TWO DEATHS OF BINARY REWARD: ARRIVAL AND ESCAPE IN FORMAL LANGUAGE TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Initial correctness does not determine how effectively binary-reward reinforcement learning adapts after obtaining a correct sample. We separate first arrival from subsequent escape using formal language tasks with exact verification and controlled output factorization. Factored policies with identical initial correctness, entropy, reward variance, and output-space cardinality exhibit a $49.5\times$ spread in median post-arrival escape time near the arrival boundary. An exact first-update calculation for no-baseline REINFORCE with plain SGD identifies dependence on factorization and loss normalization. Under isolated updates that leave the policy unchanged before the first correct sample, learning-rate changes preserve the arrival distribution. On 30 regex instances drawn uniformly from 132 buildable members of a 200-instance corpus draw, tripling the learning rate of a pretrained 0.5B model increases final success from 9/90 to 27/90 runs while preserving first-arrival episodes in all 41 arriving pairs. Shared-prompt updates allow learning-rate changes to affect arrival as well. In a 30-prompt setting with three shared training trajectories per arm, two rates both yield 0/39 final successes, while mean final accuracy differs from 0.010 to 0.156; a higher rate or longer continuation recovers success on a subset. At the tested probe depth, zero observed hits do not certify rare arrival during training. These results distinguish obtaining correct samples from converting them into successful learning, and identify sufficient update conditions under which intervention effects on the two stages separate.

1 INTRODUCTION

Reinforcement learning with verifiable rewards uses automatic checkers to convert generated outputs into training feedback. When a checker assigns reward only to an exactly correct output, learning depends on both obtaining correct samples and converting them into sustained improvements. A final success rate combines these stages and leaves the source of an unsuccessful run unresolved.

We distinguish *arrival*, the first training episode containing a positive reward, from *escape*, subsequent attainment of the stated success criterion. An unsuccessful run either never arrives within its budget or arrives without satisfying the final success criterion. These are observable outcomes relative to a policy, an update rule, and a training budget. Observing no correct sample does not establish that its probability is zero, and observing one does not establish that later updates will produce a successful policy.

Prior work establishes that weak reward variation can suppress the expected policy gradient (Razin et al., 2024), and that group-mean centering gives zero advantages to all-failure groups under binary rewards (Nie et al., 2026). We ask a more controlled question: when two configurations have the same initial chance of obtaining a correct sample, do they learn equally well after one appears? For binary reward, matching initial correctness also matches reward variance, so this comparison holds both aspects of reward scarcity fixed. Formal language tasks make that matching exact and support paired interventions that preserve first arrival while changing later updates. These empirical comparisons motivate reporting arrival and subsequent adaptation separately (Figure 1).

Contributions.

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.

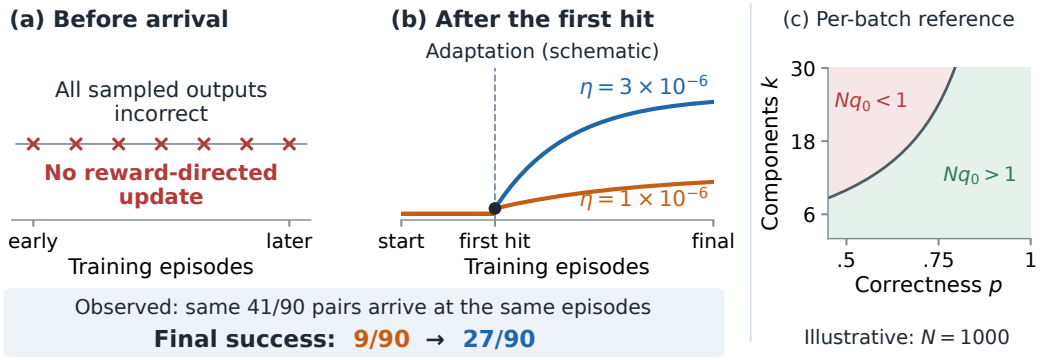


Figure 1: **Binary reward has two observable failure stages. (a)** All-failure groups give no reward-directed update under the paired experiment's centered, no-auxiliary update. **(b)** The same 41/90 pairs arrive at the same recorded episodes, yet final success changes from 9/90 to 27/90. Curves are schematic; colors link learning rates η to observed final-success counts. **(c)** The illustrative per-batch reference $Nq_0 = Np^k = 1$ uses $N = 1000$, independent components with common correctness p, and a unique target. Budget-level arrival depends on $H = ENq_0$; this is not a final-success boundary.

1. **Matched initial correctness leaves adaptation undetermined.** Formal language tasks provide exact verification and arrival calibration. Matched factored policies share initial correctness, entropy, binary-reward variance, and output-space cardinality, yet exhibit a $49.5\times$ spread in median post-arrival escape time near the arrival boundary (§5.1). A first-update calculation identifies factorization and loss-normalization dependence under no-baseline REINFORCE with plain SGD (§3.3, Appendix A.6).
2. **Learning-rate effects separate from arrival under policy stasis.** Changing the rate preserves the arrival distribution when all pre-arrival updates leave the policy unchanged (Remark 2). On a uniform draw of 30 buildable regex instances, first-arrival episodes match in all 41 arriving pairs while final success increases from 9/90 to 27/90 runs (§5.4). A larger budget increases arrival opportunity and permits subsequent updates (§5.2).
3. **Shared updates and finite measurements delimit the separation.** Shared-prompt updates allow the rate to change arrival. At 30 prompts per step, two rates yield 0/39 final successes but different mean final accuracies, 0.010 and 0.156 (§5.5). At the tested probe depth, zero hits do not certify rare arrival (§5.6).

2 Related Work

Sparse reward motivates exploration and reward shaping (Pathak et al., 2017; Ng et al., 1999), denser code feedback (Le et al., 2022; Liu et al., 2023b), and process supervision (Lightman et al., 2024). Our closest antecedents analyze optimization under sparse feedback and the relation between initial capability and subsequent training.

Reward variation and optimization. Razin et al. (2024) establish that small reward variation suppresses the expected policy gradient and demonstrate slow optimization even with access to expected gradients. Thus, optimization difficulty beyond insufficient exploration is already established. Our matched comparison fixes initial binary-reward variance as well as correctness and entropy, then measures the remaining variation in post-arrival adaptation under finite sampling.

Curriculum and capability. Huang et al. (2026) analyze transformer learning dynamics on compositional tasks, including weak gradients despite nonzero rewards and transfer across a difficulty spectrum. Our factored-policy calculation addresses a different controlled comparison, with explicit dependence on parameterization and loss normalization; the shared-prompt experiments test the scope of arrival invariance rather than prove a curriculum mechanism. Yue et al. (2025) compare base and RLVR-trained models through large-k pass@k, distinguishing coverage from sampling

efficiency. We track first arrival during training and paired changes after it, while treating a finite probe as an uncertain measurement of initial correctness.

