COUNTERFACTUAL DYNAMICS-AWARE EXPERIENCE RETRIEVAL FOR POLICY TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Changes in actuator strength, mass, or friction can make similar states require different controllers. We propose counterfactual dynamics-aware experience retrieval, which compares queries and stored experiences through their responses to shared action sequences. Reference-relative fingerprints capture response direction and magnitude; averaging their distances over each policy's memories selects a controller from a fixed library. Across four DeepMind Control tasks with six policy-training seeds per task, simulator-generated fingerprints reduce mean policy-selection regret by 49–81% relative to random selection and outperform both tested state-similarity baselines in all 24 runs. Same-probe system identification and utility regression remain competitive, with different methods leading on different tasks. With learned visual world models, validation retrieval regret selects models that improve on random selection on Hopper and Walker, whereas held-out rollout loss selects worse-than-random retrievers. These comparisons use separate query draws from the same dynamics grid. The results motivate evaluating world models through the policy choices they support.

1 INTRODUCTION

A controller library is useful only if the agent can choose an appropriate controller. Under a hidden change in dynamics, this choice can be difficult even when the current physical state is known. A walker with weakened actuators may have the same joint angles and velocities as one with nominal actuators, yet require a different gait. A memory retrieved by state proximity can match the pose and miss the property that determines whether its associated policy will work.

Policy reuse makes the selection problem explicit: choose among policies learned in previous environments instead of learning a new policy for every encounter (Rosman et al., 2016). The challenge is to compare the present environment with stored experience in a way that matters for control. Current-state similarity omits the response to action. A dynamics estimate contains that information, but recovering physical parameters is an intermediate objective; the final decision concerns the return of a policy.

We propose **counterfactual dynamics-aware experience retrieval** (Figure 1). A shared bank of action sequences probes a model at each query and memory state. Predicted trajectories and rewards form a response fingerprint. We subtract the response to one reference sequence, compare the direction and magnitude of the resulting contrasts, and average distances over memories associated with each policy. This construction puts the query and stored experiences under the same action conditions. It selects a controller without updating the policy library or fitting a predictor of policy returns.

We evaluate retrieval by the return lost relative to the best estimated policy in the same library. Simulator fingerprints outperform both state-similarity baselines across four tasks and six seeds per task; system identification and utility regression remain competitive. With visual world models, validation retrieval regret selects useful retrievers on Hopper and Walker, while prediction loss selects worse-than-random models.

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

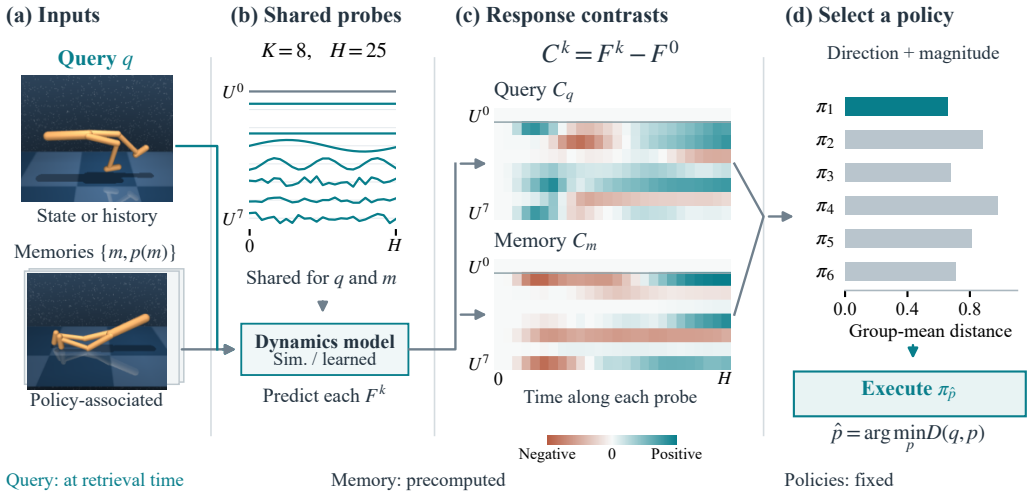


Figure 1: Counterfactual response retrieval: (a) query and memory inputs, (b) shared probes, (c) reference-relative responses, and (d) policy selection by average distance. The Walker example uses recorded data. Traces show one action coordinate; heatmaps show normalized torso-height contrasts on a shared scale. Scores use all features and all 150 memories. Memory responses are precomputed.

Our contributions are:

- A policy-retrieval method that compares reference-relative action responses and selects a controller from a fixed library.
- A four-task, six-seed comparison with state retrieval, system identification, and utility regression, including probe-prefix and aggregation ablations.
- A visual-world-model implementation evaluated under prediction-loss and retrieval-regret selection, with a Walker switching study measuring gains and losses during execution.

2 Related Work

Policy reuse. Bayesian Policy Reuse selects from an existing policy library using observations that inform beliefs about policy performance (Rosman et al., 2016). Our method uses similarity between responses to shared action sequences to select a fixed controller through its associated memories.

