Anticipating Unintended Robot Behaviors with Side Effect Critics

Ryan Lindeborg[*1], Richard Jiankun Peng[*1], Ayush Jain[2], Jesse Zhang[3], Abrar Anwar[1], Jesse Thomason[1]
[*]Equal contribution [1]University of Southern California [2]Meta [3]University of Washington

***Abstract*—A robot can complete its task while still causing *side effects*: unintended changes to the environment that are unrelated to the goal. For example, a robot may successfully open a drawer while knocking over nearby objects, undoing previously completed subtasks, or altering the environment in ways that incur downstream failures. This makes side effects difficult to address with conventional failure or anomaly detection: they often occur during otherwise successful execution, and by the time a failure signal appears, the unintended change may already have occurred. In this work, we study *side effect prediction for robotic manipulation*, aiming to anticipate unintended changes before they occur, so the robot can stop, resample, or defer to a human. We first introduce a taxonomy that organizes side effects in manipulation settings. We then present a simple solution towards capturing side effects: training a side effect critic, initialized from a pre-trained vision-language action (VLA) model, that predicts whether given action chunks will cause a side effect *before* the actions are executed. Across simulation and real-world experiments, our VLA-initialized side effect critic outperforms vision-language models, both zero-shot and finetuned, as well as a representative runtime failure-detection baseline. We further find that robot data pretraining in the critic's backbone improves side effect prediction, indicating that understanding actions is central to the task.**

I. Introduction and Related Work

As robots are deployed in increasingly open-ended environments, humans expect them not only to complete their assigned tasks but also to avoid unintended consequences of their actions. However, conventional task-success metrics in robotics are agnostic to such unintended consequences: a trajectory can be labeled successful while still producing undesirable changes elsewhere in the scene. For example, a warehouse robot may correctly assemble a new pallet while accidentally removing an item from a previously completed order. Similarly, a household robot may water plants while knocking over a nearby vase. We refer to such unintended, task-irrelevant consequences as *side effects*. This raises the central question we study: can a robot predict side effects before they happen, so it can prevent them while still completing the task?

Side effects have long been studied in the AI safety literature. Amodei et al. [2] frame side effects as arising from objective functions that are indifferent to parts of the environment outside the task, and propose impact regularizers as a mitigation. Subsequent work has developed measures of side effects based on deviations from counterfactual trajectories [12], reversible state transitions [9], attainable utility preservation [20], and planning frameworks that explicitly reason about non-Markovian side effects [19]. Much of this literature studies side effects in simple, discrete environments and reinforcement learning settings [21, 14]. In contrast, we study side effects in real-world robotic manipulation, where agents must reason over high-dimensional observations, continuous actions, and more complex physical interactions with the environment.

In robotics settings, similar work studies *safety / task failure* in robot deployment, which does not directly address *side effects*. Runtime monitoring methods such as FAIL-Detect [22], FIPER [18], and Sentinel [1] focus on identifying task failures, out-of-distribution behavior, or anomalous policy execution. Human-in-the-loop systems further combine failure detection with selective human feedback to improve deployment safety [3, 15]. These approaches are complementary to our work, but they are primarily designed to determine when a robot is close to failing a task rather than to predict whether a sequence of actions may produce an unintended consequence downstream.

There are two key challenges in side effect prediction. First, side effects must be predicted before execution: by the time a robot has knocked over an object, undone a previous subtask, or manipulated the wrong object, the unintended change has already occurred. Second, side effects depend on the physical consequences of future actions, not only on the current scene. Recent vision-language models (VLMs) are therefore a natural starting point, due to their rich visual priors, and are commonly used in robotic failure detection model backbones [8]. However, VLMs are primarily trained on image-text data, so they may not capture how robot actions transform the world. Vision-language-action (VLA) models, in contrast, are pretrained on *robot trajectories* and learn action representations from robot experience. Our key insight is to use such a VLA backbone, which is better-trained for predicting robot action outcomes, as the basis for a side effect critic, allowing the model to score proposed action chunks before execution. Across simulation and real-world experiments, we find that stronger robot-data pretraining results in more effective side effect prediction.

