ATK: Automatic Task-driven Keypoint selection for Policy Transfer from Simulation to Real World

Yunchu Zhang, Zhengyu Zhang, Liyiming Ke, Siddhartha Srinivasa, Abhishek Gupta
Paul G. Allen School of Computer Science and Engineering
University of Washington United States
{yunchuz,octipus,kayke,siddh,abhgupta}@cs.washington.edu

Abstract: Transferring robotic policies from simulation to the real world often faces perceptual challenges, where visual differences degrade performance. Policies relying on 6D pose state estimation, require task-specific scaffolding, while raw sensor-based policies lack robustness and efficiency. We propose using 2D keypoints—spatially consistent features in the image frame—as a state representation for effective sim-to-real transfer. Our method, ATK, automatically selects minimal set of task-relevant keypoints that predict optimal behavior. By distilling a teacher policy trained in simulation into a student policy operating on RGB images while tracking the selected keypoints, our system effectively tracks keypoints and transfers policies to the real world, even under perceptual challenges like transparent objects or fine-grained manipulation. We validate ATK across various tasks, showing the minimal set of task-relevant keypoint representations improved robustness to visual and environmental variations.

Keywords: representation learning, sim2real transfer

1 Introduction

Simulation has become an essential tool in modern robotics, offering low-cost data for developing policies in domains like manipulation. However, transferring these policies to real-world hardware is challenging due to the *sim-to-real gap*, caused by discrepancies in physics and perception. In this work, we focus on bridging the *perceptual gap* between simulation and reality - to perceive and *represent* the state of the world.

Prior works have studied this by varying the input modality from depth images [1, 2], 3D point clouds [3], learned latent spaces [4, 5] to pose-based estimation [6, 7]. While promising, these methods would require specialized sensors, environment-specific engineering or task-specific setup. Moreover, they may still struggle with sensor noise, transparent surfaces, and deformable or small objects. To achieve robust sim-to-real transfer, we propose using *keypoints*—2D pixel points in RGB images that can be tracked over time—as a versatile state representation for robotic policies. Keypoints, widely used in computer vision [8, 9] for object tracking, offer resilience to occlusion, lighting changes, and scale variations. Keypoints do not rely on rigid structures, making them more suitable for tracking articulated and non-rigid objects. Additionally, keypoints naturally generalize to transparent and fine-grained objects better than depth-based approaches. Recent advances in keypoint tracking, driven by models trained on large-scale web data [8, 10], have made keypoint tracking surprisingly robust across diverse visual domains, making it a promising candidate for bridging the sim-to-real perceptual gap and rendering policies robust to visual variations.

The critical question then becomes: What is the minimal set of task-relevant keypoints that can serve as an effective state representation for decision-making? Simply using all keypoints in a scene introduces redundancy, increasing computational burden and complicating the tracking problem because of occlusion and point interferences. Random sampling points or selecting too few points may fail to include critical task-relevant information. The ideal set is task-specific, as different tasks

focus on different scene elements. For example, a robot pressing a clock button or rotating a clock hand focuses on different scene parts (shown in Fig 2), suggesting that *the minimal set of keypoints must be inherently task-driven*. A task-driven representation also ignores irrelevant elements, like distractors or background changes, enhancing policy robustness.

Our key insight is that **task objectives can guide the selection of a compact representation and optimal policy**. The minimal set of task-relevant keypoints should predict the optimal policy, but this creates a challenge: the optimal policy requires a good state representation, and vice versa. To solve this, we leverage simulation's privileged information, such as Lagrangian state variables. We propose ATK, a distillation-based algorithm that selects minimal keypoints in simulation, trains a policy using these keypoints, and transfers it to the real world. Specifically, a teacher policy trained with privileged data in simulation is distilled into a student policy using only RGB images, tracking a minimal set of keypoints. This approach retains the necessary task-relevant information and ensures robust sim-to-real transfer without dedicated sensors and handles real-world visual disturbances during deployment. We validate our approach across various real-world manipulation tasks, including cluttered environments, transparent objects, and fine manipulation. Our keypoint-based representation improves performance by 60% over other perceptual representations in both the transfer and robustness capabilities of these representations in real-world environments.

2 Method

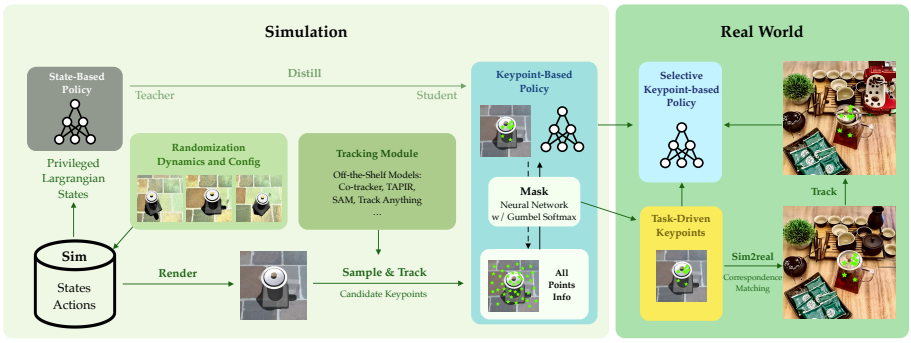

Figure 1: An overview of ATK framework. In simulation, ATK selects minimal task-relevant information through student-teacher policy distillation and selection mask optimization. The selected keypoints are transferred to the real world via the correspondence function $h_{\mathcal{C}}$. Finally, the student policy is transferred to the real world taking in RGB images while tracking the transferred keypoints.