Off-dynamics reinforcement learning. Off-dynamics RL studies transfer between environments with mismatched transition dynamics. DARC corrects source-domain rewards using classifiers that distinguish source and target transitions (Eysenbach et al., 2021). ODRL benchmarks transfer across dynamics shifts with online and offline access to source and target domains (Lyu et al., 2024). MOBODY learns target dynamics from source and target datasets and uses synthetic transitions for offline policy optimization (Guo et al., 2026). Our setting shares the dynamics mismatch, but fixes the policy library and evaluates controller selection by regret within that library. This separates retrieval quality from learning a new target-domain policy.

Sim-to-real transfer and online adaptation. Visual domain randomization trains perception models across varied renderings (Tobin et al., 2017), while dynamics randomization varies physical parameters to train transferable controllers (Peng et al., 2018). Universal policies with online system identification estimate dynamics from recent state-action history (Yu et al., 2017). RMA (Kumar et al., 2021) and subsequent locomotion work (Margolis et al., 2024) infer a dynamics embedding from interaction history to condition control. Meta-learned dynamics models can also adapt from recent transitions (Nagabandi et al., 2019). Our history-conditioned model predicts responses for ranking existing controllers. The same-probe system-identification baseline and direct history readouts test alternative uses of dynamics information; the experiments assess transfer within simulation.

Experience retrieval and imagination. Retrieval-Augmented Reinforcement Learning gives an agent direct access to an experience dataset through a learned retrieval process (Goyal et al., 2022). Memoir uses imagined future navigation states to retrieve environmental observations and behavioral histories (Xu et al., 2026). We compare the predicted responses of queries and memories to the same action sequences, then select the policy associated with the closest memory group.

Similarity for transfer. Sreenivasan et al. construct a dynamics-mismatch similarity measure from a bound on action-value differences (Sreenivasan et al., 2023). Tao et al. develop a generalized bisimulation metric between MDPs and analyze policy transfer (Tao et al., 2025). We use a finite-horizon response distance to rank policies and evaluate that ranking through policy-selection regret.

Latent context and world models. PEARL separates task inference from control using probabilistic context variables (Rakelly et al., 2019). Dreamer learns behavior through trajectories imagined in a latent dynamics model (Hafner et al., 2020). Our context conditions predicted responses to a fixed probe bank, and the resulting representation selects among existing policies. This use of a world model makes retrieval quality a downstream criterion distinct from rollout prediction loss.

3 Policy Retrieval under Hidden Dynamics

Consider environments with common state and action spaces and dynamics indexed by a parameter μ. The agent has a fixed library $\Pi = \{\pi_1, \ldots, \pi_P\}$ trained in source regimes. Each memory m stores a source state or observation history and an association $p(m)$ to a policy. The memories associated with policy p form $\mathcal{M}_p$.

A query q describes the present state or recent interaction history. Retrieval chooses an index $\hat{p}(q)$, after which $\pi_{\hat{p}(q)}$ runs in closed loop. The learned retriever receives no target-regime label. In the simulator study, the true query dynamics are available to generate probe responses; in the learned study, a history-conditioned model supplies those responses.

We evaluate selection through the finite-horizon return $J(q, p)$ of policy p from query q. Its estimate $\widehat{J}(q, p)$ is the mean of five stochastic policy replays from the same query state. The **empirical library regret** is

$$\widehat{\mathcal{R}}(q) = \max_{p \in \{1,\ldots,P\}} \widehat{J}(q, p) - \widehat{J}(q, \hat{p}(q)). \tag{1}$$

Every method within a run uses the same query-policy return matrix, holding the candidate policies and evaluation returns fixed. The maximum over finite replay estimates can be optimistic, so regret is measured against the estimated best policy in the library.

4 Counterfactual Response Retrieval

Figure 1 summarizes the retrieval procedure. We denote simulator-generated fingerprints by **CF-state** and learned fingerprints by **CF-WM**.

4.1 Shared action probes

Let $\mathcal{U} = \{U^0, \ldots, U^{K-1}\}$ be a fixed bank of action sequences, with $U^k = (a_0^k, \ldots, a_{H-1}^k)$. Starting from state s, a dynamics model g predicts a trajectory and rewards for each sequence. We concatenate the normalized states and mean reward:

$$F^k(s, g) = \left[\tilde{s}_0^k, \ldots, \tilde{s}_H^k, \frac{1}{H} \sum_{t=0}^{H-1} r_t^k \right]. \tag{2}$$

The reference-relative fingerprint is

$$C(s, g) = \mathrm{vec}\left(\{F^k(s, g) - F^0(s, g)\}_{k=0}^{K-1} \right). \tag{3}$$

Each block records how one action sequence changes the response relative to the reference sequence at the same starting state. The initial-state entries cancel exactly, and the remaining entries describe differences in subsequent motion and reward. Queries and memories use the same ordered bank.

We use eight probes of 25 steps each. Cheetah, Walker, and Hopper use a zero-action reference, sustained commands, periodic inputs, and fixed random sequences. Finger uses a contact-oriented bank with a sustained $(1, 1)$ reference command. Here, counterfactual responses are model predictions under alternative action sequences from a common starting point. Appendix D specifies the banks and state normalization.