In summary, we make the following contributions:

- We introduce side effect prediction for robotic manipulation, and present a taxonomy of side effects.
- We propose the side effect critic, which scores candidate action chunks to predict side effects before execution, built on a robot-pretrained VLA backbone.
- We show that the side effect critic outperforms state-of-the-art VLMs and a runtime failure-detection baseline in both simulation and the real world, and that robot data pretraining of the backbone improves prediction.

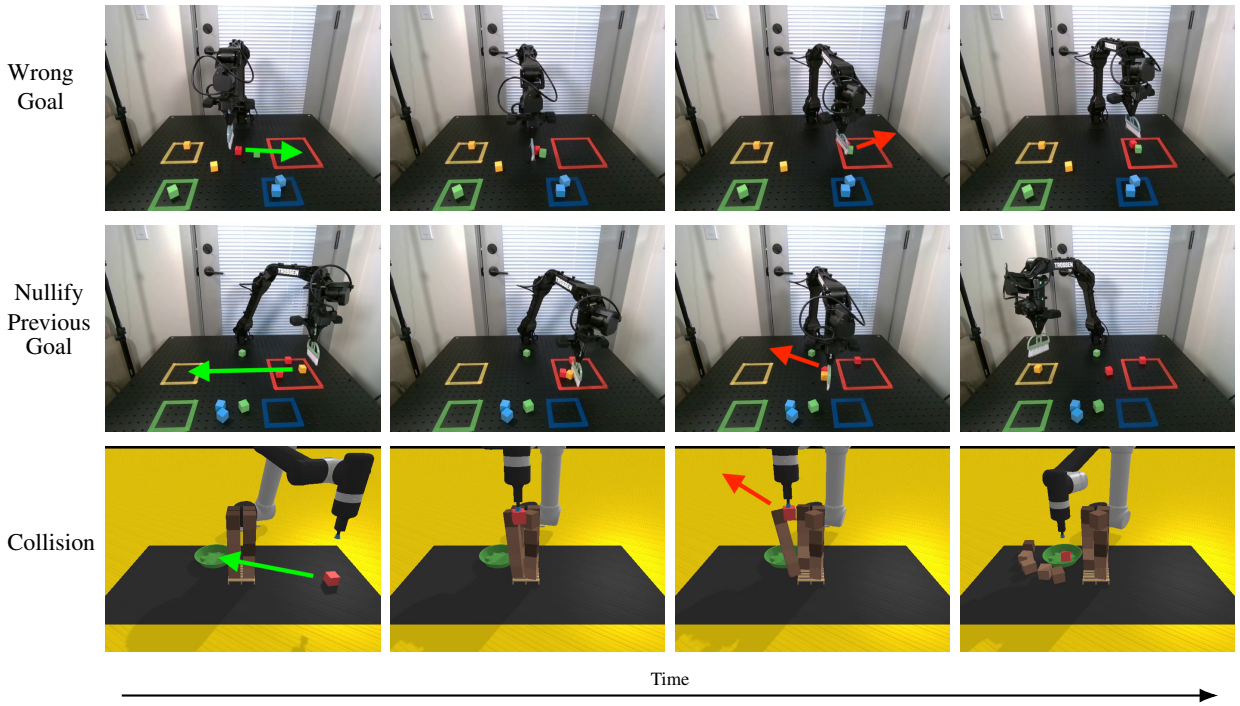

Fig. 1: Side effects the critic must anticipate. In each example, the robot successfully completes the commanded task but also produces an unintended change misaligned to the goal. The **green arrow** traces the action that achieves the commanded task instruction, while the **red arrow** traces the side effect that occurs alongside it. In *Wrong Goal*, the robot correctly places the red block in the target region but also moves the green block into the wrong region. In *Nullify Previous Goal*, the robot achieves its current objective but undoes a previously completed subtask by removing the red block from its correct region. In *Collision*, the robot reaches the desired goal but topples an unrelated stack of boxes. These side effects occur on otherwise successful trajectories, and motivates the need to predict unintended consequences before actions are executed. In this work, we introduce a side effect critic to prempt such side effects in vision-language-action models.

II. Side Effects in Robot Manipulation

We define a side effect as a change to the scene, caused by the robot agent, that is unnecessary or unrelated to the goal task. Crucially, side effects typically occur on successful trajectories, in contrast to classical safety formulations where an incident prevents task completion [9].

