Paired Action-Intervention Audits for Foundation-Model Robot Planning

Anonymous Authors

***Abstract*—Foundation-model robot planners increasingly use learned action-conditioned models, proxy simulators, or success scorers to rank candidate futures before execution. Prediction losses on successful rollouts do not test whether these planner-facing components reject action branches that are locally close to a successful plan and physically incapable of success. We study planner-exploitable false success: an invalid action sequence receives a high success score under the same initial simulator state and goal as a valid rollout, allowing a model-based planner to select it. Starting from successful ManiSkill manipulation rollouts, we construct paired action-intervention contrast sets that perturb contact, timing, support, release, direction, or stability while holding the task seed, serialized state, and goal fixed. Evaluation uses paired false-positive rate at fixed validation-positive retention, adapting FPR@95TPR to simulator-verified invalid action branches, together with invalid-action selection under planner choice. In state-observation ManiSkill tasks over five training seeds, RSSM-lite trained with pairwise ranking reduces $\mathrm{pFPR@95ValTPR}$ from 0.746 to 0.043, drives invalid-action selection from 0.914 to 0.000 on evaluated intervention families, and yields a positive planner score gap of 0.394. Held-out operator analysis exposes a no-release-stack failure, localizing a missing release/support predicate. The results support a planning audit for foundation-model robot stacks: planner-facing success must be tested at the local action boundary, not inferred from rollout MSE.**

I. Introduction

Robot planning in the era of foundation models is increasingly hybrid: large semantic models specify tasks or propose candidate behaviors, while learned action-conditioned models score, simulate, or refine candidate futures for search, policy evaluation, and model-predictive control [9, 8, 6]. Once such a model participates in candidate selection, plausible rollout prediction error is not enough to certify that the planner-facing success scorer assigns low success to action branches that cannot physically succeed. A planner that maximizes the learned score can select an impossible trajectory before deployment.

We call this failure mode *planner-exploitable false success*: a model assigns high success to a local action intervention that violates contact, timing, support, release, direction, or stability under the same initial state and goal as a successful rollout. A planning audit should evaluate the action boundary around a successful plan, since random failures can be distributionally separable from valid rollouts while leaving that local boundary unchanged.

Our paired action-intervention audit for planner-facing manipulation models fixes the task, seed, serialized simulator state, and goal for each contrast unit, then changes only the action sequence through a simulator-verified invalidity operator. The evaluation combines a compact audit protocol, collapse-resistant fixed-retention metrics, and a ManiSkill study showing that rollout MSE and random-failure supervision do not close the local action boundary around successful plans.

II. Planning Contrast Sets

For a finite-horizon manipulation rollout with environment e, initial state s_0, goal g, action sequence $a_{0:H}$, simulator rollout $\tau = \mathrm{Sim}_e(s_0, a_{0:H})$, and binary success predicate $Y(\tau, g)$, a contrast unit is

$$U_i = (e_i, s_{0,i}, g_i, a^+_{i,0:H}, \mathcal{N}_i).$$

The positive branch satisfies $Y(\mathrm{Sim}_{e_i}(s_{0,i}, a^+_{i,0:H}), g_i) = 1$. Each invalid branch in $\mathcal{N}_i$ is generated by an action-intervention operator

$$a^-_{i,k,\lambda,0:H} = T_k(a^+_{i,0:H}; s_{0,i}, g_i, \lambda),$$

where k indexes the intervention family and λ its parameters. The candidate is admitted only when simulator replay gives $Y(\mathrm{Sim}_{e_i}(s_{0,i}, a^-_{i,k,\lambda,0:H}), g_i) = 0$. The planner candidate set is $\mathcal{C}_i = \{a^+_{i,0:H}\} \cup \{a^-_{i,k,\lambda,0:H}\}$.

