A Third of Forgetting Is the Convention: How the Encoding of a Skill Decides What Later Fine-Tuning Destroys

Clark Whitehead
Independent Researcher
clarkwhitehead.com

November 2026
Rough draft v0.1 — measured result, not peer reviewed

Abstract

Most tasks have several equally correct answers, and a demonstration has to show one of them. We call that pick the *convention*: it is not content, because any of the answers would do. We ask whether the conventions carried by the demonstrations of a skill decide how much of that skill later fine-tuning destroys. In grid worlds with an exact solver, every demonstration is provably optimal, so the content of every demonstration can be held fixed while only the convention varies. Teaching a second task from demonstrations that break ties the opposite way from the first task's costs 0.10 of retention compared with demonstrations that break ties the same way, in 24 of 24 task-pair and seed cells; under a clashing convention, a third of all forgetting is the convention. A skill learned from the maximum-entropy target over all optimal actions keeps 0.10 more under the clashing convention and gains nothing when the conventions happen to agree: it is immune to the clash, not more durable in general. Choosing, among optimal demonstrations, the one the learner already prefers (an oracle form of on-policy data) recovers exactly the convention effect. At matched KL divergence from the original policy a convention-free skill loses half as much success, so KL is not a sufficient statistic for forgetting. The effect survives changes of optimizer state, learning rate and width, and a synthetic classifier reproduces it with a gap that grows with the share of inputs that have equivalent labels. The study comprises 1,088 runs with pre-registered predictions and about $4 of GPU time.

1 Introduction

Fine-tuning a network on a new task damages what it learned before. Recent work explains the damage through the *new* task's update: on-policy data forgets less than demonstrations [1, 2], the model's own outputs make safer targets [3], and the harm concentrates on tokens where the model is confident and the data disagrees [4]. A second line shows that the *old* model's state matters as well: training regimes that widen minima [5, 6], pretraining length [7], learning-rate decay and sharpness [8, 9], the weight-space entropy of the solution [10] and early exposure to the skill's data [11] all change how much later fine-tuning removes.

Both lines leave one question open. Whenever several answers are equally good, a demonstration commits to one of them, and the old skill carries that commitment. Do these arbitrary commitments matter for forgetting, and how much? The question is hard to study in language models because content and convention are mixed: a demonstration written in a different style may also differ in correctness, and a token where the model is confident and the data disagrees may be a fact or a formatting habit.

Grid-world games with an exact solver remove that confound. A breadth-first search over the full game state gives every reachable state's distance to a win and its set of optimal actions. A demonstration can then

be provably optimal while differing from another only in *which* optimal action it shows, and a target can show all optimal actions at once and so carry no convention at all. We use this to ask how the encoding of the first skill decides what a second task's training destroys, with the second task's update held fixed.

Contributions.

1. With content held fixed, the convention alone moves retention by 0.10 (24 of 24 cells), and under a clashing convention it accounts for about a third of all forgetting (Section 4.1).
2. A maximum-entropy encoding of the first skill is forgotten more slowly only when the next task's convention clashes with the canonical one; when the conventions agree it gains nothing. The encoding is immune, not durable (Section 4.2).
3. Re-choosing, among optimal demonstrations, the one the learner already prefers recovers exactly the convention effect, which gives a controlled account of one reason on-policy data forgets less (Section 4.3).
4. At matched KL divergence from the original policy a convention-free skill loses half as much; most of a canonical skill's measured drift is harmless movement among equally good actions (Section 4.4).
5. On-policy refinement after imitation leaves the convention in place unless an entropy bonus strong enough to cost accuracy flattens it; the durable encoding comes from the targets (Section 4.5).
6. The result holds with Adam's state kept across the task boundary, a three times higher learning rate, twice the width and heavy label smoothing; a synthetic classifier reproduces it, with a gap that grows from zero to 15 points as the share of inputs with equivalent labels rises from 0% to 100% (Sections 4.6 and 4.7).