A. Taxonomy of Side Effects

We introduce a taxonomy of side effects in robotic manipulation. Although side effects are inherently task-dependent, our goal is to list broad categories that capture the side effects commonly encountered in manipulation.

1) Wrong goal: While executing a task, a robot may achieve a goal that is different from the language instruction it was given. For example, when sweeping red blocks into a marked red target region, it may end up sweeping both red and green blocks into this red target region (Figure 1, top).

2) Nullify a previous goal: For longer-horizon tasks, multiple subtasks are required to achieve a goal. While working on subsequent subtasks, the robot may end up canceling out the initial subtasks' success, especially in the absence of memory, where the robot's only context may be the current subtask language instruction. Consider the high-level goal of sorting a set of differently colored blocks into the corresponding target-colored region. After successfully sorting the red blocks in the red region, while the robot attempts to sort the yellow blocks, it may accidentally remove one of the sorted red blocks from its target region (Figure 1, middle).

3) Reward hacking or misaligned behavior: A robot may achieve the specified objective in a way that is misaligned with human expectations. This is especially relevant for policies trained with reinforcement learning, where the robot may exploit loopholes in a proxy reward instead of satisfying the intended task. Examples include a game-playing agent that loops to repeatedly collect rewards instead of finishing a race [7, 13], or a robot that flips a Lego block upside down to maximize a height-based lifting reward [17].

4) Disturbance of or damage to the scene/environment: A robot may disturb parts of the scene that are irrelevant to the goal task, or may damage scene items. For example, while sweeping red blocks to a target red region, the robot may displace nearby distractor objects or push a block through a liquid spill on the way to the target.

5) Collision: The robot may collide with objects in the environment as it is executing a task. While performing a standard pick-and-place task, it might knock over a stack of boxes that is in its arm trajectory path (Figure 1, bottom).

III. Side Effect Critic

To address the problem of predicting side effects, we introduce a side effect critic to predict side effects before they occur. Given a candidate action chunk from a policy, the

	Real World			Sim		
Method	**Recall**	**Bal. Acc.**	**MCC**	**Recall**	**Bal. Acc.**	**MCC**
Qwen3-VL Finetune	0.452	0.706	0.496	0.701	0.823	0.658
GPT-5.5 Zero-Shot	0.671	0.598	0.147	0.706	0.623	0.188
FAIL-Detect-logpZO	0.274	0.444	-0.087	0.152	0.476	-0.041
FAIL-Detect-RND	0.699	0.566	0.101	0.706	0.532	0.051
Side Effect Critic (Ours)	**0.863**	**0.921**	**0.854**	**0.802**	**0.872**	**0.727**

TABLE I: The side effect critic outperforms both VLM and failure-detection baselines. Side effect prediction performance on real-world and simulation episodes, reported as recall, balanced accuracy, and Matthews correlation coefficient (MCC). Our side effect critic achieves the best performance across all metrics and environments, improving MCC by 72% over the strongest VLM baseline in the real world and 10% in simulation. While GPT-5.5 and finetuned Qwen3-VL demonstrate moderate recall, they exhibit substantially lower balanced accuracy and MCC, indicating difficulty distinguishing side effect-causing actions from benign actions. Failure-detection methods perform near random chance, suggesting that failure-detection signals based on anomaly detection do not transfer effectively to side effect prediction.

side effect critic determines whether executing the actions will also cause unintended, task-irrelevant changes to the scene. We train the side effect critic on annotated robot trajectories, using supervision distinct from task success or failure.

Concretely, at timestep t, the robot observes scene images I_t, robot state s_t, and a language instruction g. Given a candidate action chunk $A_t = (a_t, \ldots, a_{t+H})$, where $H = 50$ in our implementation, the goal is to predict whether each action in the chunk will cause a side effect.

Let $y_t = (y_t^{(1)}, \ldots, y_t^{(H)})$ denote binary side effect labels, where $y_t^{(i)} = 1$ if action a_{t+i-1} causes a side effect and 0 otherwise. The critic is a function $f_\theta(I_t, s_t, g, A_t) \to \hat{y}_t$ that outputs a scalar side effect score for every action in the chunk. At inference time, predictions are thresholded at 0.5 to determine whether an action is likely to cause a side effect.