4.2 Comparing responses and selecting a policy

Write C_q and C_m for the query and memory contrasts, and let $u = C/(\|C\|_2 + \epsilon)$. We compare them with

$$d(q, m) = 1 - u_q^\top u_m + \alpha \left| \log \left(\frac{\|C_q\|_2}{\|C_m\|_2} + \epsilon \right) \right|, \tag{4}$$

using $\alpha = 0.2$ and $\epsilon = 10^{-8}$. The first term compares the pattern of responses across probes. The magnitude term distinguishes contrasts with similar directions but different response strengths.

We average the distances within each policy's memory group and select the lowest score:

$$D(q, p) = \frac{1}{|\mathcal{M}_p|} \sum_{m \in \mathcal{M}_p} d(q, m), \qquad \hat{p}(q) = \arg\min_p D(q, p). \tag{5}$$

Averaging scores a policy across multiple source states. The **CF-ng** ablation instead selects the policy attached to the nearest memory, $\arg\min_m d(q, m)$.

Memory fingerprints can be precomputed. At a new query, the method evaluates KH model transitions and compares its contrast vector with the stored vectors. With M memories and fingerprint dimension d_C, distance computation costs $O(Md_C)$ and group aggregation costs $O(M)$. The online probes are model rollouts; the learned implementation first acquires a short interaction history to condition them.

4.3 Fingerprints from a visual world model

The learned model encodes the current image as a latent state z_0 and recent image-action-reward history as context c. A context-conditioned transition predicts $z_{t+1} = T_\theta(z_t, a_t, c)$; a reward head predicts r_t, and a state decoder maps z_t to normalized physical features. Context is fixed while the model rolls forward under each probe. Decoded trajectories and predicted rewards then enter the same contrast and retrieval equations.

The history representation includes visual latents, actions, rewards, and one-step residuals relative to a frozen nominal transition model. Products of these residuals with the current and previous mean action encode recent action-response relationships.

Training combines latent-transition prediction, state reconstruction, reward prediction, and an episode-level contrastive objective for context. Simulator-generated probe trajectories provide an additional rollout loss on decoded states and rewards. The new models use 50,000 updates, 600 training probe examples, and 40 held-out examples. Training episodes come from source regimes. Appendix D gives architecture and training details.

At inference, fingerprint generation uses images, actions, and rewards; the selected SAC controller receives physical-state observations.

5 Experimental Setup

5.1 Controlled policy-transfer matrix

We use modified DeepMind Control Suite tasks (Tassa et al., 2018): Cheetah-run, Walker-walk, Hopper-hop, and Finger-spin. Per-regime policies use a compact SAC implementation (Haarnoja et al., 2018). Each task has six source regimes and three held-out regimes. The standard grid varies actuator strength, body mass, damping, and a one-step delay on the first action channel. Finger replaces selected source perturbations with contact friction and damping changes. The exact grid is given in Appendix A.

Each run contains 25 memory states per source policy, for 150 memory items, and 12 queries per regime, for 108 queries. Each query-policy return is the mean of five 200-step closed-loop replays. We evaluate six policy-training seeds per task: Cheetah, Hopper, and Finger use seeds 0–5; Walker uses 1–6. This offset preserves Walker's initial matrix-batch numbering; its earlier seed-0 library used a different training budget and is used separately for switching (Appendix D).

The dynamics grids and distance were fixed after exploratory experiments. Held-out regimes were available during method development but excluded from the policy library; Appendix B reports their results separately.

Baselines. B0 randomly selects a policy. B1 averages observation distances by policy group. B2 uses nine nearest memory states with distance-weighted policy votes. B3 fits local linear state-transition models from the same probe trajectories and compares standardized parameter vectors. B4 predicts per-policy return by ridge regression from contrast fingerprints. B3 and B4 share the probe bank with CF; B4 additionally requires memory-policy replay-return labels.

Statistics. Tables report means and sample standard deviations across six policy-training seeds. Each run seed also controls external state sampling, so the variation includes both sources. Exact paired sign-flip tests use the six run-level mean differences and assume symmetry under the null. The minimum two-sided p-value is $2/64 = 0.03125$; values are unadjusted for multiple comparisons. We report effect sizes and the number of seeds favoring each method. Appendix D describes the additional query-level tests.

5.2 Learned-model and switching protocols

The learned-model study uses Hopper and Walker with frozen policy libraries. Each query history contains 20 random actions. Query histories and the five-replay return matrix are cached and shared across WM runs within a task. All candidates and query sets use one saved memory bank, generated by CPU actors in a separate seeded environment. Appendix D gives implementation and reproducibility details.

Hopper has three newly trained WMs. Walker WM/0 denotes an earlier model selected from two training restarts by held-out distillation loss under its original training setup; WM/1 and WM/2 are newly trained models. We report WM/0 separately because its training and prior selection differ. Query seed 1 defines the validation set used to choose a WM, and query seed 0 defines the test set. Both sets cover the same nine-regime grid and use separate query draws. Validation selection uses policy-replay returns from both source and held-out regimes. We compare selection by validation retrieval regret with selection by held-out distillation loss. The latter uses only the two new Walker WMs because WM/0 lacks comparable holdout metadata.

H1-context selects policies using distances between context embeddings from the same WM. **H2-history** maps the same packed history representation to a source-policy index, with source-regime identity as its training target and held-out-episode accuracy for model selection. H2 uses source-identity supervision, whereas B4 uses policy-return labels.