2 Related Work

The new task's update. RL's Razor [1] reports that forgetting is predicted by the KL divergence between the fine-tuned and the original policy and that on-policy reinforcement learning is KL-minimal. Retaining by Doing [2] attributes the advantage to on-policy data rather than the algorithm, measures a KL–forgetting correlation of only 0.52, and shows in a Gaussian toy that whether the initial policy has one mode or several decides whether supervised fine-tuning or reinforcement learning forgets more. Self-distillation fine-tuning [3] and related methods use the model's own outputs as targets. Entropy-adaptive fine-tuning [4] identifies "confident conflicts", tokens of low probability and low entropy, as the main driver of forgetting in supervised fine-tuning and down-weights them. Entropy-preserving losses [17] and target-distribution design [18] change the targets of the new task. Reinforcement fine-tuning is reported to mitigate forgetting in continual post-training [23], and a recent theory [22] argues that supervised fine-tuning on correct reasoning paths makes dense updates that overwrite. All of this varies the new task; our study holds it fixed.

The old model's state. Mirzadeh et al. [5] show that dropout, learning-rate decay and batch size on the first task widen its minima and reduce later forgetting, and that wider networks forget less [6]. Springer et al. [7] show that longer pretraining makes parameters more sensitive to any later change. In 2026, learning-rate decay in pretraining [8], sharpness-aware pretraining [9], the weight-space entropy of the trained solution [10] and early exposure to the skill's data [11] were each shown to change how much later fine-tuning removes; Feng et al. [11] also report that performance right after learning does not predict retention. Every lever in this line is the optimizer, the loss geometry or the timing of the data. None varies what the demonstrations commit to when several answers are equally good, and none can separate convention from content.

Interference. Kessler et al. [12] show that tasks sharing an observation but with opposite optimal actions interfere and cause forgetting; a 2026 theory [13] gives a distortion floor for conflicting targets on overlapping inputs. Here interference appears even when both actions are optimal, and the conflict is avoidable by the choice of target.

What forgetting is in language models. Fine-tuning skews which task the model infers [14], much of the measured drop is lost task alignment rather than lost knowledge [15], and alignment tuning shifts mostly stylistic tokens [16]. Style is the language-model analogue of our convention; these results suggest the convention share of forgetting could be large there, which we leave to future work.

Conventions and ties in imitation. Shih et al. [19] separate rule-dependent from convention-dependent representations in multi-agent coordination. Searchformer [20] generates A* traces with deterministic or random tie-breaking and notes that random ties give several correct responses, without a retention ablation. Label smoothing [21] is the standard soft target; we found no study that applies it to the old task and measures later forgetting, so our label-smoothing arms are that test.

3 Setting

3.1 Environments and the exact oracle

We use three task families from the Long Game toolkit [24], a public set of grid-world games with hidden rules: *keys* and *gates* (single-rule worlds, tiers 1 to 3, maps from 7×4 to 11×11) and *open*, an open-room family built on the same engine (10×9 interior, 12% random walls, start and exit at least 6 apart). A task is a distribution over levels: 8,000 training levels, 300 fresh evaluation levels from disjoint seeds, and the first 300 training levels as a memorisation set.

For every level a breadth-first search over the full game state (player position, crates, keys held, doors, gates) gives every reachable state, its exact distance to a win, the optimal action set $O(s)$, the shortest-path DAG and the number of optimal action sequences. States with more than one optimal action (ties) are 17 to 21% of shortest-path states in keys, 16 to 18% in gates and 47% in open. Evaluation is exact: greedy rollouts on the transition table, success within three times par.

3.2 Encodings of the first skill, and demonstrations for the second

Each training arm is a way of turning the oracle into targets. The actions are up, right, down, left and use.

- **c1** and **c2**: one canonical shortest path per level, ties broken by a fixed order (c1: up, right, down, left, use; c2: left, down, right, up, use). Identical content, opposite conventions.
- **rand**: one random optimal path per level, like demonstrations collected from many people.
- **soft**: on the states of the c1 path, the target is uniform over $O(s)$. Same states as c1, all optimal actions at once.
- **dag**: the maximum-entropy optimal policy on every shortest-path state.
- **ls**: c1 with label smoothing of 0.1 over all actions (0.3 and 0.5 in a follow-up).
- **closest** (second task only): every 100 steps, the shortest path most likely under the current policy is re-chosen by dynamic programming over the DAG. This is an oracle form of on-policy data: the demonstration is optimal and is the one the learner already prefers.