Our side effect critic is built on top of the $\pi_{0.5}$ VLA model [11]. The critic reuses the $\pi_{0.5}$ backbone, which is pre-trained on a large robotics dataset and builds on the PaliGemma VLM [5] and SigLIP text encoder [24]. Unlike the original $\pi_{0.5}$ policy, which predicts actions conditioned on observations and language, our critic additionally conditions on a candidate action chunk and predicts side effect scores. The action chunk is projected into the model prefix through a newly initialized linear layer, allowing action tokens to jointly attend to image, language, and proprioceptive tokens.

The critic is trained with a flow matching objective to keep in line with the $\pi_{0.5}$ base model. A newly initialized input projection maps a noised prediction vector, of length equal to the action chunk size, into the model, and an output projection predicts the flow matching velocity, which is integrated over 10 steps to yield the final per-action predictions. The side effect critic is roughly the same size as $\pi_{0.5}$, adding only about 4,000 parameters, for a total of 3.35 billion parameters. The critic evaluates a single length 50 action chunk in ~ 180 milliseconds on an NVIDIA A100 GPU.

In preliminary testing, we also trained a simpler variant that replaces the flow-matching head with a binary classifier trained using binary cross-entropy over the per-action predictions. Both formulations achieve comparable performance, suggesting that the gains arise primarily from the pretrained VLA backbone rather than the specific prediction head. To remain consistent with the base $\pi_{0.5}$ model, we use the flow-matching variant.

IV. Experiments

We validate the key design choices underlying the side effect critic, both in the real world and in simulation: (1) To validate its preventive formulation, we compare against runtime failure-detection methods, (2) To test action conditioning, we compare against zero-shot and fine-tuned VLMs, which reason about the scene but do not natively model candidate robot action chunks, (3) We estimate the effect of robot-data pretraining on side effect prediction.

A. Data Collection

We collect a real-world dataset on the 7-degree-of-freedom (DoF) Trossen WidowX AI arm, with 6 arm joints plus gripper control. Low-level actions are executed as absolute joint poses at 30Hz, and teleoperation data is collected using a standard single-arm leader-follower setup.

We also collect a simulation dataset in PyBullet using tasks extended from Ravens [23] with a Universal Robots UR5 embodiment. To generate simulation data, we orchestrate an automated pipeline that uses VLMs to propose side effects, generate candidate trajectories, and collect robot episodes. Episodes are labeled at every timestep according to whether the action at that timestep causes a side effect. All ground-truth side effect labels are provided by a human annotator.

In total, the real-world dataset contains 351 episodes (140,500 timesteps) across 2 tasks, while the simulation dataset contains 500 episodes (499,742 timesteps) across 3 tasks. Models are trained on these datasets and evaluated on 50 held-out episodes in each environment. The real-world dataset contains three side effect categories (Wrong Goal, Nullify Previous Goal, and Disturbance to Environment), while the simulation dataset additionally contains Collision side effects.

B. Evaluation Metrics

Side effect-causing actions are substantially less frequent than side effect-free actions, making the evaluation class-imbalanced. We therefore report recall, balanced accuracy, and Matthews correlation coefficient (MCC). Recall measures how

	Real World			Sim		
Method	**Recall**	**Bal. Acc.**	**MCC**	**Recall**	**Bal. Acc.**	**MCC**
No Robot Pretraining	0.822	0.899	0.819	0.733	0.827	0.633
π_0 Init	0.835	0.904	0.819	**0.802**	0.868	0.710
Side Effect Critic (Ours)	**0.863**	**0.921**	**0.854**	**0.802**	**0.872**	**0.727**

TABLE II: Robot data pretraining improves side effect prediction. Comparing backbone initializations of the base PaliGemma model reveals a consistent benefit from robot pretraining. Our side effect critic uses a $\pi_{0.5}$ backbone, and we compare it to a π_0 initialization and a VLM-trained PaliGemma initialization that does not have any robot pre-training. Our critic performs best overall, while the VLM initialization performs worst. These results suggest that predicting side effects benefits from representations learned from robot pre-training.

many true side effects are detected, balanced accuracy accounts for class imbalance by averaging recall across classes, and MCC summarizes all entries of the confusion matrix in a single score. Evaluation is performed at the action-chunk level, which is labeled side effect-causing if any action within the chunk is predicted to cause a side effect.