We aim to provide a representation that can enable policy generalization and robustness in bridging the sim-to-real observation gap. To this end, we propose the use of **2D keypoints** as the perceptual representation for sim-to-real transfer (Sec. 2.1). The crux of our proposal lies in transferring task-relevant parts of the observation by automatically *selecting* a set of task-relevant keypoints. We propose ATK that integrates keypoint selection with policy training with a distillation process (Sec. 2.2). We then describe (Sec. 2.3) how to transfer the keypoints and policies proposed in simulation to the real world for robot deployment.

2.1 Keypoints as Policy Representations

Keypoints, commonly used in computer vision, are distinct points $k_t^i = (x_t^i, y_t^i)$ in the 2D image plane at time t. A set of N keypoints, $\{k_t^i\}_{i=1}^N$, provides a compact representation of the scene. The number and selection of keypoints can be dynamically adjusted based on task complexity and requirements, making it computationally efficient for robotic tasks. Keypoints do not require explicit knowledge of object and are thus highly versatile and applicable even to deformable or fine-grained materials. To track keypoints over time, we initialize keypoints at $t = 0$ and use tracking methods (track-anything [11], co-tracker [9] and TAPIR [8]), which maintain robust semantic correspondence across time steps. The tracking process can be formalized as a correspondence function $h_{\mathcal{C}}$, which

updates the keypoints $\{k_t^i\}_{i=1}^N$ at each time step while providing correspondence measurement scores. A key challenge in leveraging keypoint is to select and track them. Different tasks require focusing on distinct elements in the scene, the primary challenge then becomes: (1) selecting keypoints in simulation and (2) robustly tracking and transferring these keypoints and resulting keypoint-based policies from simulation to the real world.

2.2 Task-Driven Distillation

Randomly sampling or using too many keypoints adds unnecessary complexity and can reduce policy robustness due to interference, mismatches, and tracking failures. Instead, we select a minimal, task-driven set of keypoints by leveraging the task objective to evaluate the performance of candidate keypoint-based policies. This ensures that only the most relevant and useful points are tracked, optimizing both computational efficiency and transfer performance. The key criteria for minimal keypoint selection include: **(1) Realizability of the Optimal Policy**: The selected keypoints must capture all necessary information to learn an effective policy for the task. **(2) Trackability**: The chosen keypoints must be reliably trackable using the correspondence function $h_{\mathcal{C}}$. We leverage privileged information in simulation to guide the keypoint selection through a student-teacher policy distillation process. The optimal policy $\pi^*(a_t|s_t)$ is derived from Lagrangian state-based representations in the simulation, and DAgger [12] is used to train a keypoint-based policy $\pi_\theta^k(a_t \mid \{k_t^i\}_{i=1}^N)$, ensuring that the selected keypoints are predictive of the optimal actions a_t^*. The learning objective is then formulated and solved using gradient descent, as detailed in the appendix 5.3.

2.3 Transfer from Simulation to Reality

Once the keypoints proposal are finalized in the simulation, the next step is to transfer them to the real world by establishing correspondence with their real-world counterparts. Sim-to-real perceptual gaps may exist when using other representations. However, following our proposal, once keypoints are identified *at the start of the task*, subsequent tracking remains unaffected by the visual gap. This allows us to focus solely on transferring the *initial* set of keypoints. Be leveraging the simulator, we sample transformed or jittered views to better align the simulated view with the real world. Using the correspondence function with confidence scores for keypoint pairs, we select the simulation and real-world keypoint pair with the highest score, ensuring accurate keypoint matching for real-world policy execution. The sampling process, utilizing diffusion-model-based features [10, 13] trained on large-scale web data, ensures confident and robust semantic correspondence matching, even under visual disturbances, across different real-world configurations.

3 Experiments

Our evaluation aims to answer the following questions: **(1) Sim-to-real transfer:** How well do keypoints and policies *transfer* to the real world, based on policy success rates? **(2) Keypoint Selection Effectiveness:** Does the selection method keep policies *robust* and *generalizable* to changes like object placement, appearance, and distractors? **(3) Representation Sufficiency:** Does the representation capture enough information for high task success rates? **(4) Interpretability and Task-Relevant Features:** Are the selected keypoints interpretable and relevant to different task objectives in complex environments? In Appendix 5.4, we provide detailed explanations of the task motivations, experimental design, baseline methods, and evaluation procedures under various conditions.