- **rl0**, **rl1** (follow-up): c1 imitation followed by 1,000 REINFORCE updates on the first task with entropy bonus 0 or 0.01 (0.1 in a further follow-up); **c1x**: 1,000 more imitation steps instead.

3.3 Protocol

The policy is a small convolutional network: a one-hot encoding of 16 cell types on a 12×12 frame, three 3×3 convolutions with 64 channels, a hidden layer of 256 units and 5 outputs, about 2.5 million parameters. Phase A trains the first task for 6,000 steps (Adam, learning rate 10^{-3}, batch 256 states). Phase B trains the second task for 3,000 steps with a fresh Adam state, learning rate 3×10^{-4} and a 100-step linear warm-up, with no data from the first task. The first task is evaluated exactly at 28 log-spaced points during phase B. The loss is cross-entropy against the arm's target distribution. The main grid is 6 ordered task pairs (keys, gates and open in both orders) $\times$ 26 arm pairs $\times$ 4 seeds $=$ 624 runs on one RTX 4090; follow-ups add 136 runs and the synthetic study 328 CPU runs.

3.4 Measures

Retention is the mean success of the first task on fresh levels over the 28 evaluation points of phase B, divided by its success at the end of phase A (1.0 means nothing forgotten). We also report per-level survival, optimal-action mass and the greedy-optimal rate on DAG states (split into states with a unique optimal action and tied states), the divergence $\mathrm{KL}(\pi_A \,\|\, \pi_t)$ on first-task states with its chain-rule split into mass leaving $O(s)$ (harmful) and drift inside $O(s)$ (harmless), the agreement of the greedy action with each tie-break convention on tied states, and the logit margin between the best optimal and the best non-optimal action. Paired contrasts compare cells with the same task pair and seed ($n = 24$); we report the mean difference, its standard error and the number of cells with the same sign.

3.5 Pre-registration

Five predictions were written before the main sweep (`PROTOCOL.md` in the repository). P1: maximum-entropy encodings are forgotten more slowly than canonical ones under identical phase-B training. P2: with the first task taught as c1, a second task taught with the same convention forgets less than with the opposite one, with rand in between. P3: learner-preferred demonstrations forget least among hard-target arms. P4: KL rises substantially even when success does not fall, and optimal-action mass tracks success better than KL. P5: memorised training levels are not protected relative to fresh levels. P2 to P5 held as written; P1 held only in a qualified form, as Section 4.2 explains.

4 Results

4.1 Conventions, not content, drive a third of forgetting

We teach the second task from demonstrations that are optimal either way but break ties the same way as the first task's demonstrations (c1 after c1) or the opposite way (c2 after c1).

- Same convention 0.798 versus opposite 0.695 retention: $+0.103 \pm 0.028$, positive in **24 of 24** pair-seed cells. The mirror image (c2 after c2 versus c1 after c2) gives $+0.108 \pm 0.029$, 24 of 24.
- A mix of conventions (rand) sits in between: $+0.042 \pm 0.019$ over the opposite convention.
- With the opposite convention the canonical skill loses 30% of its success. A third of that loss (0.10 of 0.30) is removed by matching the tie-break alone, with the content of every demonstration unchanged.