V. Results

A. Side Effect Critic outperforms runtime failure detection

A natural question is whether existing runtime failure-detection systems can already solve side effect prediction. To test this, we compare against FAIL-Detect [22], a recent failure-detection framework that learns anomaly scores from successful demonstrations and uses conformal prediction to identify problematic behavior at deployment. We evaluate both the logpZO variant, which uses a flow-based density estimate over observations, and the RND variant, which incorporates both observations and actions.

Results are shown in Table I. Neither variant transfers effectively to side effect prediction. In the real world, FAIL-Detect achieves MCC values of -0.087 (logpZO) and 0.101 (RND), while in simulation it achieves -0.041 and 0.051, respectively. These scores are nearly random chance and substantially below our side effect critic's 0.854. This result suggests that side effects are not well captured by the out-of-distribution and uncertainty signals used by failure-detection systems. These anomaly detection-based approaches monitor for deviations away from success; however, side effects often occur on otherwise successful trajectories, which likely caused runtime failure detection approaches to fail.

B. Our VLA-backbone critic outperforms VLMs for side effect prediction

We next investigate whether side effect prediction can be solved using general-purpose vision-language models. We evaluate GPT-5.5 [16] in a zero-shot setting and finetune Qwen3-VL-8B-Instruct [4] using LoRA [10] on the same training data used for our critic. Results are reported in Table I. Our side effect critic, initialized with a *smaller* $\pi_{0.5}$ backbone, substantially outperforms both VLM baselines. In the real world, the critic achieves an MCC of 0.854 compared to 0.147 for GPT-5.5 and 0.496 for finetuned Qwen3-VL. In simulation, our critic reaches 0.727 MCC, compared to 0.188 and 0.658, respectively. Interestingly, several VLM baselines achieve moderate recall but low balanced accuracy and MCC. This result suggests that VLMs frequently over-predict side effects, resulting in many false positives. While VLMs possess strong semantic understanding, we find them less reliable as side effect critics even after finetuning with task-specific data.

C. Stronger robot pretraining improves critic performance

Finally, we find that robot side effect prediction benefits from representations learned through *robot interaction data*. We vary the initialization of our critic's backbone while keeping the architecture, total parameter count, and training procedure fixed. Specifically, we compare our critic initialized from PaliGemma, which contains vision-language pretraining but no large-scale robot experience, π_0 [6], which contains a weaker robot pretraining recipe, and $\pi_{0.5}$, our default backbone with the strongest robot pretraining.

Results are shown in Table II. The $\pi_{0.5}$ initialization performs best overall, achieving MCC values of 0.854 in the real world and 0.727 in simulation. The trend is particularly clear in simulation, where MCC improves with stronger robot pretraining. In the real world, the differences are smaller, though our backbone using $\pi_{0.5}$ remains the most effective backbone. Notably, as shown in Table I, the finetuned Qwen model, whose backbone has no robot pretraining, performs similarly or worse than that of the no robot pretraining PaliGemma model. Taken together, these results suggest that robot experience contributes useful representations for side effect prediction and that understanding how actions alter the environment is an important component of the task.

VI. Conclusion

In this work, we address side effect prediction for robotic manipulation. We study the problem by introducing a taxonomy of side effects and propose the side effect critic, a model that scores a candidate action chunk and predicts whether it will cause a side effect before the actions are executed. Across simulation and real-world experiments, the side effect critic substantially outperforms zero-shot VLMs, fine-tuned VLMs, and runtime failure-detection methods. We find that stronger robot data pretraining in the critic's backbone improves prediction. As robots take on more open-ended tasks, predicting side effects before they occur is an important step toward reliable robotic deployment and systems that act in alignment with human intent.

References

[1] Christopher Agia, Rohan Sinha, Jingyun Yang, Ziang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 689–723. PMLR, 2025. URL https://arxiv.org/pdf/2410.04640.

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete Problems in AI Safety. *arXiv preprint arXiv:1606.06565*, 2016. URL https://arxiv.org/abs/1606.06565.

[3] Yashwanthi Anand, Nnamdi Nwagwu, Kevin Sabbe, Naomi T. Fitter, and Sandhya Saisubramanian. Adaptive Querying for Reward Learning from Human Feedback. *Frontiers in Robotics and AI*, 2026. URL https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2025.1734564/full.