Q1 & Q2: ATK **excels in Sim-to-real Transfer and Robustness over Baselines** As shown in Table 1, keypoint-based policies maintain high success rates in the real world compared to alternative modalities, showcasing strong resilience against randomized object poses or background variations. Altough extreme distractions, such as flashing light or occlusions, can disrupt tracking and decrease the performance, our method consistently outperform RGB, depth, and point-cloud based policies. The gap is worth-noting in tasks involving transparent objects (e.g., glass) and fine-grained

manipulation (e.g., clock tasks). This demonstrates the effectiveness of keypoint-based policies across a wide range of conditions.

	Sushi Pick-n-Place				GlassPot Lift				Clock Button Press				Clock Hand Turning				Total
	RP	RB	+RO	+ Light	RP	RB	+RO	+ Light	RP	RB	+RO	+ Light	RP	RB	+RO	+ Light	
RGB	**0.30**	0.00	0.00	0.00	**0.10**	0.00	0.00	0.00	**0.25**	0.00	0.00	0.00	**0.05**	0.00	0.00	0.00	0.04
Depth	0.25	**0.20**	0.00	0.00	0.05	0.00	0.00	0.00	0.10	**0.10**	0.00	0.00	0.00	0.00	0.00	0.00	0.04
Pointcloud	0.10	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.02
ATK	**0.85**	**0.80**	**0.55**	**0.45**	**0.75**	**0.65**	**0.60**	**0.60**	**0.90**	**0.90**	**0.80**	**0.75**	**0.50**	**0.50**	**0.40**	**0.35**	**0.64**

Table 1: **Real-world** Policy Success Rates. Varying conditions including RP (random pose), RB (background), RO (distractor object), Light. ATK consistently outperforms baseline methods using alternative modalities in sim-to-real transfer.

Q3: ATK **selects keypoints that capture sufficient task information** As shown in Table 2, ATK could outperform RGB, depth and point cloud-based policies in terms of success rate across all tasks in simulation. RGB-based policies perform well under standard conditions but degrades significantly when distractions such as novel objects, different backgrounds, or lighting conditions are introduced. Depth- and pointcloud- based policies struggle with handling distractors and varying lighting conditions, as the testing scene falls outside the training distribution. Moreover, they lack robustness in fine-grained manipulations, such as the clock tasks, where small parts are difficult to capture accurately using these modalities. In contrast, ATK maintain high robustness and generalization across all tested conditions.

	Sushi				Glass			
	RP	RB	+RO	+ Light	RP	RB	+RO	+ Light
RGB	**0.453±0.262**	0.076±0.041	**0.027±0.020**	**0.010±0.014**	**0.253±0.154**	0.109±0.098	**0.020±0.021**	0.000±0.000
Depth	0.255±0.199	0.255±0.199	0.020±0.021	**0.010±0.014**	0.110±0.001	**0.110±0.001**	0.000±0.000	0.000±0.000
Pointcloud	0.277±0.088	**0.277±0.088**	0.020±0.021	0.000±0.000	0.033±0.047	0.033±0.047	0.000±0.000	0.000±0.000
ATK	**0.893±0.073**	**0.893±0.073**	**0.893±0.073**	**0.893±0.073**	**0.933±0.034**	**0.933±0.034**	**0.933±0.034**	**0.933±0.034**
	Clock button				Clock turning			
	RP	RB	+RO	+ Light	RP	RB	+RO	+ Light
RGB	**0.456±0.293**	0.046±0.017	**0.013±0.019**	0.000±0.000	**0.367±0.205**	0.093±0.020	**0.013±0.012**	0.000±0.000
Depth	0.290±0.150	**0.290±0.150**	0.000±0.000	0.000±0.000	0.256±0.264	**0.256±0.264**	0.000±0.000	**0.020±0.021**
Pointcloud	0.107±0.056	0.107±0.056	0.010±0.014	0.000±0.000	0.077±0.056	0.077±0.056	0.010±0.014	0.010±0.014
ATK	**0.970±0.024**	**0.970±0.024**	**0.970±0.024**	**0.970±0.024**	**0.903±0.028**	**0.903±0.028**	**0.903±0.028**	**0.903±0.028**

Table 2: **Simulator** Policy Success Rates using *different input modalities* over 3 random seeds. Keypoint-based policies are easier to distill in simulator than other baselines with alternative sensor modalities.

Q4: ATK **Chooses Interpretable and Task-Relevant Keypoints for Multifunctional Tasks.** In multifunctional tasks, such as the clock manipulation, ATK selects keypoints that focus on task-relevant parts as shown in (Fig.2), selecting the clock hand for turning or the clock frame for button pressing. In the Kitchen setting, ATK successfully selected relevant keypoints for microwave closing (keypoints focusing on the microwave's control panel) and drawer closing (keypoints focusing on the handle). These results highlight the interpretable advantage of keypoint-based method and demonstrate the effectiveness of our keypoint selection method in ensuring task relevance in diverse environments.