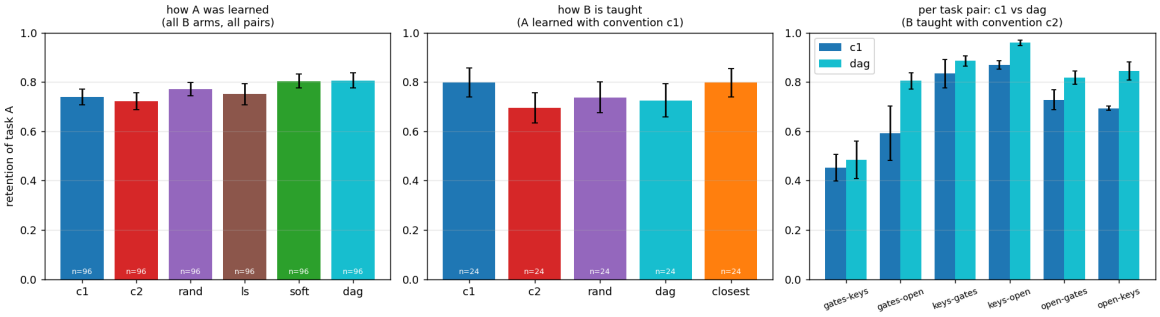


Figure 1: Retention of the first task. Left: by how the first skill was learned (c1 and c2 one canonical path each with opposite tie-breaks, rand one random optimal path per level, ls label smoothing, soft and dag all optimal actions at once). Middle: by how the second task is taught when the first used c1: same convention 0.80, opposite 0.70, learner-preferred path 0.80. Right: canonical against all-answers encodings per task pair.

- The convention is really adopted: on tied states the agreement of the greedy action with the c1 tie-break falls from 0.98 to about 0.15 after c2 training and stays near 0.9 after c1 training.

4.2 A convention-free encoding is immune to the clash, not more durable

With the second task always taught with the opposite convention (c2), the encoding of the first skill decides how much survives (Table 1).

Table 1: Retention difference against the canonical encoding c1 when the second task uses the opposite convention; $n = 24$ pair-seed cells. The last column is the policy's entropy over tied optimal actions at the end of phase A (maximum 0.69 for two-way ties).

Encoding of the first skill	Retention − c1	Same sign	Tie entropy
soft: all optimal actions, same states as c1	$+\mathbf{0.099 \pm 0.031}$	24 / 24	0.62
dag: all optimal actions, all shortest-path states	$+0.105 \pm 0.036$	23 / 24	0.64
rand: one random optimal path per level	$+0.080 \pm 0.034$	23 / 24	0.37
ls: label smoothing 0.1	$+0.020 \pm 0.040$	16 / 24	0.14
c1: one canonical path	0	—	0.02

Averaged over all four second-task arms ($n = 96$ each), retention is dag 0.806, soft 0.803, rand 0.771, ls 0.750, c1 0.739 and c2 0.722. Starting points are comparable (fresh success at the end of phase A: c1 0.898, soft 0.897, dag 0.918, rand 0.882, ls 0.919) and the second task is learned equally well in every arm (final success 0.79 to 0.84).

The decisive control is the case where the second task happens to share the first task's convention. There dag is no better than c1 (-0.007 ± 0.023). The maximum-entropy skill is therefore not more durable in general; it is immune to the convention clash that the canonical skill suffers whenever the next task's demonstrations break ties differently. The pre-registered prediction P1 holds only in this qualified form. Memorised training levels are not protected: their retention tracks fresh levels within 0.02 for every arm (P5).

4.3 Learner-preferred demonstrations remove the convention cost

Re-selecting, every 100 steps, the optimal path most likely under the current policy gives retention 0.798 versus 0.695 for the opposite convention ($+0.103 \pm 0.025$, **24 of 24**) and is indistinguishable from the same-convention arm (-0.001 ± 0.008), with the same final second-task success (0.81). It recovers exactly the