Estimator and task scope. Group-mean centering gives zero advantages on all-failure groups; alternative advantages can yield nonzero updates there (Nie et al., 2026). The arrival-invariance statement therefore requires pre-arrival policy stasis, not binary reward alone. Related methods modify advantages, rollout allocation, or prompt selection (Appendix E). Formal languages supply exact checks and computable structure, complementing architectural studies and program synthesis (Delétang et al., 2023; Liu et al., 2023a; Butoi et al., 2025; Zhong et al., 2018). The elementary probability bounds calibrate these experiments.

3 Arrival Controls and Local Adaptation

3.1 Setup and Update Conditions

We use factored policies to control structure and local adaptation; the frozen-policy arrival law also applies to autoregressive outputs. Unsubscripted $\log$ denotes the natural logarithm; entropy in bits uses $\log_2$.

Definition 1 (Structured prediction with binary reward)**.** *Let $\mathcal{Y} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_k$ be a finite output space, $\phi : \mathcal{Y} \to \{0, 1\}$ a correctness predicate, and $\mathcal{Y}^* = \{y : \phi(y) = 1\}$ the correct set, with $M = |\mathcal{Y}^*| \geq 1$. For a factored policy $\pi_\theta(y) = \prod_{j=1}^{k} \pi_{\theta,j}(y_j)$, the objective is $J(\theta) = P_{\text{correct}} = \mathbb{E}[\phi(Y)]$. The no-baseline REINFORCE estimator from N independent samples is $\widehat{\nabla J} = N^{-1} \sum_{i=1}^{N} \phi(Y_i) \nabla_\theta \log \pi_\theta(Y_i)$.*

Initial correctness and arrival. Let $q_0 = \Pr_{Y \sim \pi_{\theta_0}}[\phi(Y) = 1]$ denote the probability that a complete output is correct. For E episodes with N independent samples per episode, define $B = EN$ and $H = Bq_0$. If the policy remains unchanged before the first positive reward, the probability of arrival within the budget is

$$\Pr(T_A \leq E) = 1 - (1 - q_0)^{EN} \approx 1 - \exp(-H).$$

Here H counts expected hits under the fixed initial policy; it does not count rewards accumulated after learning changes the policy. This law assumes a fixed correctness predicate as well as policy stasis. For a unique target under independent components with equal correctness probability p_0, $q_0 = p_0^k$. For other output conventions, q_0 includes every accepted output. Since binary reward is Bernoulli, its initial variance is $q_0(1 - q_0)$. Equal q_0 therefore matches this variance exactly, without matching the parameter gradients or their update geometry.

Remark 2 (Scope of pre-arrival stasis)**.** *No-baseline REINFORCE has zero reward gradient on an all-failure batch. Group-centered GRPO also has zero advantages on an all-failure group. These facts imply an unchanged policy only when the optimizer and auxiliary objectives introduce no parameter change. Our arrival-invariance statement applies to isolated-prompt updates that satisfy this condition. It does not cover arbitrary baselines, entropy terms, weight decay, or updates from other prompts. The GRPO KL term $r - \log r - 1$, with $r = \pi_{\text{ref}}/\pi_\theta$, has zero derivative at the reference; this preserves the stated condition only if the remaining update components also leave the policy unchanged.*

Thus $H = 1$ corresponds approximately to a 63.2% arrival probability and $H = 3$ to 95.0% under the frozen-policy model. A low H describes a rare event at a stated budget; it does not establish impossibility. Appendix A.8 gives the conditional gradient for a fixed baseline and separates zero reward gradients from zero parameter updates.

3.2 Multiplicity and Effective Complexity

Theorem 3 (Maximum-to-sum calibration bound)**.** *Let $p_{\max} = (\max_{y^* \in \mathcal{Y}^*} \pi_\theta(y^*))^{1/k}$. Then*

$$\boxed{p_{\max}^k \leq P_{\text{correct}} \leq M p_{\max}^k} \tag{1}$$

For any boundary policy θ^* satisfying $NP_{\text{correct}}(\theta^*) = 1$, define $p^* = p_{\max}(\theta^*)$. Then

$$\boxed{(NM)^{-1/k} \leq p^* \leq N^{-1/k}} \qquad \textit{(proof in Appendix A.1).} \tag{2}$$

The policy-specific boundary depends on the *effective multiplicity* $M_{\text{eff}}(\theta) = P_{\text{correct}}/p_{\max}^k \in [1, M]$: a policy concentrated on one canonical output has $M_{\text{eff}} \approx 1$; a symmetric policy over all relabelings has $M_{\text{eff}} \approx M$.

The bound is a maximum-to-sum inequality and holds for any joint policy. Its specialization to independent components makes P_{correct} computable in the controlled tasks. A complete minimal DFA with a fixed start-state label has $(m-1)!$ correct relabelings, provided the output convention admits all of them. This yields the optimistic multiplicity discount $M^{-1/k}$, while a policy concentrated on one labeling has $M_{\text{eff}} \approx 1$ and receives little discount. Only coordinates constrained by correctness enter an exact product calculation: unconstrained unreachable transitions sum out, and generated acceptance flags must also be counted (Appendix A). For heterogeneous components the product of their correctness probabilities replaces p_0^k.

3.3 Waiting Time and Post-Arrival Progress

Proposition 4 (Dead-zone trapping under zero-update failures)**.** *Consider no-baseline REINFORCE with batch size N and initial correctness probability $0 < P_0 \le Mc^{-k}$ for a constant $c > 1$. Assume that a batch without a correct sample leaves the policy unchanged, and that each batch containing a correct sample increases P_{correct} by at most a factor $1 + \varrho$, where $\varrho > 0$. Let $T_{1/N}$ be the first episode at which $P_{\text{correct}} \ge 1/N$. If $NP_0 \le \varrho/(1+\varrho)$, then*

$$\mathbb{E}[T_{1/N}] \ge \frac{c^k}{\varrho NM}.$$

Independent uniform components with $|\mathcal{C}_i| \ge c$ satisfy the initial bound; arity alone does not ensure it for a biased policy. The proposition concerns reaching probability $1/N$ under the stated updates. It does not predict the empirical time to the final success criterion. An estimated typical growth rate does not establish the uniform per-update upper bound assumed by the proposition. We therefore use the proposition to organize the waiting-time mechanism and report empirical post-arrival behavior separately.

Appendix A proves the bound and gives the simpler first-arrival lower bound $c^k/(NM)$. A complementary local calculation shows how adaptation can differ at equal q_0. For a uniform factored policy with a unique target, no-baseline REINFORCE, plain SGD, and h correct outputs in one batch, put $a = \eta h/N$. The first-update gain is

$$\frac{q_1}{q_0} = \left[\frac{c}{1 + (c-1)e^{-a}}\right]^k, \qquad q_0 = c^{-k}. \tag{3}$$

At fixed c^k, this gain increases with k for $a > 0$. The initial log-gain sensitivity is $k(1 - 1/c)$, the squared norm of the score of the correct output in these logit coordinates. Dividing the loss by k instead replaces a with a/k and changes the ordering in the numerical illustration (Appendix A.6). Thus initial correctness does not determine adaptation independently of the update convention. The calculation does not predict multi-step Adam or LLM escape times.

