theory. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[7] Z. Jiang, C.-C. Hsu, and Y. Zhu. Ditto: Building digital twins of articulated objects from interaction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[8] J. Liu, A. Mahdavi-Amiri, and M. Savva. PARIS: Part-level reconstruction and motion analysis for articulated objects. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[9] Q. Yu, X. Yuan, Y. Jiang, J. Chen, D. Zheng, C. Hao, Y. You, Y. Chen, Y. Mu, L. Liu, and C. Lu. ArtGS: 3d gaussian splatting for interactive visual-physical modeling and manipulation of articulated objects. *arXiv preprint arXiv:2507.02600*, 2025.

[10] Z. Mandi, Y. Weng, D. Bauer, and S. Song. Real2Code: Reconstruct articulated objects via code generation. *arXiv preprint arXiv:2406.08474*, 2024.

[11] P. Mishra, R. Zurbrügg, M. Wilder-Smith, M. Hutter, M. Pollefeys, Z. Bauer, and H. Blum. ArtiTwinSplat: Interactable digital twin reconstruction via gaussian splatting from rgb-d videos. *arXiv preprint arXiv:2606.24628*, 2026.

[12] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

[13] J. Sturm, C. Stachniss, and W. Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41:477–526, 2011.

[14] R. Martín-Martín and O. Brock. Coupled recursive estimation for online interactive perception of articulated objects. *The International Journal of Robotics Research*, 41(8):741–777, 2022.

[15] K. Hausman, S. Niekum, S. Osentoski, and G. S. Sukhatme. Active articulation model estimation through interactive perception. In *IEEE International Conference on Robotics and Automation*, pages 3305–3312. IEEE, 2015. doi:10.1109/ICRA.2015.7139655.

[16] D. Katz and O. Brock. Manipulating articulated objects with interactive perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[17] P. R. Barragán, L. P. Kaelbling, and T. Lozano-Pérez. Interactive bayesian identification of kinematic mechanisms. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2013–2020, 2014.

[18] K. Mo, L. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani. Where2Act: From pixels to actions for articulated 3d objects. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[19] B. Eisner, H. Zhang, and D. Held. FlowBot3D: Learning 3d articulation flow to manipulate articulated objects. In *Robotics: Science and Systems (RSS)*, 2022.

[20] Y. Wang, R. Wu, K. Mo, J. Ke, Q. Fan, L. Guibas, and H. Dong. AdaAfford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions. In *European Conference on Computer Vision (ECCV)*, 2022.

[21] S. Y. Gadre, K. Ehsani, and S. Song. Act the part: Learning interaction strategies for articulated object part discovery. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[22] C.-C. Hsu, Z. Jiang, and Y. Zhu. Ditto in the house: Building articulation models of indoor scenes through interactive perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[23] T. Engelbracht, R. Zurbrügg, M. Pollefeys, H. Blum, and Z. Bauer. SpotLight: Robotic scene understanding through interaction and affordance detection. *arXiv preprint arXiv:2409.11870*, 2024.

[24] M. Memmel, A. Wagenmaker, C. Zhu, D. Fox, and A. Gupta. ASID: Active exploration for system identification in robotic manipulation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/ae8afec5dabb07091d8f85128250f2ff-Abstract-Conference.html.

[25] R. Buchanan, A. Röfer, J. Pousa de Moura, A. Valada, and S. Vijayakumar. Online estimation and manipulation of articulated objects. *Autonomous Robots*, 50(13):1–19, 2026. doi:10.1007/s10514-025-10233-4.

[26] D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[27] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[28] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations (ICLR)*, 2019.

[29] T. Engelbracht, R. Zurbrügg, M. Mittal, M. Hutter, M. Pollefeys, H. Blum, and Z. Bauer. ForceTwin: Physics-informed digital twins for robotic manipulation from instrumented human interaction. *arXiv preprint arXiv:2609.21751*, 2026.

[30] H. Yin, S. Tao, Y. Chen, and L. Gan. PhysCoRe: Physics-corrected residual world models for material-aware deformable dynamics. In *Conference on Robot Learning (CoRL)*, Proceedings of Machine Learning Research. PMLR, 2026. arXiv:2607.20653.

[31] N. Karaev, I. Makarov, J. Wang, N. Neverova, A. Vedaldi, and C. Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.

[32] M. Seitzer, A. Tavakoli, D. Antic, and G. Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

[33] T. J. Boerner, S. Deems, T. R. Furlani, S. L. Knuth, and J. Towns. ACCESS: Advancing innovation: NSF's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing 2023: Computing for the Common Good*, pages 173–176. Association for Computing Machinery, 2023. doi: 10.1145/3569951.3597559.