The Walker switching study evaluates known, unknown, randomized, and absent change times using one WM configuration. Retrieval checks every 40 steps, requires a 15% score improvement, and uses a 60-step cooldown. It compares the query with an averaged response prototype per policy (Appendix C). Within each scenario and shift, all methods share eight initial states and change times. A stochastic replay reference averages three full-remaining-horizon rollouts per candidate and commits once; retrieval can switch again. Evaluation runs on CPU with separate random streams for execution and branch replay.

6 Results

6.1 Simulator fingerprints reduce selection regret

Simulator-generated fingerprints reduce mean regret relative to random selection by 81%, 66%, 49%, and 63% on Cheetah, Walker, Hopper, and Finger, respectively (Table 1). CF improves on random selection and both current-state retrieval baselines in every seed of every task (Figure 2). All twelve task-baseline comparisons have an unadjusted exact seed-level p-value of 0.03125.

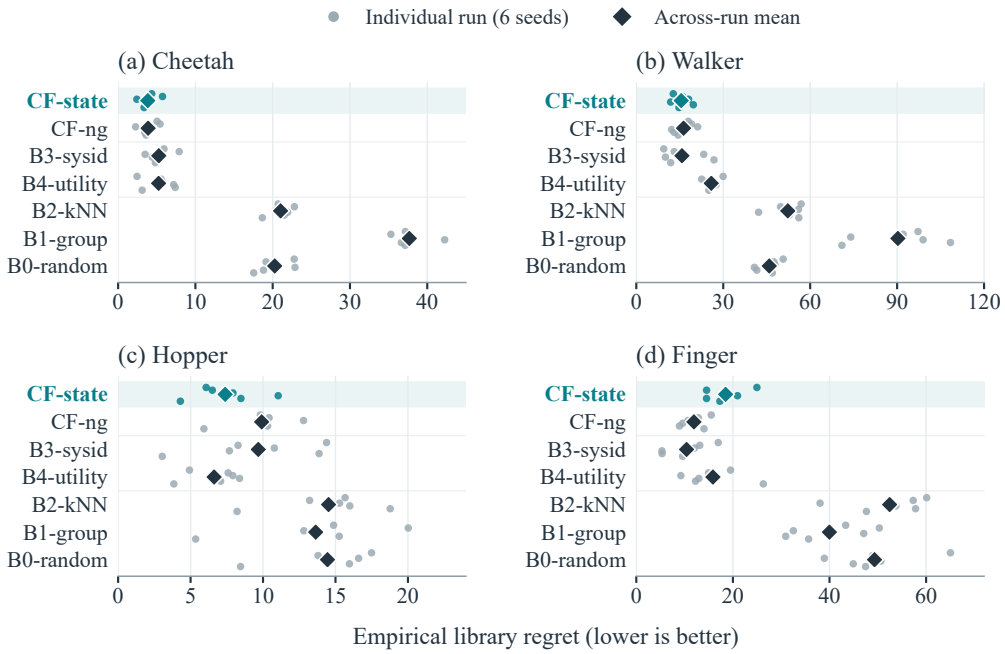


Figure 2: Policy retrieval with simulator-generated fingerprints across four tasks. Small circles show each run's mean regret over 108 queries; diamonds show the mean across six policy-training seeds. Lower is better. All seven methods from Table 1 are shown. Horizontal axes use task-specific scales.

Table 1: Empirical library regret with simulator-generated fingerprints, mean ± sample standard deviation over six policy-training seeds. Lower is better. All methods share the query return matrix; B4 additionally uses memory return labels. CF-state uses the true query dynamics to generate simulator probes. Bold denotes the lowest mean in each task.

Method	Cheetah	Walker	Hopper	Finger
CF-state	**3.84** ± 1.12	**15.50** ± 3.01	7.39 ± 2.32	18.49 ± 4.01
CF-ng	3.88 ± 1.16	16.31 ± 3.60	9.91 ± 2.23	11.93 ± 2.61
B3-sysid	5.26 ± 1.52	15.76 ± 7.38	9.67 ± 4.26	**10.40** ± 4.59
B4-utility	5.24 ± 2.05	25.87 ± 2.57	**6.62** ± 1.82	15.87 ± 6.14
B2-kNN	21.02 ± 1.43	52.28 ± 5.61	14.52 ± 3.57	52.45 ± 8.29
B1-group	37.68 ± 2.37	90.27 ± 14.79	13.63 ± 4.78	39.98 ± 8.06
B0-random	20.26 ± 2.19	45.87 ± 3.88	14.45 ± 3.24	49.32 ± 8.71

Against dynamics-informed baselines, CF has lower average regret than B3 on Cheetah, where it leads in six of six seeds ($p = 0.03125$), and on Hopper (four of six seeds). Walker means are close (15.50 versus 15.76), although CF leads in only two of six seeds. B4 has similar mean regret on Cheetah and Hopper and higher regret on Walker, where CF leads in all six seeds ($p = 0.03125$). On Finger, CF-ng, B3, and B4 all outperform group-averaged CF; B3 leads CF in all six seeds ($p = 0.03125$). Appendix B separates source-regime and held-out-regime results.