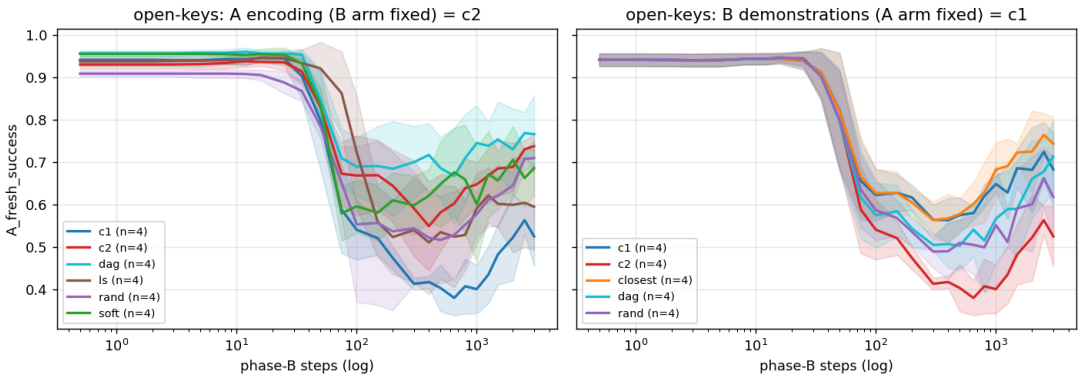


Figure 2: One pair in detail (open rooms, then keys): first-task success against second-task steps, log scale. Left, the first skill's encoding; the all-answers encodings stay highest. Right, the second task's demonstrations; the opposite convention hurts most and the learner-preferred path matches the same-convention line.

convention effect and no more. Among equally correct demonstrations, the ones the learner already prefers do not overwrite its conventions: a fully controlled measurement of one reason on-policy data forgets less [1, 2].

4.4 KL divergence is the wrong currency

$\mathrm{KL}(\pi_A \,\|\, \pi_t)$ on first-task states correlates with the success drop over all cells and times (Spearman 0.86), but at matched KL the loss depends on the encoding. In the KL bin $[0.5, 1)$ the drop is 0.29 (soft) and 0.30 (dag) against 0.40 (c1) and 0.42 (c2); in $[1, 2)$ it is 0.44 (soft) against 0.50 (c1), 0.56 (c2) and 0.63 (ls). A convention-free skill can move far in KL, because its preferences among equivalent actions are free to drift, without failing.

The chain-rule split says why. For a canonical skill (c1, second task c2, end of phase B), 1.41 of its 2.59 nats is drift *inside* the optimal set, its tie-breaks flipping to c2's, and 1.00 is mass leaving the optimal set. For the same-convention control (c2 after c2) the inside part is 0.13 of 0.97. Margins tell the same story: hard targets build the largest margin between the best optimal and the best non-optimal action (17.4 at the end of phase A against 11.6 for soft and 12.1 for dag) and lose the most of it during phase B (to 10.3), while the soft and dag margins slightly grow (13.1 to 13.2).

Forgetting is not concentrated on tied states. The greedy-optimal rate falls from 0.98 to 0.84–0.88 on states with a unique optimal action and from 0.99 to 0.89–0.93 on tied ones, for every arm. The convention is encoded as a global feature (prefer up and right), and overwriting it disturbs unique-action states as much as tied ones.

4.5 On-policy refinement leaves the convention in place

After 6,000 imitation steps with convention c1, we run 1,000 REINFORCE updates on the first task (batch 256 levels, reward 1 at a win, discount 0.97, fresh Adam at 3×10^{-4}; 3 pairs $\times$ 4 seeds, second task c2).

- Without an entropy bonus the policy stays optimal (batch success 1.00) and committed (entropy over tied optimal actions 0.04); retention $+0.030 \pm 0.026$ against c1 (9 of 12, not significant).
- With a bonus of 0.01: entropy 0.14, retention -0.001 ± 0.028. 1,000 more imitation steps instead: $+0.010 \pm 0.017$. dag on the same cells: $+0.098 \pm 0.038$ (12 of 12).

- With a strong bonus of 0.1 (the two open-room pairs) the policy does spread over the equivalents (entropy 0.65, like dag) and retention rises to 0.803 ± 0.051 against 0.711 for c1 and 0.832 for dag, but first-task success at the end of phase A falls from 0.94 to 0.85, because the bonus also puts mass on non-optimal actions.

On-policy refinement as usually practised, with little or no entropy bonus, leaves the convention in place and buys no durability. The durable encoding comes from the targets; reinforcement learning produces it only when made to flatten preferences among equivalent actions, at a cost in accuracy that the exact soft target does not pay.