4 Conclusion and Limitations

We present ATK, a system for automatically selecting task-relevant keypoints in simulation, learning keypoint-based policies, and transferring them to the real world. While promising, the system faces challenges in tracking and optimization. Policies using 2D keypoints are sensitive to camera perspective changes, and off-the-shelf trackers may lack robustness for robotic tasks. Additionally, the method is sensitive to hyperparameters due to the non-smooth nature of the optimization problem, making tuning difficult. Despite these issues, our work demonstrates the robustness of keypoint-based policies and provides an effective approach for automatic keypoint selection, with room for developing more automated and robust techniques to address these challenges.

Acknowledgments

This work was (partially) funded by grants from the National Science Foundation NRI (#2132848), DARPA RACER (#HR0011-21-C-0171), the Office of Naval Research (#N00014-24-S-B001 and #2022-016-01 UW), and funding from the Toyota Research Institute through the University Research Program 3.0. We gratefully acknowledge gifts from Amazon, Collaborative Robotics, Cruise, and others.

References

[1] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In N. M. Amato, S. S. Srinivasa, N. Ayanian, and S. Kuindersma, editors, *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017. doi:10.15607/RSS.2017.XIII.058. URL http://www.roboticsproceedings.org/rss13/p58.html.

[2] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84):eadc9244, 2023. doi:10.1126/scirobotics.adc9244. URL https://www.science.org/doi/abs/10.1126/scirobotics.adc9244.

[3] M. Torne, A. Simeonov, Z. Li, A. Chan, T. Chen, A. Gupta, and P. Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *CoRR*, abs/2403.03949, 2024. doi:10.48550/ARXIV.2403.03949. URL https://doi.org/10.48550/arXiv.2403.03949.

[4] R. Julian, E. Heiden, Z. He, H. Zhang, S. Schaal, J. J. Lim, G. S. Sukhatme, and K. Hausman. Scaling simulation-to-real transfer by learning a latent space of robot skills. *Int. J. Robotics Res.*, 39(10-11), 2020. doi:10.1177/0278364920944474. URL https://doi.org/10.1177/0278364920944474.

[5] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[6] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. V. Wyk, A. Zhurkevich, B. Sundaralingam, and Y. S. Narang. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. In *IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023*, pages 5977–5984. IEEE, 2023. doi:10.1109/ICRA48891.2023.10160216. URL https://doi.org/10.1109/ICRA48891.2023.10160216.

[7] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang. Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. In *Conference on Robot Learning*, pages 594–605. PMLR, 2023.

[8] C. Doersch, Y. Yang, M. Vecerík, D. Gokay, A. Gupta, Y. Aytar, J. Carreira, and A. Zisserman. TAPIR: tracking any point with per-frame initialization and temporal refinement. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 10027–10038. IEEE, 2023. doi:10.1109/ICCV51070.2023.00923. URL https://doi.org/10.1109/ICCV51070.2023.00923.

[9] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht. Cotracker: It is better to track together. *CoRR*, abs/2307.07635, 2023. doi:10.48550/ARXIV.2307.07635. URL https://doi.org/10.48550/arXiv.2307.07635.

[10] L. Tang, M. Jia, Q. Wang, C. P. Phoo, and B. Hariharan. Emergent correspondence from image diffusion. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine,

editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/0503f5dce343a1d06d16ba103dd52db1-Abstract-Conference.html.

[11] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng. Track anything: Segment anything meets videos. *CoRR*, abs/2304.11968, 2023. doi:10.48550/ARXIV.2304.11968. URL https://doi.org/10.48550/arXiv.2304.11968.

[12] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[13] Y. Ju, K. Hu, G. Zhang, G. Zhang, M. Jiang, and H. Xu. Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation. *arXiv preprint arXiv:2401.07487*, 2024.

[14] M. Laskin, A. Srinivas, and P. Abbeel. CURL: contrastive unsupervised representations for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5639–5650. PMLR, 2020. URL http://proceedings.mlr.press/v119/laskin20a.html.

[15] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. VIP: towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=YJ7o2wetJ2.

[16] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3M: A universal visual representation for robot manipulation. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pages 892–909. PMLR, 2022. URL https://proceedings.mlr.press/v205/nair23a.html.

[17] A. Zhang, R. T. McAllister, R. Calandra, Y. Gal, and S. Levine. Learning invariant representations for reinforcement learning without reconstruction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.

[18] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 1134–1141. IEEE, 2018. doi:10.1109/ICRA.2018.8462891. URL https://doi.org/10.1109/ICRA.2018.8462891.

[19] N. Morrical, J. Tremblay, Y. Lin, S. Tyree, S. Birchfield, V. Pascucci, and I. Wald. Nvisii: A scriptable tool for photorealistic image generation, 2021.

[20] 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 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 23–30. IEEE, 2017. doi:10.1109/IROS.2017.8202133. URL https://doi.org/10.1109/IROS.2017.8202133.

[21] A. Yu, A. Foote, R. Mooney, and R. Martín-Martín. Natural language can help bridge the sim2real gap. *CoRR*, abs/2405.10020, 2024. doi:10.48550/ARXIV.2405.10020. URL https://doi.org/10.48550/arXiv.2405.10020.