6.2 Probe coverage and aggregation affect retrieval

The eight-probe configuration has lower task-average regret than either tested prefix on all four tasks, although individual seeds can reverse this ordering. Reducing K from eight to two increases Walker regret from 15.50 to 26.67. Longer prefixes add both action responses and model transitions, so this comparison changes probe coverage and computation together.

Table 2: Probe-prefix ablation, mean ± sample standard deviation across the same policy seeds. Probe selection order is fixed. K changes both response information and rollout budget.

Task	K=8	K=4 prefix	K=2 prefix
Cheetah	3.84 ± 1.12	4.63 ± 1.50	5.73 ± 2.82
Walker	15.50 ± 3.01	19.61 ± 4.36	26.67 ± 7.68
Hopper	7.39 ± 2.32	7.75 ± 2.20	9.29 ± 3.57
Finger	18.49 ± 4.01	30.27 ± 4.29	31.10 ± 7.61

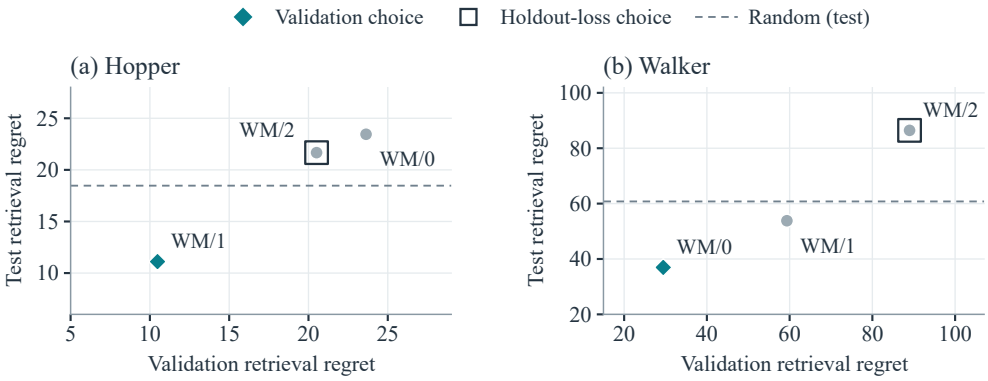


Figure 3: World-model selection on separate validation and test query draws. Each point is one candidate from Table 3; lower regret is better on both axes. Diamonds mark the lowest validation regret, and outlined squares mark the lowest holdout rollout loss. The dashed line is random policy selection on the test set. Walker's loss-based choice is restricted to its two new runs; WM/0 has no comparable holdout loss.

Removing group aggregation increases Hopper regret from 7.39 to 9.91 and Walker regret from 15.50 to 16.31, while reducing Finger regret from 18.49 to 11.93. On Finger, the nearest-memory rule is more effective than averaging over each policy's memory group.

6.3 Validation regret and prediction loss select different models

Figure 3 compares validation and test regret for every world-model candidate.

Table 3: Regret for each evaluated WM. Query sets and policy libraries are fixed within task. Walker WM/0 is the earlier model selected from two restarts. CF-state uses simulator fingerprints at the same query states.

Task / WM run	CF-WM	CF-state	H1-context	H2-history	Random
Hopper / 0	23.45	6.99	23.40	18.15	18.47
Hopper / 1	11.11	6.99	19.02	19.25	18.47
Hopper / 2	21.67	6.99	23.30	21.24	18.47
Walker / 1	53.81	10.75	80.60	19.17	60.78
Walker / 2	86.45	10.75	31.18	20.92	60.78
Walker / 0 (prior)	36.96	10.75	10.09	24.64	60.78

Validation retrieval regret selects Hopper WM/1 and Walker WM/0. On Hopper, WM/1 has validation regret 10.48 and test regret 11.11, compared with 18.47 for random selection: a 39.8% reduction. On Walker, WM/0 has validation regret 29.45 and test regret 36.96, compared with 60.78 for random selection: a 39.2% reduction.

Held-out distillation loss selects WM/2 on both tasks, with test regrets of 21.67 on Hopper and 86.45 on Walker, both worse than random selection. Hopper's holdout losses span 0.43–18.30, yet this large

separation does not identify the model with the lowest retrieval regret. Within the two new Walker candidates, validation regret ranks run 1 ahead of run 2 (59.34 versus 88.99), and their test regrets follow that order (53.81 versus 86.45). Holdout loss ranks them in the reverse order.

Learned-model performance varies widely across runs (Table 3). Hopper regrets range from 11.11 to 23.45, with one of three runs improving on random selection. Simulator fingerprints on the same queries achieve 6.99 on Hopper and 10.75 on Walker, below every learned candidate.

On Hopper, the validation-selected CF-WM achieves regret 11.11, compared with 19.02 for H1-context and 19.25 for H2-history using the same WM. On Walker, H2-history outperforms CF-WM for every evaluated model, and WM/0 gives H1-context regret 10.09. Predicting probe responses improves on these direct history readouts for the selected Hopper model, but not for Walker.

6.4 Switching helps some shifts and harms others

Table 4: Walker demonstration with known change time. Mean post-change return over eight episodes; higher is better. The replay reference chooses once using full-horizon stochastic estimates. The learned and frame retrievers can switch again. This application study uses one world-model configuration.