4.6 Robustness

On the two open-room pairs (open then keys, open then gates; second task c2; 4 seeds):

- Keeping Adam's state across the task boundary instead of resetting it: dag $-$ c1 $= +0.131 \pm 0.032$ (8 of 8).
- Phase-B learning rate 10^{-3} instead of 3×10^{-4}: $+0.123 \pm 0.027$ (8 of 8).
- Twice the width (128 channels, 512 hidden units): $+0.116 \pm 0.069$ (6 of 8); retention 0.853 (dag) against 0.737 (c1).
- Heavier label smoothing: 0.3 gives retention 0.835 ± 0.035 and 0.5 gives 0.814 ± 0.022, against 0.711 for c1 and 0.832 for dag on the same pairs, while 0.1 gave only $+0.02$ (Table 1).

The common factor is how strongly the policy commits to its tie-break. The log-odds between the canonical and the alternative optimal action at the end of phase A are about 20 for c1, 3.8 for smoothing 0.1, 2.5 for 0.3, 1.8 for 0.5 and 0 for soft and dag, and retention rises in that order (0.69–0.71, 0.73, 0.835, 0.81, 0.83). In the synthetic model below, label smoothing instead hurts, so the smoothing route is not general; the route through all correct answers held in both.

4.7 A synthetic classifier shows the same signature

A minimal model reproduces the effect without any grid world. A multilayer perceptron classifies inputs from 64 Gaussian clusters, and each cluster has a *set* of correct labels (1 or 3 of 10). The second task is a second set of clusters sharing the output layer. Arms: canonical (always the lowest correct label), a fixed random correct label per cluster, a random correct label per sample, all correct labels at once (soft), and label smoothing. 8 seeds, 3,000 first-task steps, 2,000 second-task steps (fresh Adam, 3×10^{-4}, warm-up 100).

Table 2: Synthetic model: first-task accuracy after 2,000 second-task steps, canonical against all-correct targets, as the share of inputs with equivalent labels varies.

Inputs with equivalents	Canonical	All correct labels
0%	0.749	0.749
25%	0.727	0.740
50%	0.734	0.792
100%	0.722	0.870

The gap is zero when no input has equivalents, as it must be, and 15 points when every input does (Table 2). Conventions clash here too: when the second task shares the first task's tie-break the canonical encoding keeps 0.760, when it uses the opposite tie-break 0.667; a per-cluster random convention is unaffected (0.757

against 0.756), as is the all-correct encoding (0.729 against 0.753). With equivalents on every input and an opposing convention, the gap between canonical and all-correct targets is 19 to 28 points for 2, 3 or 5 equivalent labels and for hidden widths from 64 to 1,024 (Figure 3).

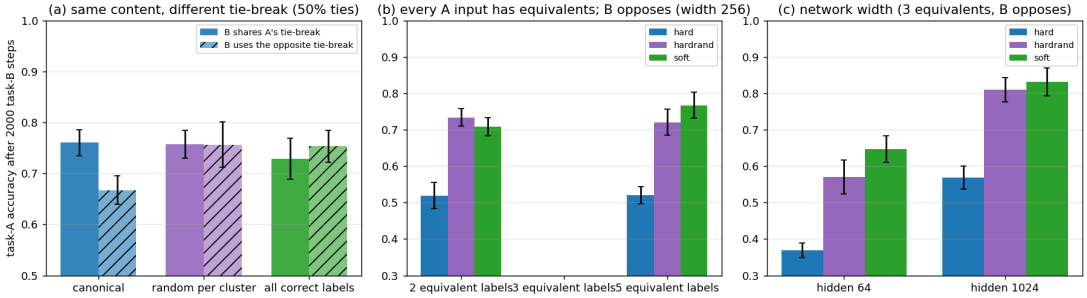


Figure 3: Synthetic model. (a) The canonical encoding keeps 0.76 when the next task shares its tie-break and 0.67 when it opposes it; random-per-cluster and all-correct encodings do not care. (b, c) With equivalents on every input and an opposing convention, the gap holds for 2, 3 or 5 equivalents and for narrow or wide networks.