Additional scope. Appendices B–D give transition-width calculations and a binary-oracle query bound. That oracle result concerns an adversarial hidden correct set with binary-only access; it is not a computational lower bound for solving an explicitly given automaton. A fitted dependence factor for autoregressive policies is descriptive and need not exceed one (Appendix C).

4 Finite Automata as a Controlled Testbed

4.1 Tasks, Probabilities, and Outcome Measures

DFA equivalence, minimization, and product construction provide exact verification and controllable output structure. Appendix I specifies output conventions, parameterizations, seeds, and budgets; Appendix H defines the reward controls. We distinguish analytical arrival quantities from measured learning outcomes. The fixed-policy quantity $H = ENq_0$ concerns first arrival. Final accuracy also depends on updates after arrival, so a learning-success curve need not follow $1 - e^{-H}$.

Outcome definitions. We report the first positive-reward episode as T_A. For the LLM experiments, final success means empirical joint accuracy of at least 0.9 over 64 post-training samples, requiring at least 58/64 hits. The first training group whose empirical accuracy reaches 0.9 is a separate crossing-time statistic. It is affected by group size and sampling variation and does not establish that the final policy satisfies the success criterion. The factored-policy convergence thresholds and the theoretical threshold $P_{\text{correct}} = 1/N$ are stated separately.

Algorithm 1 records arrival and final outcomes; Appendix I.17 maps the main comparisons to their controls and remaining alternative explanations.

4.2 Structural Calibration and Corpus Coverage

DFA equivalence, minimization, and product experiments assess which structural probabilities track the measured boundary (Appendix I). In minimization the observed boundary correlates with the single-labeling boundary p^*_{alive} at 0.994; the optimistic factorial multiplicity discount is not attained. This distinguishes the combinatorial count M from the policy's effective multiplicity M_{eff}.

On 100 regex-benchmark DFAs, the from-scratch criterion labels 45 instances high-arrival ($H \geq 3$) and 55 low-arrival ($H < 0.3$). Binary reward achieves the experiment's success criterion on 225/225 and 1/275 runs, respectively. However, 41 of the 45 high-arrival instances have one state. A preregistered, disjoint redraw of 45 high-arrival instances from the 348 non-degenerate candidates again gives 225/225 successful runs, with first-correct episodes at a median ratio of 1.00 to $1/(Nq_0)$. On these corpus draws H remains confounded with state count, so they do not establish an advantage over nominal size as a predictor. A ten-instance transition band preserves the predicted ordering (Spearman $\rho = 0.81$), but its final-success curve is sharper and displaced relative to the arrival approximation (Table 5). Across 20,901 parsed corpus instances, 74% have low H under the specified from-scratch policy at $E = 5{,}000$, $N = 256$. This percentage does not describe failure prevalence for pretrained LLMs.

5 Matched Arrival and Post-Arrival Adaptation

5.1 Exactly Matched Initial Correctness

We match initial joint correctness and output-space cardinality while varying the factorization (Figure 2). For the R24 ladder, six pairs $(k, c) \in \{(3, 256), (4, 64), (6, 16), (8, 8), (12, 4), (24, 2)\}$ all have $c^k = 2^{24}$, initial entropy 24 bits, $q_0 = 2^{-24}$, and binary-reward variance $2^{-24}(1 - 2^{-24})$. Entropy and variance are consequences of the uniform distribution and matched q_0, rather than additional independent matching conditions. At $N = 256$ and $E = 40{,}000$, every rung therefore has $H = 0.610352$ and the same theoretical arrival probability, approximately 45.7%. The comparison changes k and c together at fixed initial entropy; it does not isolate k while holding component arity fixed.

Across 100 seeds per rung, the observed arrival fraction has no monotone trend (Spearman $+0.38$), while median post-arrival escape time falls $49.5\times$ (Spearman -1.000; log-rank trend $z = +18.2$). Two base-3 ladders yield $35.5\times$ and $47.1\times$ spreads. The large ratios occur near the boundary; crossed-grid controls with arrival-matched optimizer ages show smaller interior effects, 1.0–$1.9\times$ (Appendix I.10). The escape-time trend also occurs under plain SGD (log–log slope -1.57, versus -1.88 under Adam), so it is not exclusive to Adam. These controls support a factorization-dependent adaptation result under the tested update conventions; they do not isolate k from c or establish invariance to loss normalization. Equation 3 supplies a local analytical illustration, while the multi-step magnitudes remain empirical observations.

5.2 Learning Rate and Budget Under Isolated Updates

Under the pre-arrival stasis condition of Remark 2, the learning rate cannot change arrival: the policy and sampling distribution remain the same until the first positive reward. In 2,400 paired factored-policy runs with a shared random stream and different rates, no first-arrival episode differs. After arrival, tripling the rate increases success conditional on arrival from 69.6% to 97.4%, recovering

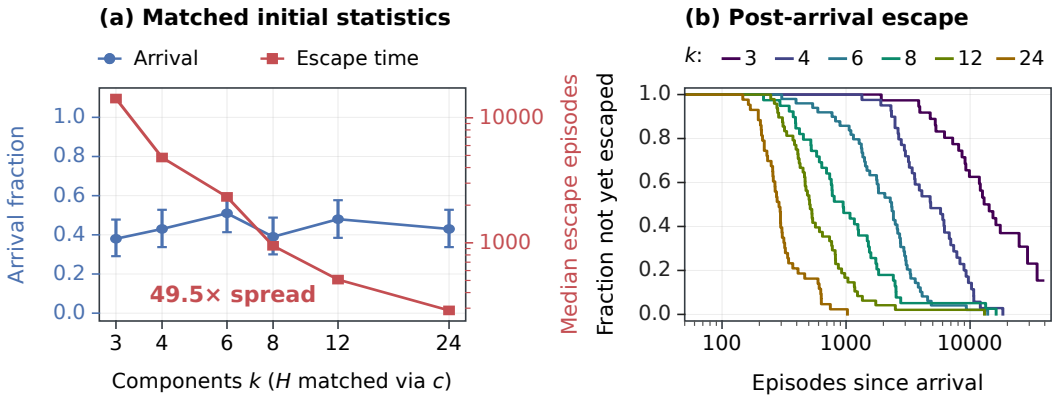


Figure 2: **Matched arrival probabilities and different post-arrival dynamics.** The R24 ladder holds $c^k = 2^{24}$ and $q_0 = 2^{-24}$ fixed while varying both k and c (100 seeds per rung). At $E = 40{,}000$, $H = 0.610352$ throughout; the base-budget value is 0.076294. Panel (a) reports arrival fractions with Wilson intervals and median post-arrival escape times; panel (b) reports Kaplan–Meier curves with budget censoring. The $49.5\times$ escape-time spread describes this boundary setting, not a universal complexity law.

approximately 91% of the available headroom (pre-registered, disjoint seed stream; reported $p = 7.5 \times 10^{-28}$).