The audit instantiates these contrast units in ManiSkill2 manipulation tasks [4], using state observations to isolate action-conditioned success scoring from visual recognition. Each unit fixes task, seed, serialized simulator state, and goal. A unit is excluded if the positive branch fails ($\mathcal{B}_{pos}$), the negative branch succeeds ($\mathcal{B}_{neg}$), the initial-state hashes differ ($\mathcal{B}_{reset}$), the goals differ ($\mathcal{B}_{goal}$), or the base unit would cross train/validation/test splits ($\mathcal{B}_{split}$):

$$\mathcal{B} = \mathcal{B}_{pos} \cup \mathcal{B}_{neg} \cup \mathcal{B}_{reset} \cup \mathcal{B}_{goal} \cup \mathcal{B}_{split}.$$

Contrast-unit splits place a successful rollout, its task seed, serialized state, and all paired interventions in a single train, validation, or test split; unpaired random failures are train-only. The predeclared intervention families edit phase-scripted actions at contact, grasp timing, gripper release/support, goal transport, push direction/contact, push magnitude, or post-placement stability, and every negative branch is admitted only after simulator replay failure. The admitted dataset contains 150 successful rollouts, 799 verified paired negatives, and 150 train-only random failures across PickCube, StackCube, and PushCube. One unstable-release candidate was omitted because simulator verification did not produce a failure. Table II reports split and held-out operator counts.

III. Metrics and Training

With model success score $q_\theta(s_0, g, a_{0:H}) \in [0,1]$, valid successful rollouts are positives and paired invalid action interventions are negatives. For a target validation-positive retention β, the threshold $\widehat{\tau}_\beta$ is selected on validation positives as the largest threshold retaining at least a β fraction of valid rollouts:

$$\widehat{\tau}_\beta = \max_{\tau \in [0,1]} \tau \quad \text{s.t.} \quad \frac{1}{|\mathcal{D}^+_{val}|} \sum_{x^+ \in \mathcal{D}^+_{val}} \mathbf{1}[q_\theta(x^+) \geq \tau] \geq \beta.$$

Using this validation threshold, test valid retention and paired false-positive rate are

$$\text{ValidRet@}\beta\text{Val} = \Pr_{x^+ \sim \mathcal{D}^+_{test}} [q_\theta(x^+) \geq \widehat{\tau}_\beta],$$

$$\text{pFPR@}\beta\text{ValTPR} = \Pr_{x^- \sim \mathcal{D}^-_{test}} [q_\theta(x^-) \geq \widehat{\tau}_\beta].$$

The main experiments use $\beta = 0.95$, following the FPR@95TPR convention used in OOD detection [5], with the 95% criterion fixed on validation positives and reported test retention allowed to differ. False positives are paired invalid action branches scored as successful. If $q_\theta(x) = c$ for every input, then $\widehat{\tau}_\beta = c$ and every paired negative is counted, so $\text{pFPR@}\beta\text{ValTPR} = 1$; constant-score collapse is counted as unsafe.

Random-failure supervision can separate unrelated failures while leaving the local action boundary around a successful plan unresolved. If positives and paired negatives share the same representation atom u but random failures map to a separable atom v, a scorer with $q(u) = 1$ and $q(v) = 0$ obtains perfect success-versus-random-failure classification while assigning success to every paired invalid action.

Planner exploitability is evaluated over contrast sets $\mathcal{C}_i$ containing the verified successful action sequence and its paired invalid interventions. The model-based planner selects

$$a_i^\star = \arg\max_{a \in \mathcal{C}_i} q_\theta(s_{0,i}, g_i, a).$$

Ties are broken in favor of the valid branch, which makes invalid selection conservative. We report invalid-action selection rate, simulated success of the selected branch, and paired ranking AUC,

$$\Pr[q_\theta(x^+) > q_\theta(x^-)] + \tfrac{1}{2} \Pr[q_\theta(x^+) = q_\theta(x^-)],$$

computed over verified positive-negative pairs. We also use the planner score gap

$$\Delta_i = q_\theta(s_{0,i}, g_i, a_i^+) - \max_{a^- \in \mathcal{C}_i \setminus \{a_i^+\}} q_\theta(s_{0,i}, g_i, a^-).$$

Nonnegative Δ_i for every contrast unit implies zero invalid-action selection under the specified tie-breaking rule.