[22] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12627–12637. Computer Vision Foundation / IEEE, 2019. doi:10.1109/CVPR.2019.01291. URL http://openaccess.thecvf.com/content_CVPR_2019/html/James_Sim-To-Real_via_Sim-To-Sim_Data-Efficient_Robotic_Grasping_via_Randomized-To-Canonical_Adaptation_Networks_CVPR_2019_paper.html.

[23] D. Ho, K. Rao, Z. Xu, E. Jang, M. Khansari, and Y. Bai. Retinagan: An object-aware approach to sim-to-real transfer. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pages 10920–10926. IEEE, 2021. doi:10.1109/ICRA48506.2021.9561157. URL https://doi.org/10.1109/ICRA48506.2021.9561157.

[24] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning. *CoRR*, abs/2401.00025, 2024. doi:10.48550/ARXIV.2401.00025. URL https://doi.org/10.48550/arXiv.2401.00025.

[25] M. Vecerík, C. Doersch, Y. Yang, T. Davchev, Y. Aytar, G. Zhou, R. Hadsell, L. Agapito, and J. Scholz. Robotap: Tracking arbitrary points for few-shot visual imitation. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 5397–5403. IEEE, 2024. doi:10.1109/ICRA57147.2024.10611409. URL https://doi.org/10.1109/ICRA57147.2024.10611409.

[26] L. Manuelli, W. Gao, P. R. Florence, and R. Tedrake. KPAM: keypoint affordances for category-level robotic manipulation. In T. Asfour, E. Yoshida, J. Park, H. Christensen, and O. Khatib, editors, *Robotics Research - The 19th International Symposium ISRR 2019, Hanoi, Vietnam, October 6-10, 2019*, volume 20 of *Springer Proceedings in Advanced Robotics*, pages 132–157. Springer, 2019. doi:10.1007/978-3-030-95459-8_9. URL https://doi.org/10.1007/978-3-030-95459-8_9.

[27] L. Manuelli, Y. Li, P. R. Florence, and R. Tedrake. Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning. In J. Kober, F. Ramos, and C. J. Tomlin, editors, *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, volume 155 of *Proceedings of Machine Learning Research*, pages 693–710. PMLR, 2020. URL https://proceedings.mlr.press/v155/manuelli21a.html.

[28] X. Ma, D. Hsu, and W. S. Lee. Learning latent graph dynamics for visual manipulation of deformable objects. In *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pages 8266–8273. IEEE, 2022. doi:10.1109/ICRA46639.2022.9811597. URL https://doi.org/10.1109/ICRA46639.2022.9811597.

[29] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, J. Ichnowski, E. R. Novoseller, M. Hwang, M. Laskey, J. Gonzalez, and K. Goldberg. Untangling dense non-planar knots by learning manipulation features and recovery policies. In D. A. Shell, M. Toussaint, and M. A. Hsieh, editors, *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021. doi:10.15607/RSS.2021.XVII.013. URL https://doi.org/10.15607/RSS.2021.XVII.013.

[30] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation, 2024.

[31] F. Liu, K. Fang, P. Abbeel, and S. Levine. Moka: Open-vocabulary robotic manipulation through mark-based visual prompting. *arXiv preprint arXiv:2403.03174*, 2024.

[32] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning, 2023.

[33] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

[34] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

[35] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi:10.1109/IROS.2012.6386109.

[36] Y. Zhang, L. Ke, A. Deshpande, A. Gupta, and S. Srinivasa. Cherry-picking with reinforcement learning: Robust dynamic grasping in unstable conditions. *arXiv preprint arXiv:2303.05508*, 2023.

5 Appendix

5.1 Related Work

Visual Representations. Previous research has explored various visual representation learning for robotics [14, 15, 16, 17, 18], using both self-supervised and supervised objectives [18, 14]. These representations often rely on large-scale pretraining or auxiliary information-theoretic objectives [15, 17]. In contrast, this work focuses on visual representation suitable for sim-to-real transfer and leverages privileged simulator information to identify effective visual representations.

Sim-to-real Transfer. Bridging the perceptual gap between simulation and the real world remains a significant challenge due to discrepancies in the observation space. While simulations have become more photorealistic [19], the direct transfer of policies across domains continues to suffer from performance degradation. Prior works have proposed various methods to mitigate this gap, including domain randomization [20], latent representation learning [16, 21], unsupervised image translation [22, 23], depth-based policy [3, 2] and explicit pose estimation [6]. While promising, they still face challenges in handling complex, precise tasks or rely on task-specific scaffolding (estimate the pose of a certain object) or restrictive assumptions (availability of accurate depth sensors). In this work, we emphasize on considering task-driven objectives in designing the visual representations.