A larger budget acts on both stages. Pooled over the ladder grid of Appendix I.10, whose cells span four orders of magnitude in H rather than the single matched value of Figure 2, an $8\times$ budget lifts arrival from 30.7% to 49.3%, where the rate arm leaves it at 30.7% against 30.2%, and the budget lifts escape as well where escape has room. The arrival-dominated denominator can attenuate a rate effect in unconditional final success: the same rate arm gives $+0.9$ percentage points overall ($p = 0.60$), compared with $+4.6$ conditional on arrival ($p = 3.9 \times 10^{-5}$). These contrasts motivate reporting both quantities (Appendix I.10). They do not imply that raising the learning rate always helps: on a 400K-parameter autoregressive control 10^{-3} succeeds in $3/3$ seeds while 3×10^{-3}, 10^{-2} and 3×10^{-2} succeed in none, and arrival is unchanged across all four rates, so the reversal is entirely in escape (Appendix I.9).

5.3 A Pretrained Model on Verified Generation Tasks

The isolated LLM experiments start each instance from a fresh checkpoint. Qwen2.5-0.5B-Instruct is trained with GRPO on position-constrained generation and regex generation. Success means final joint accuracy at least 0.9 over 64 post-training samples. In the position-constrained study (108 runs; Figure 3), all ten instances with probe estimate $\widehat{H} \geq 14$ achieve 3/3 binary-reward successes, and all six zero-hit instances achieve 0/3. The latter observation is compatible with low initial correctness but does not certify it. Across 128 binary GRPO runs pooled from the position-constrained studies, the regex grid, and the Qwen3-8B cell, 113 arrive, of which 19 fail the final criterion; 15 never arrive (Appendix I.11).

On selected regex instances, 75 paired runs have identical first-arrival episodes across rates. A fourfold budget increase changes arrival on 14 independent streams, while tripling the rate reduces the reported Kaplan–Meier escape time from 117 to 16 steps (stratified paired log-rank $p = 1.5 \times 10^{-8}$), recovering both selected trapped instances (Appendix I.16). These experiments motivate the following comparison on a uniform draw.

5.4 A Uniform Draw of Buildable Regex Targets

We drew 30 instances uniformly without replacement from the 132 buildable targets and trained each instance from a fresh checkpoint at two learning rates with three seeds per rate. Of the 90 runs in each arm, 41 arrived. The first positive-reward episode matched in every arriving pair, while final

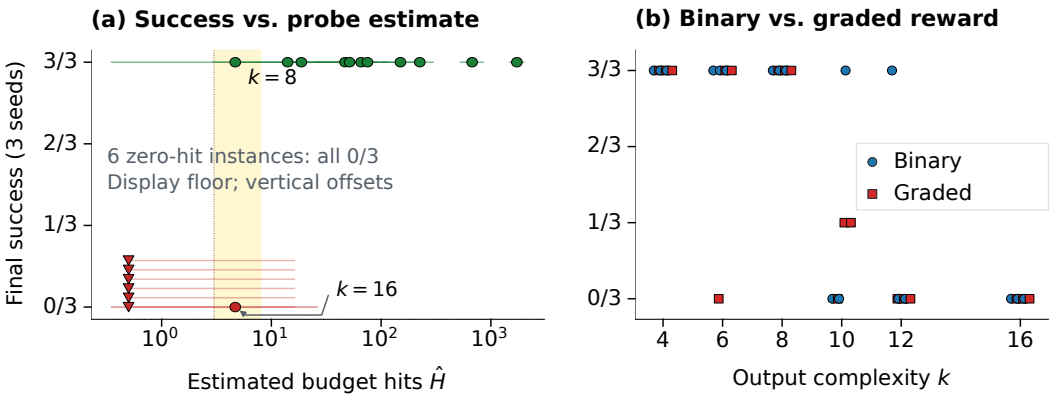


Figure 3: **Pretrained-model outcomes on 18 position-constrained instances.** GRPO on Qwen2.5-0.5B-Instruct, three training seeds per reward condition. **(a)** Binary success versus $\hat{H} = EG\hat{P}_0$ from pooled probes (up to 4,096 samples; 95% Poisson CIs; triangles denote zero-hit probes stacked at a display floor). **(b)** Success by k under binary and sequence-level graded rewards. Success requires at least 58/64 correct final samples. The association describes this grid; zero probe hits do not certify low arrival (§5.6).

success increased from 9/90 at $\eta = 10^{-6}$ to 27/90 at $\eta = 3 \times 10^{-6}$. Success conditional on arrival therefore increased from 22.0% to 65.9%. All nine instances with discordant success counts favored the higher rate, with the reported one-sided instance-level sign test giving $p = 0.002$. All-failure groups skip the optimizer step, preserving the pre-arrival policy. Conditioning on recorded arrival compares the same 41 runs, so it does not select different cohorts after the rate change. The 20-percentage-point overall gain is exactly $(41/90)[27/41 - 9/41] = 18/90$ in the reported outcomes. This locates the observed rate effect after arrival without estimating the initial probability. Post-hoc cutoffs of 56–59/64 preserve the same 9/90 versus 27/90 success counts (Appendix G).

Table 1: Arrival and final success on the uniform draw of 30 buildable regex instances, with three seeds per instance and rate. Counts refer to runs. The same 41 paired runs arrive in both arms, while final success changes after arrival.

Outcome	$\eta = 10^{-6}$	$\eta = 3 \times 10^{-6}$
Never arrived	49/90	49/90
Arrived, final criterion unmet	32/90	14/90
Final criterion met	9/90	27/90
Success given arrival	22.0%	65.9%

Each arm uses GRPO with $G = 64$, $E = 300$, and $\beta = 0$. The draw proceeds from 20,901 corpus entries to 200 uniformly sampled targets, 132 buildable targets, and finally 30 uniformly sampled targets. Coverage is conditional on buildability. Verification conventions and sensitivity are reported in Appendix I.14. At the instance level, the never-arrived/trapped/successful split changes from $14/13/3$ to $14/4/12$. The table uses run counts to avoid mixing these denominators. Two registered predictions fail: never-arrived instances are a plurality rather than the predicted majority, and three zero-hit-probe instances succeed at the higher rate (Appendix I.14). These outcomes limit prospective probe-based prediction while preserving the observed paired comparison.

5.5 Shared Updates and the KL Estimator

Updates from another prompt change a shared policy even when the focal prompt has no positive reward, removing the pre-arrival stasis premise. With two prompts per step and $\beta = 0$, tripling the rate increases final success of the six selected trapped prompts from 8/36 to 24/36 outcomes across two partners and three seeds. Five of six instances improve and none worsens (reported instance-level sign $p = 0.031$). Arrival also changes, with some trajectories arriving earlier and others losing arrival entirely; the registered general suppression prediction fails.