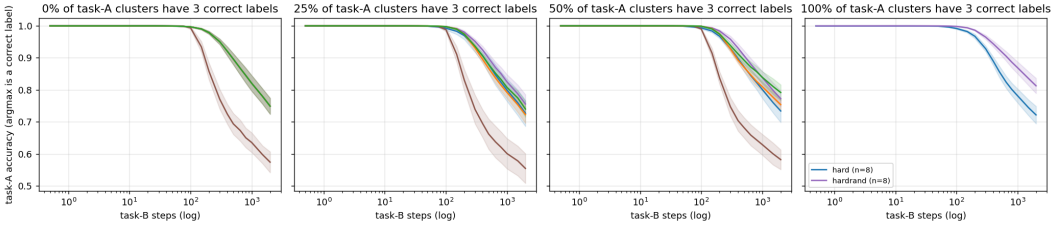


Figure 4: Synthetic model: the gap opens with the share of inputs that have equivalents (0, 25, 50 and 100%). Label smoothing (brown) is worse everywhere in this model.

5 Discussion

Mechanism. A canonical skill is confident at every tied state, and the next task's demonstrations, when they break ties differently, contradict it exactly there. This is the "confident conflict" of Diao et al. [4] seen from the old side, and in our setting the conflict is provably about convention, not content. A skill that is uncommitted where answers are interchangeable has no such conflicts to resolve, and so gains exactly when the conventions clash and not otherwise (Section 4.2). The convention is stored as a global preference rather than a per-state fact (Section 4.4), which is why overwriting it disturbs states with a unique optimal action as much as tied ones.

On-policy data. Retaining by Doing [2] argues that on-policy data, not the learning algorithm, explains why reinforcement learning forgets less. Our learner-preferred arm is the cleanest version of that data: optimal, and chosen to agree with the learner. It recovers the convention effect and nothing beyond it (Section 4.3). On-policy refinement of an already canonical skill does not undo its commitment (Section 4.5), so the benefit of on-policy data is in what it does not contradict, not in what it adds.

KL. The matched-KL comparison (Section 4.4) gives a controlled reason why KL predicts forgetting weakly in language models (correlation 0.52 in [2]): a large part of a canonical skill's measured drift is

movement among equally good answers, which costs nothing, and KL cannot tell that movement from the loss of optimal mass.

Language models. Most of what alignment tuning changes is style [16], and much of what fine-tuning removes is task alignment rather than knowledge [15]. Style is a convention: several phrasings, orderings and formats are equally correct, and a demonstration shows one. The mechanism measured here predicts that a skill taught from one canonical style will lose more under later fine-tuning with a different style than the same skill taught from a mixture of valid styles or from the learner's own preferred outputs, and that the confident-conflict tokens of [4] will be disproportionately convention tokens. We have not tested this.

Practical reading. When several answers are correct, train the skill you want to keep on all of them, or on the learner's own preferred ones. When fine-tuning later, prefer demonstrations that match the model's existing conventions.

6 Limitations

- Small networks (2.5 million parameters) and small grid worlds; the synthetic model is a multilayer perceptron. Whether the convention share of forgetting is as large in language models is untested here, though the mechanism (several equally correct outputs, demonstrations that pick one arbitrarily) is the normal case in supervised fine-tuning data.
- "Optimal" means fewest actions under the exact solver; demonstrations never contain mistakes, so the design isolates conventions from content, which real demonstration sets mix.
- Two tie-break conventions and one random-path mix were tested; conventions in real data are richer (style, ordering, formatting).
- One second-task budget (3,000 steps, learning rate 3×10^{-4}, fresh Adam with warm-up); the kept-Adam and higher-learning-rate follow-ups cover two alternatives only.
- Retention is measured on success within three times par under greedy decoding; the rankings are the same on memorised levels and under the per-level survival measure.
- Pre-registered prediction P1 held only in its qualified form (immunity to convention clash, not durability in general), which the data made clear.

7 Conclusion

With content held fixed by an exact solver, the arbitrary choices in a skill's demonstrations account for about a third of what later fine-tuning destroys under a clashing convention, and a skill taught from all equally good answers at once has no convention to lose. Prior work on the old model's state varied the optimizer, the geometry or the timing of training; this study shows that what the demonstrations commit to is a lever of its own, and it gives the confident-conflict and on-policy accounts of forgetting a controlled test from the other side. The next step is the same experiment in a language model, with style as the convention.