Shift	Fixed	Replay reference	CF-WM	Frame
D1:D2	130.1	334.9	164.0	171.6
D1:D9	489.0	489.0	465.5	426.6
D8:D2	286.3	334.7	207.9	137.3
D9:D7	30.8	123.6	73.2	72.9
D1:D8	359.2	411.7	329.5	270.3

CF-WM improves on the fixed policy for D1→D2 and D9→D7. In these known-time conditions, its returns are 164.0 and 73.2, compared with 171.6 and 72.9 for frame retrieval. It loses return on D8→D2, D1→D9, and D1→D8. With no change in D8, return falls from 413.4 to 310.3, a 24.9% loss.

When the change time is unknown or randomized, retrieval receives no change-time input and shows the same pattern of gains and losses (Appendix C).

7 Discussion and Limitations

Response information improves retrieval over current-state matching, but its readout matters: CF has the lowest mean regret on Cheetah and Walker, utility regression on Hopper, and system identification on Finger.

Validation retrieval regret selects better retrievers than rollout loss in both candidate pools, using policy-return labels from the same dynamics grid. The single validation/test split and fixed libraries and memories leave the stability of these gains across new draws untested.

Comparing probe banks at a fixed rollout budget would separate action coverage from computation. Finger motivates testing aggregation rules for contact-rich tasks. Online selection must also account for switch costs, including the cost of leaving a suitable policy when dynamics remain unchanged.

8 Conclusion

Counterfactual dynamics-aware retrieval selects policies by matching action responses. Simulator fingerprints outperform state-similarity retrieval across four tasks and six seeds per task. With learned fingerprints, validation retrieval regret selects useful models that prediction loss misses, motivating model evaluation through downstream policy choices.

AI USE STATEMENT

During the preparation of this work the authors used generative AI tools to search for and identify related literature. After using these tools, the authors reviewed the results and take full responsibility for the content of this article.

REPRODUCIBILITY STATEMENT

Appendices A–D detail the experiments. The supplementary package provides code, frozen policies, archived results, and instructions for regenerating the tables and figures.

REFERENCES

Benjamin Eysenbach, Swapnil Asawa, Shreyas Chaudhari, Sergey Levine, and Ruslan Salakhutdinov. Off-dynamics reinforcement learning: Training for transfer with domain classifiers. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2006.13916.

Anirudh Goyal, Abram L. Friesen, Theophane Weber, Andrea Banino, Nan Rosemary Ke, Adrià Puigdomènech Badia, Arthur Guez, Mehdi Mirza, Peter C. Humphreys, Ksenia Konyushkova, Laurent Sifre, Michal Valko, Simon Osindero, Timothy Lillicrap, Nicolas Heess, and Charles Blundell. Retrieval-augmented reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 7740–7765. PMLR, 2022. URL https://proceedings.mlr.press/v162/goyal22a.html.

Yihong Guo, Yu Yang, Pan Xu, and Anqi Liu. MOBODY: Model-based off-dynamics offline reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2506.08460.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.01603.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Proceedings of Robotics: Science and Systems*, 2021. doi: 10.15607/RSS.2021.XVII.011. URL https://www.roboticsproceedings.org/rss17/p011.html.

Jiafei Lyu, Kang Xu, Jiacheng Xu, Mengbei Yan, Jingwen Yang, Zongzhang Zhang, Chenjia Bai, Zongqing Lu, and Xiu Li. ODRL: A benchmark for off-dynamics reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6e37ba8a27b53fccd367c648ff365b8f-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Gabriel B. Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. *The International Journal of Robotics Research*, 43(4), 2024. doi: 10.1177/02783649231224053. URL https://journals.sagepub.com/doi/10.1177/02783649231224053.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1803.11347.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation*, 2018. doi: 10.1109/ICRA.2018.8460528. URL https://xbpeng.github.io/projects/SimToReal/SimToReal_2018.pdf.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 5331–5340. PMLR, 2019. URL https://proceedings.mlr.press/v97/rakelly19a.html.

Benjamin Rosman, Majd Hawasly, and Subramanian Ramamoorthy. Bayesian policy reuse. *Machine Learning*, 104(1):99–127, 2016. doi: 10.1007/s10994-016-5547-y. URL https://arxiv.org/abs/1505.00284.

Ram Ananth Sreenivasan, Hyun-Rok Lee, Yeonjeong Jeong, Jongseong Jang, Dongsub Shim, and Chi-Guhn Lee. A learnable similarity metric for transfer learning with dynamics mismatch. In *PRL Workshop at IJCAI: Bridging the Gap Between AI Planning and Reinforcement Learning*, 2023. URL https://openreview.net/pdf?id=78n3SG6xMY.

Zhenyu Tao, Wei Xu, and Xiaohu You. A generalized bisimulation metric of state similarity between Markov decision processes: From theoretical propositions to applications. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/458567910b6d21f438f22aa20c036723-Abstract-Conference.html.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind control suite, 2018. URL https://arxiv.org/abs/1801.00690. arXiv:1801.00690.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017. URL https://arxiv.org/abs/1703.06907.