The KL implementation matters as well. An auxiliary sampled $\log \pi_\theta - \log \pi_{\text{ref}}$ term produces nonzero realized gradients on all-failure batches in this implementation and disrupts isolated arrival. With the $r - \log r - 1$ estimator at $\beta = 0.02$, first signals match in 18/18 isolated pairs. This timing agreement does not establish exact parameter stasis when optimizer steps still execute. In two-prompt training with that estimator, final success changes from 0/18 to 9/18 outcomes, with three of six instances improving and none worsening (instance-level $p = 0.125$). The stronger seed-pair result does not replace this limited instance-level evidence (Appendix I.15).

Shared-prompt training. At eight prompts per step, final success increases from 0/18 to 9/18 prompt–seed outcomes at $G = 16$, and to 11/18 at $G = 64$. No instance has a lower success count at the higher rate. The six-instance tests do not establish whether the learning-rate effect is preserved in magnitude.

At 30 prompts per step, the 13 prompts classified as trapped under isolated training contribute 39 prompt–seed outcomes from three shared training seeds (Figure 4). With $G = 16$, $\beta = 0.02$, and the GRPO KL estimator, $\eta = 10^{-6}$ and 3×10^{-6} both yield 0/39 final successes after 300 steps, while mean final joint accuracy increases from 0.010 to 0.156. At $\eta = 9 \times 10^{-6}$, 5, 6, and 7 prompts succeed in the three seeds; continuing the 3×10^{-6} trajectories to 900 steps yields 3, 6, and 4. The zero-success count therefore hides differences in final accuracy, and both interventions recover success on a subset: 7 of the 13 instances put a run past the bar at 9×10^{-6}, all 13 are monotone in the rate by seed median, and five instances stay under 0.10 in every run of every dose. These comparisons describe three shared policy trajectories per arm; the longer arm continues the same trajectories. They do not isolate prompt-count, group-size, or batch-composition effects. The higher-rate arm also changes arrival from 32/39 to 29/39. Arrival invariance holds under the pre-arrival stasis conditions in Remark 2, and that condition is testable rather than definitional: on the same 18 isolated pairs, a sampled $\log \pi_\theta - \log \pi_{\text{ref}}$ term at $\beta = 0.02$ cuts arrival to $14/18$ and then $6/18$ as the rate triples, while the $r - \log r - 1$ estimator leaves it at $18/18$ at both rates (Appendix I.15).

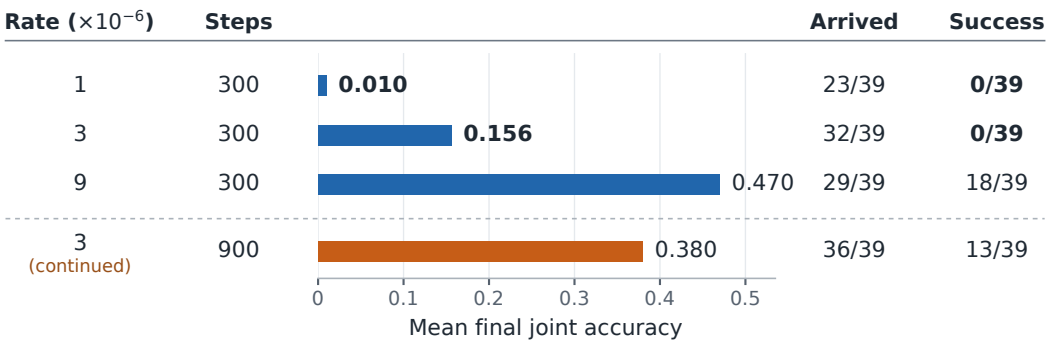


Figure 4: **Final accuracy reveals progress below the success threshold.** Thirty prompts share each update ($G = 16$, $\beta = 0.02$, GRPO KL estimator). Bars summarize the 13 prompts trapped under isolated training (three shared trajectories; 39 coupled prompt–seed outcomes per arm). Success denotes evaluated joint accuracy ≥ 0.9 per outcome. The 900-step condition continues the 3×10^{-6} trajectories; comparisons are descriptive.

Five base-rate prompt–seed outcomes also cross the noisy training-group threshold before step 300 but finish below the final criterion. These training-threshold crossings are distinct from the final performance reported in Figure 4.

5.6 The Resolution of an Initial-Policy Probe

An initial-policy probe estimates total correctness as $\widehat{q}_0 = h/n$, already including all accepted outputs. Zero hits among $n = 2{,}048$ samples give the 95% upper bound $H_U = 28.1$ for $B = 19{,}200$, which does not certify rare arrival. On the uniform draw, 19 of 30 instances have zero probe hits; three of them arrive and satisfy the final criterion at the higher rate. At 8B the same frozen rule called

one instance dead from zero probe hits, and training refuted it in 2 of 3 seeds (Appendix I.11). These observations support reporting probe counts and uncertainty, without treating either zero or positive hits as a learning guarantee. Certifying $H < 3$ from a zero-hit probe at this budget would need $19{,}172$ samples and $H < 1$ would need $57{,}517$, against the $2{,}048$ drawn here. Appendix G gives the exact bound; Appendix I.14 reports the instance-level associations. The larger 132-target probe and its representation-dependent comparison with from-scratch probabilities are in Appendix I.13.

Partial credit and final correctness. On position-constrained generation, graded reward succeeds in 26/54 runs versus 33/54 under binary reward (Figure 3, right). Near-miss trajectories concentrate probability on incorrect completions. The contrast with factored-policy controls does not isolate reward decomposition because policy, initialization, and optimization also differ. Both our dense factored and graded LLM controls use scalar fraction-correct rewards. VeRPO uses weighted per-test aggregation and local and global advantages (Wang et al., 2026a); it is not a matched control (Appendix E).

6 Discussion and Conclusion

Practical interpretation. The comparisons identify three distinct questions for a training report: did a correct sample appear, did the subsequent policy improve, and did its final accuracy meet the chosen criterion? Under pre-arrival stasis, a learning-rate change leaves the first question unchanged; the paired regex comparison identifies its effect on the second stage. Shared updates require reporting changed arrival alongside final accuracy because the arriving cohort can also change. Continuous accuracy prevents a common threshold from concealing different amounts of progress. This is an experimentally supported measurement strategy; an automatic intervention-selection policy and its compute advantage remain untested.

Limitations. Exact structural probabilities require the stated finite output conventions. The matched-factorization result concerns factored policies, and its largest effects occur near the boundary. The central regex interventions use a 0.5B Qwen model; auxiliary 8B and autoregressive studies do not establish generality of that intervention result across model scales or tasks. The uniform regex draw is restricted to buildable targets. Shared-prompt comparisons have only three independent training seeds per arm and confound prompt count, group size, and composition. Finite probes and 64-sample final evaluations have limited resolution; parse validity and eligibility must accompany arrival and final-outcome reporting. The pre-registration ledger retains all 75 decided predictions, including 29 failures, and distinguishes registered tests from the interpretations supported here (Appendix I.12).