Keypoints as Representations for Learning-based control. Keypoints have been utilized as robust state representations for robotic manipulation in several prior works [24, 25, 26, 27]. These techniques have been applied in areas including deformable object manipulation [28, 29], few-shot imitation learning [25], model-based reinforcement learning [27], and learning from videos [24, 30]. However, these approaches often rely on heuristic or manual keypoint selection [31, 32]. Our work differs by focusing specifically on sim-to-real transfer and introducing a task-driven, simulation-guided method for automatic keypoint selection.

5.2 Problem Formulation

We study decision-making in finite-horizon Markov Decision Processes (MDPs) defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \rho_0, \mathcal{R}, \gamma)$, where $\mathcal{S}$ represents the Lagrangian state space (the compact, physical state of the system), $\mathcal{O}$ is the observation space, $\mathcal{A}$ is the action space, $\mathcal{P}(s'|s, a)$ defines the transition dynamics, ρ_0 is the initial state distribution, $\mathcal{R}$ is the reward function, and γ is the discount factor. In simulation, agents have access to the Lagrangian state $\mathcal{S}$, which provides a compact, complete description of the environment (for instance object positions, velocities and so on). However, in the real world, agents can only access sensor observations $\mathcal{O}$ (e.g., RGB images) instead of the true Lagrangian state. Although the real world might be partially observable, we assume that the observation $o \in \mathcal{O}$ is sufficient to make optimal decisions, bypassing the need for an explicit belief state as in Partially Observable MDPs. The observation o is produced by an invertible emission function f, such that $o = f(s)$.

Our goal is to derive a visuomotor policy π^* that maximizes the expected cumulative reward in the real world $\mathbb{E}\left[\sum_t \gamma^t \hat{r}(s_t)\right]$ when acting on observations o_t. In this work, we will derive such policies in simulation using an arbitrary decision making method (which could include imitation learning, reinforcement learning, trajectory optimization, or motion planning), and then transfer this to the real world. The key challenge is the perceptual gap between simulation and the real world. We formalize this with two MDPs: $\mathcal{M}_{\text{sim}} = (\mathcal{O}_{\text{sim}}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \rho_0, \mathcal{R}, \gamma)$ for simulation, and $\mathcal{M}_{\text{real}} = (\mathcal{O}_{\text{real}}, \mathcal{A}, \mathcal{P}, \rho_0, \mathcal{R}, \gamma)$ for the real world. The same underlying state s goes through different emission functions and leads to different observations in simulation, $o_{\text{sim}} = f_{\text{sim}}(s)$, and the real world, $o_{\text{real}} = f_{\text{real}}(s)$. Notably, a simulation agent has access to both observation $o_{\text{sim}} \in \mathcal{O}_{\text{sim}}$ and privileged access to the Lagrangian state $s \in \mathcal{S}$. In the real world, however, the agent relies solely on sensor observations $o_{\text{real}} \in \mathcal{O}_{\text{real}}$, such as RGB camera images. The challenge in transferring end-to-end visuomotor policies $\pi^*(a_t|o_t)$ from simulation to the real world lies in the distribution mismatch between $\mathcal{O}_{\text{sim}}$ and $\mathcal{O}_{\text{real}}$.

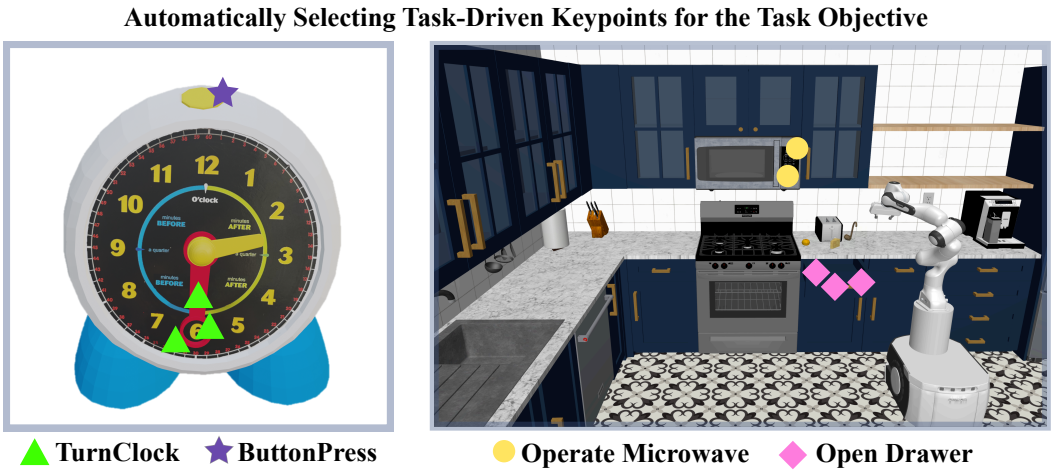

Figure 2: We propose ATK to select task-driven, minimal but necessary keypoints to represent the state such that robotic policies can solve manipulation tasks in sim-to-real transfer. Given a clock and two distinct tasks (Turn the clock hand or press the button), our proposal selects keypoints near the clock hand or on the clock body respectively. In a kitchen environment, it automatically selects keypoints around the microwave's control panel for the task of pressing buttons; around the handle for the task of drawer closing. These task-driven keypoints enable robust sim-to-real policy transfer.