Yunzhe Xu, Yiyuan Pan, and Zhe Liu. Dream to recall: Imagination-guided experience retrieval for memory-persistent vision-and-language navigation, 2026. URL https://arxiv.org/abs/2510.08553. arXiv:2510.08553. Accepted by IEEE TPAMI.

Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Proceedings of Robotics: Science and Systems*, 2017. doi: 10.15607/RSS.2017.XIII.048. URL https://www.roboticsproceedings.org/rss13/p48.html.

A Dynamics Grid

All parameters below are multipliers of the task configuration unless specified. D1/D2/D3/D7/D8/D9 form the policy library; D4/D5/D6 are held-out regimes. The standard grid applies to Cheetah, Walker, and Hopper.

Table 5: Dynamics regimes. Multipliers are relative to the task configuration. The standard grid applies to Cheetah, Walker, and Hopper; Finger overrides are listed separately.

Regime	Standard grid	Finger override	Split
D1	Nominal	Nominal	Source
D2	Actuator gear ×0.45	Gear ×0.75	Source
D3	Main-body mass ×1.6	Contact friction ×0.3	Source
D4	Gear ×0.7	Same	Held out
D5	Main-body mass ×1.3	Contact friction ×0.6	Held out
D6	Selected joint damping ×3	Same	Held out
D7	Gear ×0.3	Selected joint damping ×2.5	Source
D8	Gear ×0.6	Same	Source
D9	First action channel delayed by one step	Same	Source

B Source and Held-out Regimes

Table 6: Mean regret across policy seeds. Source regimes contribute 72 queries per run; held-out regimes contribute 36. Exact per-seed values are retained in the source CSV files.

Task	Regimes	CF-state	CF-ng	B3-sysid	B4-utility
Cheetah	seen	4.65	4.56	6.26	4.32
Cheetah	unseen	2.23	2.50	3.26	7.08
Walker	seen	15.83	15.65	16.98	23.60
Walker	unseen	14.84	17.63	13.33	30.41
Hopper	seen	6.12	8.17	7.30	5.22
Hopper	unseen	9.92	13.39	14.43	9.42
Finger	seen	19.93	13.63	10.73	16.02
Finger	unseen	15.62	8.53	9.75	15.57

C Complete Switching Means

Each episode lasts 990 steps. Known- and unknown-time changes occur at step 500. Retrieval begins after the change in the known-time setting and after an 80-step warmup in the unknown-time setting. Random changes occur between steps 300 and 700; no-change runs retain the initial dynamics. Return is accumulated from the designated change point to the end of the episode.

This application averages normalized memory contrasts within each policy and renormalizes the result to a direction prototype $\bar{u}_p$. Let $\bar{\ell}_p$ be the mean log norm of that policy's memory contrasts. The switching score is

$$D_{\mathrm{switch}}(q,p) = 1 - u_q^\top \bar{u}_p + 0.2\left|\log(\|C_q\|_2 + \epsilon) - \bar{\ell}_p\right|. \tag{6}$$

This prototype score differs from averaging item distances in the transfer matrix. The fixed margin and cooldown apply to every tested shift.

Table 7: Rows with randomized change time have different remaining episode lengths. Comparisons are within each row. These tables report means; the accompanying episode logs retain individual returns, switch counts, seeds, and initial-state identifiers.

Scenario	Shift	Fixed	Replay reference	CF-WM	Frame
known	D1:D2	130.1	334.9	164.0	171.6
known	D1:D9	489.0	489.0	465.5	426.6
known	D8:D2	286.3	334.7	207.9	137.3
known	D9:D7	30.8	123.6	73.2	72.9
known	D1:D8	359.2	411.7	329.5	270.3
unknown	D1:D2	126.6	331.5	192.9	164.6
unknown	D1:D9	489.0	489.0	472.3	478.7
unknown	D8:D2	287.0	334.6	159.6	150.5
unknown	D9:D7	32.3	120.9	62.8	75.6
unknown	D1:D8	379.4	412.6	303.9	205.3
random	D1:D2	140.2	348.1	173.1	141.4
random	D1:D9	503.0	503.0	489.9	497.8
random	D8:D2	295.5	344.2	173.6	137.4
random	D9:D7	30.4	131.1	79.3	80.9
random	D1:D8	381.2	423.7	321.4	278.1
none	D1:D1	489.7	489.7	467.1	387.9
none	D8:D8	413.4	413.4	310.3	255.4
none	D9:D9	474.2	474.2	476.0	355.9

D Implementation and Reproducibility

D.1 State features and probe banks

Fingerprints concatenate generalized positions and velocities. The first three position coordinates are divided by 2, the remaining position coordinates by π, and all velocity coordinates by 8. Probe responses contain all 26 states, including the starting state, and one mean-reward value.

For the locomotion matrix tasks, with action dimension A, the first three probes are zero action, a sustained vector with the first $\lfloor A/2 \rfloor + 1$ entries equal to 1 and the remainder equal to -1, and its negative. The fourth probe is $\sin(2\pi t/H)$ with opposite signs on the two action halves. The fifth is $\cos(6\pi t/H)$ with alternating signs across coordinates. Three fixed random sequences complete the bank, drawn from $[-1, 1]$, $[-0.5, 0.5]$, and $[-1, 1]$, respectively, using NumPy generator seed 777.