Conclusion. Matched initial correctness does not determine post-arrival adaptation in binary-reward learning. Formal language tasks expose this separation through exact verification and controlled factorization, with the largest measured escape-time differences near the arrival boundary. Under pre-arrival policy stasis, learning-rate changes preserve arrival, and paired regex experiments show substantial changes in final success at identical first-arrival episodes. Shared updates allow arrival itself to change, while continuous final accuracy reveals differences hidden by equal success counts. These findings make the update rule and training budget essential to interpreting reward scarcity and final failure. Reporting arrival, subsequent adaptation, and final accuracy separately exposes which stage an observed shortfall concerns. Extending these comparisons to updates that change the policy before arrival and to broader reasoning tasks remains open.

Reproducibility Statement

Appendix A states the assumptions and proofs, including the one-step calculation in Appendix A.6. Algorithm 1 specifies arrival and final-outcome reporting. Appendices I–I.16 give experimental settings, seeds, budgets, learning rates, tables, and curves; Appendix I.12 records decided predictions and failures. The reported checkpoint-replay nondeterminism is disclosed in Appendix I. Code, DFA corpora, probe calls, pre-registration records, and run logs are planned for public release on acceptance.

AI USE STATEMENT

Generative AI tools assisted conceptual analysis, auxiliary derivations and proof exposition, methodological feedback, experiment and analysis code, result interpretation, and manuscript editing and literature triage. The analytical assistance included the local first-update calculation in Appendix A.6. ChatGPT/Codex and Claude assisted these revisions and the accompanying analysis re-runs; Google's Paper Assistant Tool (PAT) supplied manuscript feedback. The authors take responsibility for the final text, claims, code, references, and reported results.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Dana Angluin. A note on the number of queries needed to identify regular languages. *Information and Control*, 51(1):76–87, 1981.

Dana Angluin. Negative results for equivalence queries. *Machine Learning*, 5:121–150, 1990.

Richard E. Barlow and Frank Proschan. *Statistical Theory of Reliability and Life Testing: Probability Models*. Holt, Rinehart and Winston, 1975.

Andrei Baroian and Rutger Berger. Prompt replay: Speeding up grpo with on-policy reuse of high-signal prompts. *arXiv preprint arXiv:2603.21177*, 2026.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pp. 41–48, 2009.

Jean Bourgain, Jeff Kahn, Gil Kalai, Yitzhak Katznelson, and Nathan Linial. The influence of variables in product spaces. *Israel Journal of Mathematics*, 77:55–64, 1992.

Alexandra Butoi, Ghazal Khalighinejad, Anej Svete, Josef Valvoda, Ryan Cotterell, and Brian DuSell. Training neural networks as recognizers of formal languages. In *International Conference on Learning Representations (ICLR)*, 2025.

Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Where the really hard problems are. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 331–337, 1991.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. Neural networks and the chomsky hierarchy. In *International Conference on Learning Representations (ICLR)*, 2023.

Marc Dymetman. Binary rewards and reinforcement learning: Fundamental challenges. *arXiv preprint arXiv:2605.02375*, 2026.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nature*, 590:580–586, 2021.

Ehud Friedgut. Sharp thresholds of graph properties, and the k-sat problem. *Journal of the American Mathematical Society*, 12(4):1017–1054, 1999.

Abhishek Gupta, Aldo Pacchiano, Yuexiang Zhai, Sham Kakade, and Sergey Levine. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15281–15295. Curran Associates, Inc., 2022.

Yu Huang, Zixin Wen, Yuejie Chi, Yuting Wei, Aarti Singh, Yingbin Liang, and Yuxin Chen. On the emergence of implicit curriculum in RLVR learning dynamics. In *International Conference on Machine Learning (ICML)*, 2026.

Sham Machandranath Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, University College London, 2003.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. PAC reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven C. H. Hoi. CodeRL: Mastering code generation through pretrained models and deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21314–21328. Curran Associates, Inc., 2022.

Thanh-Long V. Le, Myeongho Jeon, Kim Vu, Viet Lai, and Eunho Yang. No prompt left behind: Exploiting zero-variance prompts in llm reinforcement learning via entropy-guided advantage shaping. In *International Conference on Learning Representations (ICLR)*, 2026.

Andrew C. Li, Zizhao Chen, Toryn Q. Klassen, Pashootan Vaezipoor, Rodrigo Toro Icarte, and Sheila A. McIlraith. Reward machines for deep RL in noisy and uncertain environments. In *Advances in Neural Information Processing Systems*, volume 37, pp. 110341–110368. Curran Associates, Inc., 2024.

Gen Li, Yuting Wei, Yuejie Chi, and Yuxin Chen. Softmax policy gradient methods can take exponential time to converge. *Mathematical Programming*, 201(1–2):707–802, 2023.

Xin-Ye Li, Ren-Biao Liu, Yun-Ji Zhang, Hui Sun, Zheng Xie, and Ming Li. Exploring pass-rate reward in reinforcement learning for code generation. *arXiv preprint arXiv:2605.02944*, 2026.

Shuang Liang, Haoyang Zhou, Yifan Gong, Guowei Wang, and Xiting Wang. Leeps: Latent-guided explore-exploit prompt sampling for efficient rlvr in large language models. *arXiv preprint arXiv:2607.28077*, 2026.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *International Conference on Learning Representations (ICLR)*, 2023a.

Jiate Liu, Yiqin Zhu, Kaiwen Xiao, Qiang Fu, Xiao Han, Wei Yang, and Deheng Ye. RLTF: Reinforcement learning from unit test feedback. *Transactions on Machine Learning Research*, 2023b.

Nicholas Locascio, Karthik Narasimhan, Eduardo DeLeon, Nate Kushman, and Regina Barzilay. Neural generation of regular expressions from natural language with minimal domain knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1918–1923, 2016.

Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

Rémi Monasson, Riccardo Zecchina, Scott Kirkpatrick, Bart Selman, and Lidror Troyansky. Determining computational complexity from characteristic "phase transitions". *Nature*, 400(6740): 133–137, 1999.

Alexander Mordvintsev. Differentiable finite state machines. *Google Research Blog*, 2022.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning (ICML)*, pp. 278–287, 1999.

Wenhua Nie, Jianan Wu, Junlin Liu, Ziwei Li, Zheng Lin, Zhang Zijian, Yilong Fan, Haoran Zheng, and Jyh-Shing Roger Jang. Gradient starvation in binary-reward GRPO: Why group-mean centering fails and why the simplest fix works. *arXiv preprint arXiv:2605.07689*, 2026.

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

Yujuan Pang, Jiaxin Li, Xin Sheng, Ran Peng, and Yong Ma. Beyond variance: Prompt-efficient rlvr via rare-event amplification and bidirectional pairing. *arXiv preprint arXiv:2602.03452*, 2026.

Jun-U Park, Sang-Ki Ko, Marco Cognetta, and Yo-Sub Han. SoftRegex: Generating regex from natural language descriptions using softened regex equivalence. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6425–6431, 2019.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pp. 2778–2787. PMLR, 2017.

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1256–1272. Curran Associates, Inc., 2021.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Ali Rad, Khashayar Filom, Darioush Keivan, Peyman Mohajerin Esfahani, and Ehsan Kamalinejad. Rate or fate? RLV$^{\varepsilon}$R: Reinforcement learning with verifiable noisy rewards. In *International Conference on Machine Learning (ICML)*, 2026.