To address this challenge, we aim to select a state representation that retains invariance between simulation and real-world observations: $g_{\text{sim}}(o_{\text{sim}}) = g_{\text{real}}(o_{\text{real}})$. While many such choices are feasible, this work primarily focuses on keypoint-based representations. Our proposed keypoint-based encoder g selectively outputs task-relevant keypoints from observation o_{real} to transfer a policy from sim-to-real domains.

5.3 Details for selecting Task-driven keypoint representation

Concretely, we feed the student policy with many candidate keypoints, enforcing information bottleneck while optimizing the resulting policy's performance. The student keypoint-based policy has access to a large batch of candidate keypoints, $\{k_0^i\}_{i=1}^M$, that were initially obtained via random sampling in the image plane. We then enforce sparsity of keypoints by applying a mask, $\mathbb{M}_\phi$ to zero out the information in some keypoints. Effectively, the student policy operates on the resulting subset of keypoints. The mask is parameterized using a neural network for optimization. Since masking is a discrete sampling operation and non-differentiable, we employ the Gumbel softmax approximation [33] for tractable gradient through categorical reparameterization. Intuitively, this method approximates the discrete categorical distribution with a continuous and differentiable function, ensuring that the forward pass outputs are discrete while the backward pass remains differentiable.

The resulting objective for selecting a minimal set of keypoints while learning a policy is

$$\min -\mathbb{E}_{(k,a)\sim\mathcal{D}}\left[\log \pi_\theta^k(a_t^* \mid \{k_t^i\}_{i=1}^N)\right] + \alpha\|\mathbb{M}_\phi(\{k_t^i\}_{i=1}^N)\|_1$$

where $\log \pi$ considers the performance of the resulting policy and α controls the sparcity penality and ensure the selection is minimal. This training procedure filters out points that are irrelevant to predicting the actions suggested by the optimal state-based policy $\pi^*(a_t|s_t)$, while it also filters out points that are challenging to track using $h_{\mathcal{C}}$ since their representations over time $(\{k_t^i\}_{i=1}^M)$ will be unreliable to optimal actions prediction.

5.4 Experiment Setting

Tasks and Challenges We consider **three fine manipulation tasks for quantitative analysis** as shown in Fig 3: (1) The *sushi pick-and-place task* requires grasping a piece of sushi in a cluttered environment with distracting objects. (2) The *glasspot tip lifting task* requires precise grasp and lift of the tip of a glass pot. This task is particularly challenging due to the pot's reflective surface and the

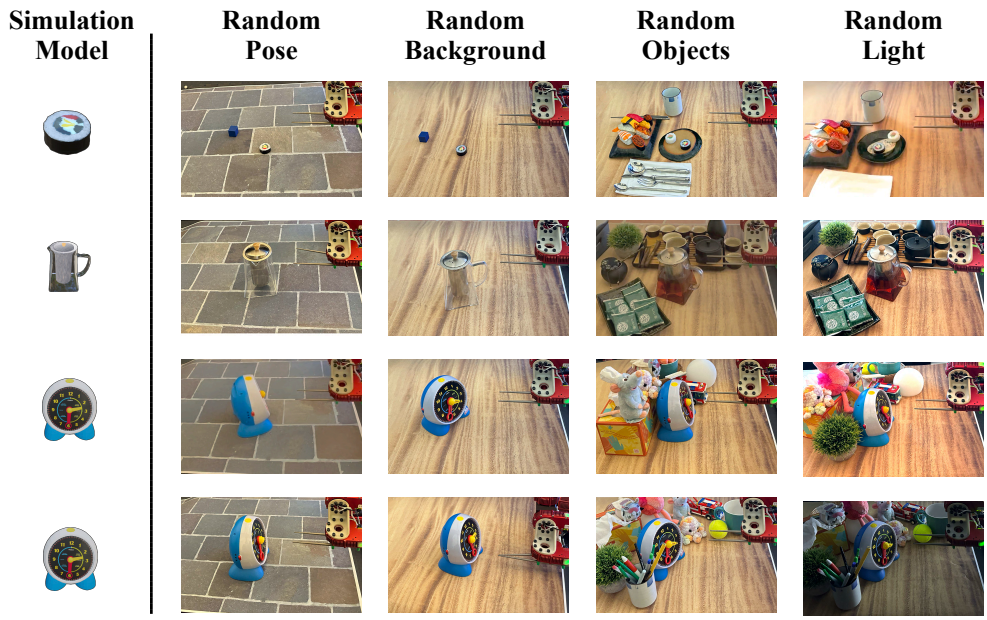

Figure 3: Depiction of various evaluation tasks: From top down - grasping sushi, lifting a glass teapot lid, clock turning, clock button pressing. Depicted are various testing variations - different backgrounds, distractor objects and light conditions.

small size of the tip. (3) The *clock manipulation task* contains two distinct tasks: turning the button at the top of the clock or turning the clock hand on its surface, requiring task-specific representations for manipulation of articulated objects. Each task brings in challenge from tracking difficulty, the precision of manipulation, need of task-specific focus and variations in scene configuration. We also consider **two tasks in a multi-functional kitchen scene** [34] for qualitative study: closing microwave or closing drawer.