Pairwise contrastive training targets the planner-facing ordering that causes invalid branches to be selected by adding a margin ranking loss to the dynamics and success losses. With success logit $z_\theta(x)$,

$$\mathcal{L}_{pair} = \frac{1}{N} \sum_{(x^+, x^-)} \left[m - z_\theta(x^+) + z_\theta(x^-) \right]_+ ,$$

and $\mathcal{L} = \mathcal{L}_{dyn} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{pair}\mathcal{L}_{pair}$.

IV. Empirical Audit

Experiments use lightweight GRU and RSSM-lite state-based sequence world models. RSSM-lite denotes the compact recurrent state-space baseline used here, with a GRU latent transition, sampled latent state, observation/predicate heads, and a success head. Training regimes are success-only, success plus random failures, success plus paired action interventions, and pairwise ranking. Thresholds for pFPR@95ValTPR are selected only on validation positives.

a) Fixed-retention false positives: A raw false-success diagnostic can classify collapsed models as safe when the evaluation threshold is unconstrained. pFPR@95ValTPR counts these cases: GRU random-failure training has raw false-success rate 0.000 but pFPR@95ValTPR 0.943±0.050, because the validation-positive threshold still admits paired invalid branches. Among sequence models, RSSM-lite pairwise ranking attains pFPR@95ValTPR 0.043±0.031 and pair AUC 1.000, compared with RSSM-lite action-intervention pFPR@95ValTPR 0.188±0.060.

b) Planner exploitability: Planner exploitability is measured by allowing the model to select among valid and paired invalid candidate actions. RSSM-lite success-only selects invalid branches in 0.914±0.127 of test contrast sets, giving simulated success 0.086±0.127. Pairwise ranking increases the RSSM-lite planner score gap to 0.394±0.047, reduces invalid-action selection to 0.000, and raises simulated success to 1.000. Action-intervention training leaves residual exploitability, with invalid-action selection 0.133±0.123.

The GRU pairwise row is a negative control: a ranking loss alone does not guarantee low false-success when the sequence model fails to represent the relevant contact, release, or support boundary. The audit measures the trained scorer's local ordering over simulator-verified contrast units.

c) Held-out operators: An operator-held-out stress test removes open-lift, no-release-stack, and wrong-direction-push interventions from training, then evaluates RSSM-lite pairwise ranking on those unseen operators. The raw false-success diagnostic is 0.062 and the paired score gap is 0.757. Transfer depends on the operator family: open-lift ($n = 7$) and wrong-direction-push ($n = 7$) reach false-success rate 0.000, while no-release-stack has false-success rate 1.000 ($n = 7$), identifying release/support as a missing predicate family in the trained scorer.

V. Related Work

Foundation-model robot planners require evaluation of the learned components that shape action selection, including world models, proxy simulators, and success scorers used before execution. MiraBench is the closest concurrent diagnostic benchmark: it frames action-conditioned reliability as a central target for robotic world-model evaluation and studies physics adherence, action-following fidelity, and optimism bias under failure-inducing actions [10]. The audit studied here fixes simulator state and goal, intervenes only on the action sequence