Noam Razin, Hattie Zhou, Omid Saremi, Vimal Thilak, Arwen Bradley, Preetum Nakkiran, Joshua Susskind, and Etai Littwin. Vanishing gradients in reinforcement finetuning of language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems*, volume 36, pp. 55565–55581. Curran Associates, Inc., 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Michel Talagrand. On Russo's approximate zero-one law. *Annals of Probability*, 22(3):1576–1587, 1994.

Rodrigo Toro Icarte, Toryn Q. Klassen, Richard Valenzano, and Sheila A. McIlraith. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73:173–208, 2022.

Elena Umili and Roberto Capobianco. DeepDFA: Automata learning through neural probabilistic relaxations. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI)*, pp. 1051–1058. IOS Press, 2024.

Longwen Wang, Yirui Liu, Xuan'er Wu, Xiaohui Hu, Yuankai Fan, Kaidong Yu, Qizhen Weng, Wei Xi, and Xuelong Li. Beyond binary: Turning partial success into dense verifiable rewards for reinforcement learning in code generation. *arXiv preprint arXiv:2601.03525*, 2026a.

Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Ellie Evans, Daniel Egert, Hoo-Chang Shin, Felipe Soares, Yi Dong, and Oleksii Kuchaiev. RLBFF: Binary flexible feedback to bridge between human feedback and verifiable rewards. In *International Conference on Learning Representations (ICLR)*, 2026b.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Xin Xu, Clive Bai, Kai Yang, Tianhao Chen, Yangkun Chen, Weijie Liu, Hao Chen, Yang Wang, Saiyong Yang, and Can Yang. Composition-rl: Compose your verifiable prompts for reinforcement learning of large language models. *arXiv preprint arXiv:2602.12036*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775.

Song Yu, Li Li, Wenwen Zhao, and Zhisheng Yang. EP-GRPO: Entropy-progress aligned group relative policy optimization with implicit process guidance. *arXiv preprint arXiv:2605.04960*, 2026.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, volume 38, pp. 57654–57689. Curran Associates, Inc., 2025.

Zexuan Zhong, Jiaqi Guo, Wei Yang, Jian Peng, Tao Xie, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. SemRegex: A semantics-based approach for generating regular expressions from natural language specifications. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1608–1618, 2018.

Tianshu Zhu, Wenyu Zhang, Xiaoying Zuo, Lun Tian, Haotian Zhao, Yucheng Zeng, Jingnan Gu, Daxiang Dong, Jianmin Wu, Dawei Yin, and Dou Shen. Rollout pass-rate control: Steering binary-reward RL toward its most informative regime. *arXiv preprint arXiv:2605.05112*, 2026.

A PROOFS

A.1 PROOF OF THEOREM 3

Proof. The correct outputs $\mathcal{Y}^* = \{y_1^*, \ldots, y_M^*\}$ are disjoint elements, so $P_{\text{correct}} = \sum_{j=1}^{M} \pi_\theta(y_j^*)$. The lower bound is $P_{\text{correct}} \geq \max_j \pi_\theta(y_j^*) = p_{\max}^k$. The upper bound is $P_{\text{correct}} \leq M \max_j \pi_\theta(y_j^*) = M p_{\max}^k$. Substituting into $N P_{\text{correct}} = 1$ yields the interval. □

A.2 PROOF OF PROPOSITION 9 (TRANSITION SHARPNESS)

Under $P_{\text{correct}} = Cp^k$, the probability of at least one correct sample in N trials is $q = 1 - (1 - Cp^k)^N$. Solving for p:

$$p_q = \left(\frac{1 - (1-q)^{1/N}}{C} \right)^{1/k}. \tag{4}$$

When Cp^k is small, the Poisson approximation gives $q \approx 1 - e^{-NCp^k}$, hence $p_q \approx p^*(-\ln(1-q))^{1/k}$ where $p^* = (NC)^{-1/k}$. Using $a^{1/k} = 1 + (\ln a)/k + O(k^{-2})$:

$$\Delta p = p_{1-\epsilon} - p_\epsilon \approx \frac{p^*}{k} \ln\left(\frac{-\ln \epsilon}{-\ln(1-\epsilon)} \right). \tag{5}$$

A.3 PROOF OF COROLLARY 5 (DFA MINIMIZATION)

Corollary 5 (Multiplicity under a stated DFA encoding)**.** *For complete minimal DFAs with m states, a fixed start-state label, and an output convention admitting every relabeling, $M = (m-1)!$. With $k(m) = dm$ modeled coordinates for a fixed $d > 0$, the optimistic boundary in the maximum-output-probability coordinate is $p^*_{\text{dead}}(m) = (N(m-1)!)^{-1/k(m)}$. Here $d = s$ for a transition-coordinate illustration and $d = s + 1$ when one acceptance flag per state is also a modeled coordinate. Its continuous extension is non-monotone and peaks at $m^* \approx \ln N$ when the asymptotic stationary point lies in the admissible range. This is a bound on the optimistic boundary, not a statement that a policy realizes all relabelings equally.*

Multiplicity. The Myhill–Nerode theorem gives uniqueness of the complete minimal DFA up to isomorphism. A start-preserving automorphism is the identity: every state is reached by some word w, and an automorphism must map its state to the state reached by the same w. The $(m-1)!$ start-preserving relabelings are therefore distinct. Their acceptance flags are relabeled along with their transitions. A restricted output convention, such as fixed accepting-state labels, requires recounting the admitted correct outputs.

Boundary coordinate. The root probability $p_{\max} = (\max_{y^*} \pi_\theta(y^*))^{1/k(m)}$ is not an arithmetic mean of component accuracies. Theorem 3 supplies the bound with the full-output M. Plots using $k = sm$ illustrate transition-coordinate scaling; if acceptance flags are generated, their probability mass must enter P_{correct} separately or be included in k.

Non-monotonicity. Writing $\log p^*_{\text{dead}}(m) = -(\log N + \log \Gamma(m))/(dm)$, the stationary condition is $m\psi(m) - \log \Gamma(m) = \log N$. Stirling's approximation gives $m^* \approx \ln N$; the fixed coordinate factor d cancels.

A.4 PROOF OF REMARK 10 (LE CAM LOWER BOUND)

We use Le Cam's two-point method with a pre-hit coupling argument. The placement of the two correct sets must be *adversarial to the algorithm*: for fixed sets, an instance-aware algorithm could query an element of $\mathcal{Y}_0^*$ first and distinguish the instances in one query.