Reproducibility and status. Code, the pre-registered protocol, the full report and the literature check are in the author's research repository (`experiments/exact-oracle-forgetting`); the game engine is the public Long Game toolkit, v0.2.0 [24]. The whole study ran on one RTX 4090 for 4.8 hours ($4.24) plus about two CPU hours. This is a rough first draft prepared on November 2, 2026 from the experiment report; it has not been peer reviewed. Open items before submission: a language-model replication, more than two conventions, and a check that the confident-conflict tokens of entropy-adaptive fine-tuning are convention tokens.

References

[1] I. Shenfeld, J. Pari, and P. Agrawal. RL's Razor: Why online reinforcement learning forgets less. *Proceedings of ICLR*, 2026. arXiv:2509.04259.

[2] H. Chen, N. Razin, K. Narasimhan, and D. Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. *Proceedings of ICML*, 2026. arXiv:2510.18874.

[3] Self-distillation fine-tuning. arXiv:2601.19897, 2026.

[4] Diao et al. Entropy-adaptive fine-tuning: Resolving confident conflicts to mitigate forgetting. *Findings of EMNLP*, 2026. arXiv:2601.02151.

[5] S. I. Mirzadeh, M. Farajtabar, R. Pascanu, and H. Ghasemzadeh. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 2020. arXiv:2006.06958.

[6] S. I. Mirzadeh et al. Wide neural networks forget less catastrophically. *Proceedings of ICML*, 2022. arXiv:2110.11526.

[7] J. M. Springer et al. Overtrained language models are harder to fine-tune. *Proceedings of ICML*, 2025. arXiv:2503.19206.

[8] (How) learning rates regulate catastrophic overtraining. arXiv:2604.13627, 2026.

[9] Sharpness-aware pretraining mitigates catastrophic forgetting. *Proceedings of ICML*, 2026. arXiv:2605.02105.

[10] The grokked illusion: True equilibrium mitigates catastrophic forgetting. arXiv:2607.29503, 2026.

[11] Feng, Ghosal, Springer, Zhong, and Raghunathan. Early data exposure improves robustness to subsequent fine-tuning. *Conference on Language Modeling*, 2026. arXiv:2605.12705.

[12] S. Kessler, J. Parker-Holder, P. Ball, S. Zohren, and S. J. Roberts. Same state, different task: Continual reinforcement learning without interference. *Proceedings of AAAI*, 2022. arXiv:2106.02940.

[13] Interference and retention in continual learning. arXiv:2607.09202, 2026.

[14] S. Kotha, J. M. Springer, and A. Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. *Proceedings of ICLR*, 2024. arXiv:2309.10105.

[15] J. Zheng et al. Spurious forgetting in continual learning of language models. *Proceedings of ICLR*, 2025. arXiv:2501.13453.

[16] B. Y. Lin et al. The unlocking spell on base LLMs: Rethinking alignment via in-context learning. *Proceedings of ICLR*, 2024. arXiv:2312.01552.

[17] Z. Li et al. Preserving diversity in supervised fine-tuning of large language models. *Proceedings of ICLR*, 2025. arXiv:2408.16673.

[18] T. Xie et al. A unifying lens on supervised fine-tuning through target distribution design. arXiv:2606.11189, 2026.

[19] A. Shih et al. On the critical role of conventions in adaptive human-AI collaboration. *Proceedings of ICLR*, 2021. arXiv:2104.02871.

[20] L. Lehnert et al. Beyond A*: Better planning with transformers via search dynamics bootstrapping. arXiv:2402.14083, 2024.

[21] R. Müller, S. Kornblith, and G. Hinton. When does label smoothing help? *Advances in Neural Information Processing Systems*, 2019.

[22] Learning without overwriting: A theory of self-distillation and supervised fine-tuning in continual reasoning. arXiv:2610.05200, 2026.

[23] Reinforcement fine-tuning naturally mitigates forgetting in continual post-training. arXiv:2507.05386, 2025.

[24] C. Whitehead. Long Game: grid-world games with hidden rules for continual-learning research, v0.2.0. https://github.com/Nova-Rift/longgame, 2026.