Model	Training	MSE ↓	ValidRet@95Val ↑	pFPR@95ValTPR ↓	Gap ↑	Pair AUC ↑	Invalid Sel. ↓	Sim Success ↑
GRU	success-only	28.875±8.465	0.943±0.035	0.882±0.059	0.000	0.613±0.093	0.695±0.109	0.305±0.109
GRU	random-fail	39.319±19.436	0.914±0.080	0.943±0.050	-0.001±0.001	0.355±0.161	0.790±0.105	0.210±0.105
GRU	action-interv.	9.489±3.004	0.886±0.105	0.827±0.136	-0.052±0.061	0.696±0.079	0.638±0.070	0.362±0.070
GRU	pairwise-rank	9.894±4.106	0.971±0.037	0.954±0.036	-0.060±0.040	0.455±0.153	0.876±0.070	0.124±0.070
RSSM-lite	success-only	2.206±0.401	0.962±0.054	0.746±0.114	-0.001±0.001	0.413±0.145	0.914±0.127	0.086±0.127
RSSM-lite	random-fail	47.680±12.636	0.886±0.048	0.845±0.304	0.000±0.002	0.348±0.288	0.771±0.212	0.229±0.212
RSSM-lite	action-interv.	0.754±0.139	0.943±0.090	0.188±0.060	0.115±0.050	0.959±0.035	0.133±0.123	0.867±0.123
RSSM-lite	pairwise-rank	1.874±0.760	0.971±0.023	0.043±0.031	0.394±0.047	1.000	0.000	1.000

TABLE I

COLLAPSE-RESISTANT FALSE-SUCCESS AND PLANNER-EXPLOITABILITY METRICS. ValidRet@95Val IS MEASURED ON HELD-OUT POSITIVES USING THE THRESHOLD SELECTED TO RETAIN 95% OF VALIDATION POSITIVES. pFPR@95ValTPR USES THE SAME VALIDATION-POSITIVE THRESHOLD AND IS MEASURED ON HELD-OUT PAIRED INVALID BRANCHES. GAP IS THE MEAN PLANNER SCORE GAP BETWEEN THE VALID BRANCH AND THE BEST PAIRED INVALID BRANCH. ROWS REPORT MEAN ± 95% t-INTERVAL OVER FIVE TRAINING SEEDS. METRICS ARE COMPUTED ON THE SAME HELD-OUT CONTRAST-UNIT SPLIT FOR EACH SEED; INTERVALS MEASURE SEED-LEVEL VARIABILITY BECAUSE PAIRED INVALID BRANCHES ARE CLUSTERED BY BASE ROLLOUT.

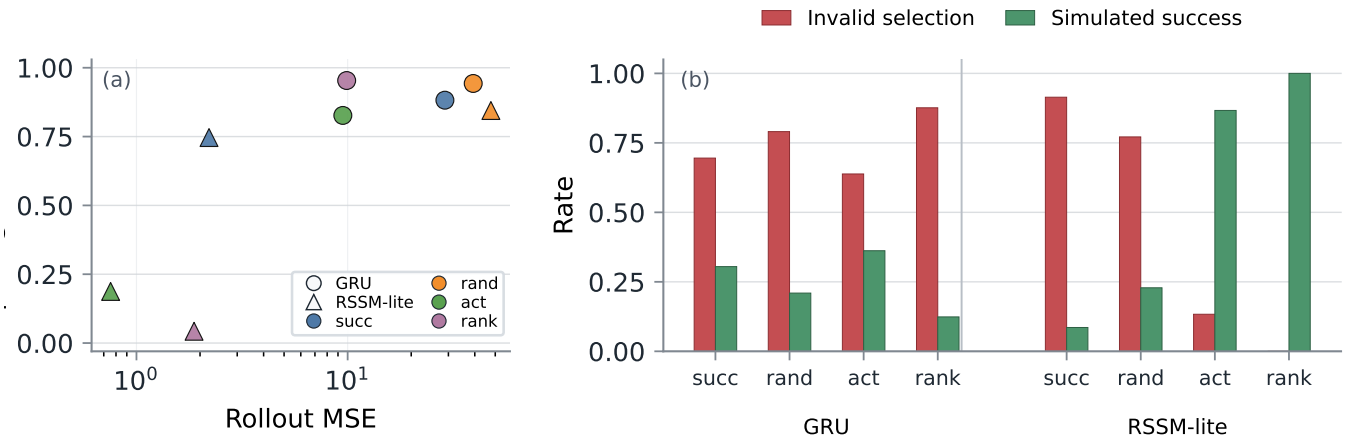

Fig. 1. Rollout-error and planner-selection results. (a) Rollout MSE does not determine paired false-positive rate under fixed validation-positive retention; circles denote GRU, triangles denote RSSM-lite, and colors follow the Table I training order. (b) Model-based selection among valid and simulator-verified invalid candidates. RSSM-lite pairwise ranking drives invalid selection to zero while preserving simulated success.