[4] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

[5] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL https://arxiv.org/abs/2407.07726.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. URL https://arxiv.org/abs/2410.24164.

[7] Jack Clark and Dario Amodei. Faulty reward functions in the wild. OpenAI Blog, 2016. URL https://openai.com/index/faulty-reward-functions/.

[8] Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[9] Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no Trace: Learning to Reset for Safe and Autonomous Reinforcement Learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1vuO-bCW.

[10] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[11] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

[12] Victoria Krakovna, Laurent Orseau, Miljan Martic, and Shane Legg. Penalizing Side Effects using Stepwise Relative Reachability. In *Proceedings of the Workshop on Artificial Intelligence Safety 2019 co-located with the 28th International Joint Conference on Artificial Intelligence, AISafety@IJCAI 2019, Macao, China, August 11-12, 2019*, CEUR Workshop Proceedings. CEUR-WS.org, 2019. URL https://ceur-ws.org/Vol-2419/paper_1.pdf.

[13] Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike, and Shane Legg. Specification gaming: the flip side of ai ingenuity. DeepMind Blog, 2020. URL https://deepmind.google/blog/specification-gaming-the-flip-side-of-ai-ingenuity/.

[14] Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A. Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. AI Safety Gridworlds. *arXiv preprint arXiv:1711.09883*, 2017. doi: 10.48550/arXiv.1711.09883. URL https://arxiv.org/abs/1711.09883.

[15] Huihan Liu, Yu Zhang, Vaarij Betala, Evan Zhang, James

Liu, Crystal Ding, and Yuke Zhu. Multi-task interactive robot fleet learning with visual world models. In *8th Annual Conference on Robot Learning (CoRL)*, 2024. URL https://arxiv.org/pdf/2410.22689.

[16] OpenAI. Gpt-5.5 system card. Technical report, OpenAI, April 2026. URL https://deploymentsafety.openai.com/gpt-5-5/gpt-5-5.pdf.

[17] Ivaylo Popov, Nicolas Heess, Timothy Lillicrap, Roland Hafner, Gabriel Barth-Maron, Matej Vecerik, Thomas Lampe, Yuval Tassa, Tom Erez, and Martin Riedmiller. Data-efficient deep reinforcement learning for dexterous manipulation, 2017. URL https://arxiv.org/abs/1704.03073.

[18] Ralf Römer, Adrian Kobras, Luca Worbis, and Angela P. Schoellig. Failure Prediction at Runtime for Generative Robot Policies. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2510.09459.

[19] Aishwarya Srivastava, Sandhya Saisubramanian, Praveen Paruchuri, Akshat Kumar, and Shlomo Zilberstein. Planning and Learning for Non-Markovian Negative Side Effects Using Finite State Controllers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2023. URL https://ojs.aaai.org/index.php/AAAI/article/view/26767.

[20] Alexander Matt Turner, Dylan Hadfield-Menell, and Prasad Tadepalli. Conservative Agency via Attainable Utility Preservation. *arXiv preprint arXiv:1902.09725*, 2019. doi: 10.48550/arXiv.1902.09725. URL https://arxiv.org/abs/1902.09725.

[21] Carroll L. Wainwright and Peter Eckersley. SafeLife 1.0: Exploring Side Effects in Complex Environments. In *Proceedings of the AAAI 2020 Workshop on Artificial Intelligence Safety (SafeAI 2020)*, volume 2560, pages 117–127, 2020. URL https://ceur-ws.org/Vol-2560/paper46.pdf.

[22] Chen Xu, Tony Khuong Nguyen, Emma Dixon, Christopher Rodriguez, Patrick Miller, Robert Lee, Paarth Shah, Rares Ambrus, Haruki Nishimura, and Masha Itkina. Can We Detect Failures Without Failure Data? Uncertainty-Aware Runtime Failure Detection for Imitation Learning Policies. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.073. URL https://www.roboticsproceedings.org/rss21/p073.html.

[23] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Ayzaan Wahid, Vikas Sindhwani, and Johnny Lee. Transporter Networks: Rearranging the Visual World for Robotic Manipulation. In *Proceedings of the Conference on Robot Learning*, 2020. URL https://arxiv.org/abs/2010.14406.

[24] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952, 2023. doi: 10.1109/ICCV51070.2023.01100.