Finger begins with sustained commands $(1, 1)$, $(1, -1)$, $(0.6, 0.6)$, and $(-0.8, 0.8)$. Its fifth probe uses $(\sin \omega_t, \cos(\omega_t/2))$ with $\omega_t = 2\pi(0.5 + 2t/H)$. The sixth alternates signs with a three-step circular offset between joints. The seventh applies $(v_t, -v_t)$, where $v_t = 0.9$ for the first $\lfloor H/3 \rfloor$ steps and -0.5 thereafter. The final probe is a fixed uniform random sequence in $[-1, 1]^2$.

The learned Hopper study uses the same bank. The learned Walker study uses a separate eight-probe bank, shared by WM/0, WM/1, and WM/2: zero action; sustained $(1, 1, 1, -1, -1, -1)$ and its negative; the same sinusoidal and alternating-sign cosine inputs; sustained $(0.8, -0.8, 0.4, -0.4, 0.8, -0.8)$; and two fixed random sequences with ranges $[-1, 1]$ and $[-0.5, 0.5]$. Each study shares its bank across all compared methods.

D.2 Baseline estimators

State distances are squared Euclidean distances between normalized generalized state vectors. B1 averages these distances within policy groups. B2 uses $k = 9$ and votes with weights $1/(d + \epsilon)$.

For B3, each memory or query yields KH transitions. With rows $x_t = [\tilde{s}_t, a_t, 1]$ and targets $y_t = \tilde{s}_{t+1} - \tilde{s}_t$, the fitted coefficient matrix is

$$W = (X^\top X + 10^{-3} I)^{-1} X^\top Y. \tag{7}$$

Flattened coefficients are standardized using the memory-bank mean and standard deviation. Squared parameter distances are averaged over each policy group.

For B4, let X_m and X_q be contrast features standardized using the memory-bank statistics, and let J_m contain replay-return labels for every memory-policy pair. Predicted utilities are

$$\widehat{J}_q = X_q X_m^\top (X_m X_m^\top + I)^{-1} J_m. \tag{8}$$

The chosen policy maximizes its predicted return. B4 uses the same contrast features as CF and additional return labels collected by replaying the policy library from memory states.

D.3 Learned model and training

SAC uses two 256-unit ReLU hidden layers per actor and critic, a replay capacity of one million transitions, and one million training steps per source policy. The first 5,000 steps use random actions. Updates use batch size 256, Adam at 3×10^{-4}, discount 0.99, target-update rate 0.005, and learned entropy temperature with target entropy -1. Return evaluation uses stochastic policies. Learned retrieval fixes the Hopper seed-0 and Walker seed-1 matrix libraries; switching uses an earlier Walker seed-0 library trained for 600,000 steps per policy.

Images are rendered at 84×84, converted to grayscale, and standardized per frame. The encoder has three convolutional layers with output channels 32/64/64, kernel sizes 8/4/3, strides 4/2/1, and SiLU activations, followed by a linear projection to 48 latent coordinates. The context encoder maps a flattened 20-step history through a 512-unit hidden layer to 24 coordinates. The transition and state-decoder heads have 256-unit hidden layers; the reward and nominal-transition heads have 128-unit hidden layers.

New-model training collects 40 episodes of 100 steps per source regime using uniform actions or sustained and periodic exploratory commands. The nominal transition is trained on nominal-regime episodes for 1,500 Adam updates at learning rate 10^{-3} and then frozen. The remaining trainable parameters use Adam at 2×10^{-4} for 50,000 updates. Context features are refreshed every 5,000 updates.

Ordinary minibatches contain 64 transitions. Their loss combines latent squared error and cosine error, decoded-state squared error, reward squared error weighted by 0.5, and episode-level InfoNCE weighted by 0.1. With probability 0.35, an update instead uses three simulator probe examples. This rollout loss combines state squared error, trajectory cosine error weighted by 0.5, absolute log-magnitude error weighted by 0.3, and mean-reward squared error weighted by 0.5. Rewards are scaled by the mean absolute training reward. The 40 held-out probe examples measure this same rollout loss.

H2 has a 256-unit SiLU hidden layer and six outputs. It trains for 30 epochs with Adam at 10^{-3} and batch size 256. Ten percent of source episodes are held out for checkpoint selection by classification accuracy. Each head, including Walker WM/0's, is saved and reused.

D.4 Randomness and evidence

The matrix extends Walker seeds 1–3 with 4–6 and other tasks' seeds 0–2 with 3–5 under the same protocol, preserving the initial numbering. The learned-model study keeps the policy library fixed.

Separate seeded environments generate queries and memories. Models and query sets share a saved CPU-built memory bank. The probe anchor image is the post-history frame, aligned with the state used for return replay.

The CPU switching evaluation isolates sampling, execution, and branch-replay random states. Its 720 records form 144 paired groups. In no-change controls, the fixed policy and both branch selectors have equal per-episode returns.

Query-level tests use 10,000 sign flips with correction $(b + 1)/10{,}001$, where b counts sampled statistics at least as extreme as observed. Seed-level tests enumerate all 64 sign assignments.