Item	Count	Note
Tasks	3	PickCube, StackCube, PushCube
Valid bases	150	50 per task
Paired invalid branches	799	simulator-failed
Random failures	150	train-only
Unit splits	105/24/21	train/val/test
Invalid split branches	559/128/112	train/val/test
Excluded candidates	1	unstable-release succeeded
Held-out open-lift	0.000 FSR	$n = 7$
Held-out wrong-direction	0.000 FSR	$n = 7$
Held-out no-release-stack	1.000 FSR	$n = 7$

TABLE II

AUDIT CONSTRUCTION AND HELD-OUT OPERATOR DIAGNOSTIC. SPLITS ARE ASSIGNED AT THE CONTRAST-UNIT LEVEL. FSR IS THE RAW FALSE-SUCCESS RATE UNDER THE HELD-OUT RSSM-LITE PAIRWISE SCORER.

around a successful manipulation plan, admits negatives only after simulator-verified failure, and tests whether a planner-facing success scorer would select the invalid branch.

Action-conditioned robot world models are used as proxy environments for policy evaluation, search, and model-predictive control [9, 8, 6]. WPE evaluates robot policies inside an action-conditioned video world model and reports overestimation for out-of-distribution actions as well as difficulty modeling object interaction. WorldPlanner uses action-conditioned visual world models for MCTS/MPC and addresses hallucinations during planning. Generative world simulators such as Veo-based robot policy evaluation support nominal, OOD, and safety-oriented evaluation [3]. Our audit fixes initial state and goal, intervenes on the action sequence, and evaluates whether the world-model success scorer rejects simulator-verified invalid branches.

Failure-centric robotics datasets and VLMs such as AHA, RoboFAC, and Dream2Fix generate or analyze failed executions for diagnosis, reasoning, or recovery [1, 11, 7]. The contrast sets in this paper use local action interventions to evaluate whether a planner-facing world model scores physically impossible branches as successful. The contrast-set construction follows local decision-boundary evaluation [2], instantiated as simulator-verified action perturbations under fixed state and goal. The fixed-retention false-positive metric follows FPR@95TPR evaluation in OOD detection [5], adapted to valid and invalid manipulation rollouts with thresholds selected

on validation positives. Related concerns also appear in model-based offline RL, where pessimistic objectives such as MOPO penalize uncertain model rollouts [12]; our audit isolates an action-conditioned robotics instance with simulator-verified paired interventions.

VI. Scope and Limitations

The experiment uses simulation and state observations to isolate action-conditioned success scoring from visual recognition, language grounding, and high-level task parsing. This privileged-state setting evaluates the action-success boundary after perception and task interpretation have been removed from the test. The reported claims concern planner-facing action-conditioned scoring under the stated simulator protocol; visual grounding, real-robot deployment, cross-embodiment transfer, and language-conditioned task planning are outside the evaluation. The main paper reports the definitions, admission criteria, metric construction, and aggregate results. Simulator verification supplies the valid/invalid labels; the model ranks recorded candidate sequences, selected branches are scored by their simulator outcomes, and validation thresholds are selected without access to test negatives.

VII. Conclusion

Foundation-model robot planning stacks need success scorers that reject physically impossible action branches under the same state and goal. Paired action-intervention rollouts measure this planning failure by testing local simulator-verified invalid branches around successful plans. Under fixed validation-positive retention, RSSM-lite with pairwise ranking reduces paired false positives and invalid-action selection on evaluated operators, while held-out no-release-stack failures show that coverage of physical predicates is still required. Prediction error alone is not a sufficient audit for planner-facing robot planning models.

References

[1] 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. URL https://openreview.net/forum?id=JVkdSi7Ekg.