Simulation and Real-World Setup We test quantitative tasks in both sim and real. We create MuJoCo [35] simulation using an iPhone app, Scaniverse, to scan and import the meshes of real-world objects and adding joints for articulated objects. We conduct real world robotic experiments using a 6-DOF Hebi robot arm equipped with chopsticks, following [36]. For RGB and depth streaming, we employ Azure Kinect RGB-D cameras.

Baselines We compare our approach with two groups of six baselines. *1) Input modality:* Policies trained with different input types: **RGB images**, **Depth images**, and **Point clouds**. We obtain them using similar distillation process from the same teacher policy but vary the input modality. *2) Keypoint selection methods:* We consider three more baselines: **FullSet** uses all sampled keypoints across the image plane; **Random Select** randomly selects the same number of keypoints as our method; and **GPTSelect** uses GPT-4 to select the same number of keypoints based on the image and task, please visit the website for details.

Evaluation We evaluate each agent on 100 trajectories across 3 seeds in simulation and 20 trajectories in the real world, all with varying initial configurations. To assess robustness and generalization, we introduce disturbances: **RP** (random object poses), **RB** (background texture shuffling), **RO** (random distractor objects), and **Light** (altered lighting). We replicate these disturbances in both sim and real.

5.5 Additional experiments results

Q1: Accurate Sim-to-real Keypoints Transfer We consider two metrics: *Confidence Score* and *Mean Distance Error* to measure the correspondence between real-world keypoints and their simulated version. The prior measure the average cosine similarity between the points' underlying features. The Mean Distance Error measures the average pixel distance between manually-labeled correspondence points and those identified by the correspondence function $h_{\mathcal{C}}$. As shown in Table 3, our method achieves a high confidence score ($0.76 - 0.82$), and low distance error (6 pixel coordinates, $< 5\%$ of the object size), demonstrating accurate sim-to-real keypoints transfer.

Method	Sushi	Glass	Clock button	Clock turning
Confident Score	0.78	0.76	0.80	0.82
Mean Distance Error	3.24	5.21	2.79	6.75

Table 3: Quantitative metrics measurement for the keypoints transfer between simulation and real world.

Q2: ATK Selects a Robust Subset of Keypoints Compared to Baselines. In Table 4, our method demonstrates notable robustness compared to FullSet, RandomSelect and GPTSelect baselines. The FullSet baseline preserves most information but suffers from unreliable tracking. The RandomSelect often captures irrelevant information, causing the policy to lose critical information when the object's pose changes and lowering the success rates. GPT-based selection performs well in some tasks but occasionally suffers from spatial and language mismatches, leading to incorrect selections.

	Sushi				Glass			
	RP	RB	+RO	+ Light	RP	RB	+RO	+ Light
FullSet	0.122±0.057	0.053±0.036	0.010±0.008	0.013±0.012	**0.311±0.150**	**0.069±0.056**	0.013±0.012	**0.013±0.012**
RandomSelect	**0.337±0.315**	**0.246±0.360**	**0.233±0.370**	**0.226±0.375**	0.120±0.082	0.031±0.044	**0.116±0.151**	0.006±0.009
GPTSelect	0.032±0.009	0.020±0.008	0.013±0.005	0.006± 0.004	0.133±0.188	0.020±0.028	0.010±0.014	0.010±0.014
ATK	**0.893±0.073**	**0.893±0.073**	**0.893±0.073**	**0.893±0.073**	**0.933±0.034**	**0.933±0.034**	**0.933±0.034**	**0.933±0.034**
	Clock button				Clock turning			
	RP	RB	+RO	+ Light	RP	RB	+RO	+ Light
FullSet	0.474±0.317	0.126±0.090	0.026±0.030	0.020±0.016	0.253±0.183	0.083±0.880	0.010±0.014	0.010±0.014
RandomSelect	0.107±0.030	0.080±0.045	0.036±0.032	0.026±0.020	**0.253±0.166**	0.076±0.088	0.000±0.000	0.000±0.000
GPTSelect	**0.913±0.041**	**0.913±0.041**	**0.913±0.041**	**0.913±0.041**	0.065±0.053	**0.146±0.179**	**0.077±0.088**	**0.020±0.028**
ATK	**0.970±0.024**	**0.970±0.024**	**0.970±0.024**	**0.970±0.024**	**0.903±0.028**	**0.903±0.028**	**0.903±0.028**	**0.903±0.028**

Table 4: **Simulator** Policy Success rate using **different keypoint selection methods** over 3 random seeds. ATK consistently outperforms alternative keypoint selection methods using random sampling or ChatGPT selection.

Overall, our method consistently selects task-relevant and trackable keypoints, outperforming baselines by 0.72 points.