[2] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating Models' Local Decision Boundaries via Contrast Sets. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.117. URL https://aclanthology.org/2020.findings-emnlp.117/.

[3] Gemini Robotics Team, Krzysztof Choromanski, Coline Devin, Yilun Du, Debidatta Dwibedi, Ruiqi Gao, Abhishek Jindal, Thomas Kipf, Sean Kirmani, Isabel Leal, Fangchen Liu, Anirudha Majumdar, Andrew Marmon, Carolina Parada, Yulia Rubanova, Dhruv Shah, Vikas Sindhwani, Jie Tan, Fei Xia, Ted Xiao, Sherry Yang, Wenhao Yu, and Allan Zhou. Evaluating Gemini Robotics Policies in a Veo World Simulator. *arXiv preprint arXiv:2512.10675*, 2025. doi: 10.48550/arXiv.2512.10675. URL https://arxiv.org/abs/2512.10675.

[4] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. ManiSkill2: A Unified Benchmark for Generalizable Manipulation Skills. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=b_CQDy9vrD1.

[5] Dan Hendrycks and Kevin Gimpel. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Hkg4TI9xl.

[6] R. Khorrambakht, Joaquim Ortiz-Haro, Joseph Amigo, Omar Mostafa, Daniel Dugas, Franziska Meier, and Ludovic Righetti. WorldPlanner: Monte Carlo Tree Search and MPC with Action-Conditioned Visual World Models. *arXiv preprint arXiv:2511.03077*, 2025. doi: 10.48550/arXiv.2511.03077. URL https://arxiv.org/abs/2511.03077.

[7] Dayou Li, Jiuzhou Lei, Hao Wang, Lulin Liu, Yunhao Yang, Zihan Wang, Bangya Liu, Minghui Zheng, and Zhiwen Fan. Learning Actionable Manipulation Recovery via Counterfactual Failure Synthesis. *arXiv preprint arXiv:2603.13528*, 2026. doi: 10.48550/arXiv.2603.13528. URL https://arxiv.org/abs/2603.13528.

[8] Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. WorldGym: World Model as An Environment for Policy Evaluation. *arXiv preprint arXiv:2506.00613*, 2025. doi: 10.48550/arXiv.2506.00613. URL https://arxiv.org/abs/2506.00613.

[9] Fangyuan Wang, Ziyuan Wang, Guorui Pei, Mengshi Zhang, Canxi Liang, Jun Hu, Zhongxuan Li, Jinsong Wu, Ning Han, Zeqing Zhang, Jiaming Qi, Hongmin Wu, Shiyao Zhang, Pai Zheng, Jia Pan, David Navarro-Alarcon, Sichao Liu, and Peng Zhou. World Models for Robotic Manipulation: A Survey. *arXiv preprint arXiv:2606.00113*, 2026. doi: 10.48550/arXiv.2606.00113. URL https://arxiv.org/abs/2606.00113.

[10] Tianzhuo Yang, Zihan Shen, Zirui Mi, Zhaoyi Zhang, Jiayi Zhou, Jiaming Ji, Juntao Dai, Jiawei Chen, Boyuan Chen, and Yaodong Yang. MiraBench: Evaluating

Action-Conditioned Reliability in Robotic World Models. *arXiv preprint arXiv:2605.29360*, 2026. doi: 10.48550/arXiv.2605.29360. URL https://arxiv.org/abs/2605.29360.

[11] Zewei Ye, Weifeng Lu, Minghao Ye, Tao Lin, Shuo Yang, Junchi Yan, and Bo Zhao. RoboFAC: A Comprehensive Framework for Robotic Failure Analysis and Correction. *arXiv preprint arXiv:2505.12224*, 2025. doi: 10.48550/arXiv.2505.12224. URL https://arxiv.org/abs/2505.12224.

[12] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based Offline Policy Optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 14129–14142. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/a322852ce0df73e204b7e67cbbef0d0a-Abstract.html.