Null execution and placement. Fix any (possibly randomized) algorithm and run it against the *all-zero-feedback oracle* ($r_t = 0$ for every query); this null execution is well-defined independently of any instance. Let $q(y) = \Pr[\text{null execution queries } y \text{ within its first } Q \text{ queries}]$; then $\sum_{y \in \mathcal{Y}} q(y) \leq Q$. Order $\mathcal{Y}$ by increasing q and let S be the first $2M$ elements; by averaging, $\sum_{y \in S} q(y) \leq$

$2MQ/|\mathcal{Y}| \le 2MQ/c^k$. Partition S into disjoint $\mathcal{Y}_0^*, \mathcal{Y}_1^*$ with $|\mathcal{Y}_0^*| = |\mathcal{Y}_1^*| = M$, and let P_0 (resp. P_1) denote the observation distribution when $\mathcal{Y}^* = \mathcal{Y}_0^*$ (resp. $\mathcal{Y}_1^*$).

Coupling. Define the first-hit time $\tau = \inf\{t : y_t \in \mathcal{Y}_0^* \cup \mathcal{Y}_1^*\}$. For $t < \tau$, feedback is $r_t = 0$ under both instances *and* under the null execution, so all three processes can be coupled to issue identical queries up to τ. A maximal coupling gives $D_{\mathrm{TV}}(P_0, P_1) \le \Pr[\tau \le Q]$.

Hit probability. Up to τ the coupled process is the null execution, so $\Pr[\tau \le Q] = \Pr[\text{null execution hits } S] \le \sum_{y \in S} q(y) \le 2MQ/c^k$ by the placement of S.

Le Cam's inequality. $\inf_{\mathrm{ALG}} \sup_{\mathcal{Y}^*} \Pr[\hat{y} \notin \mathcal{Y}^*] \ge \frac{1}{2}(1 - D_{\mathrm{TV}}) \ge \frac{1}{2}(1 - 2MQ/c^k)$. Setting this $\le \delta$ yields $Q \ge (1 - 2\delta)c^k/(2M)$. □

Binary versus per-component feedback. For a unique product target and an oracle returning coordinate correctness on arbitrary queries, coordinate-wise search uses $O(kc)$ queries when all component arities are c. This is a stronger feedback model than a scalar fraction-correct reward on sampled outputs. The binary-only bound is a worst case over hidden target sets. It does not rule out exploiting the structure of an explicitly supplied DFA, nor does it establish an exponential separation for every structured task in the experiments.

A.5 Proof of Proposition 4 (Dead-Zone Trapping)

Proof. **Part (a): Zero-gradient stasis.** When $m_t = 0$ (no correct sample), all rewards are zero. No-baseline REINFORCE: $\widehat{\nabla} J = N^{-1} \sum_i r(y_i) \nabla \log \pi(y_i) = 0$. GRPO: $A_i = (r_i - \bar{r})/\sigma = 0/0$; implementations return zero advantage. Under the proposition's additional assumption that such batches cause no parameter update, $\theta_{t+1} = \theta_t$ and $P_{t+1} = P_t$. Zero reward gradient alone is insufficient when optimizer state, weight decay, or auxiliary objectives move the parameters.

If $P_0 = 0$, the same stasis assumption gives $T_{1/N} = \infty$ almost surely.

Part (b): Stasis probability. $\Pr[m_t = 0] = (1 - P_t)^N \ge 1 - NP_t$ (Bernoulli).

Part (c): First-signal time. Since $P_t = P_0$ for $t < \tau$ (by Part a), τ is geometric with rate $q_0 \le NP_0$. $\mathbb{E}[\tau] \ge 1/(NP_0) \ge c^k/(NM)$.

Part (d): Birth-process escape. After j productive steps, $P \le (1 + \varrho)^j P_0$ and the productive-batch probability is $q_j \le N(1 + \varrho)^j P_0$. The expected waiting time at stage j is $\ge 1/q_j$. Summing over $L = \lceil \log_{1+\varrho}(1/(NP_0)) \rceil$ stages:

$$\mathbb{E}[T_{1/N}] \ge \sum_{j=0}^{L-1} \frac{1}{q_j} \ge \frac{1}{NP_0} \sum_{j=0}^{L-1} (1 + \varrho)^{-j} \ge \frac{1 + \varrho}{\varrho} \cdot \frac{1 - NP_0}{NP_0} \ge \frac{c^k}{\varrho NM}.$$

The last step uses $1/(NP_0) \ge c^k/(NM)$ and the assumed condition $(1 + \varrho)(1 - NP_0) \ge 1$. The exact stage count is $L = \lceil \log(1/(NP_0))/\log(1 + \varrho) \rceil$; the approximation $\log(1 + \varrho) \approx \varrho$ requires small ϱ. A fitted typical growth rate, including the reported trajectory estimate 0.106, does not verify the uniform upper bound required here.

□

Quantities independent of ϱ. The first-arrival lower bound $\mathbb{E}[T_A] \ge c^k/(NM)$ does not depend on ϱ; Part (c) uses pre-arrival policy stasis. This information constraint under binary reward complements the exponential-time convergence results of Li et al. (2023), which concern optimization under dense reward.

A.6 Exact First-Update Gain

Exact first-update gain under plain SGD. Consider a unique correct output under a factored policy with k components, each initialized uniformly over c choices. Condition on a batch of N samples containing $h \ge 1$ correct outputs. For one no-baseline REINFORCE update with learning rate η, using the batch-mean gradient of the joint log probability without component-count normalization, and plain SGD without auxiliary terms or clipping, let $a = \eta h/N$. The correct logit in each

component increases by $a(1-1/c)$ and each incorrect logit decreases by a/c. Consequently,

$$p_1 = \frac{1}{1+(c-1)e^{-a}}, \qquad \frac{q_1}{q_0} = \left[\frac{c}{1+(c-1)e^{-a}}\right]^k, \qquad q_0 = c^{-k}.$$

Thus policies with the same initial joint correctness can respond differently to the same number of correct samples.

For fixed $Q = c^k > 1$ and $a > 0$, this first-update correctness increases with k along the matched factorizations. Let $L = \log Q$ and $f(x) = \log(1 - e^{-a} + e^{-a}e^x)$. The function f is convex with $f(0) = 0$, so $f(x)/x$ is nondecreasing for $x > 0$. Since $q_1 = \exp[-kf(L/k)]$, increasing k increases q_1. This local calculation gives an exact mechanism for a first-update factorization effect. The R24 implementation sums component log probabilities and takes a batch mean, with no $1/k$ normalization, but uses group-centered advantages and Adam. Equation 3 instead uses no-baseline REINFORCE and plain SGD. It is an exact analytical illustration under those conditions, not the exact first update of the R24 implementation or a prediction of its multi-step escape times.

For the R24 factorizations, conditioning on $h = 1$ with $N = 256$ and setting $\eta = 100$ gives the following values of q_1/q_0, calculated from Equation 3:

(k, c)	$(3, 256)$	$(4, 64)$	$(6, 16)$	$(8, 8)$	$(12, 4)$	$(24, 2)$
Joint loss	3.210	4.631	8.733	14.308	28.027	68.897
Joint loss divided by k	1.475	1.468	1.441	1.406	1.339	1.215

These are analytical illustrations, not additional training measurements or fitted predictions for the reported Adam trajectories. The first row uses the batch mean of the joint log probability. The second divides that loss by k at the same nominal learning rate, replacing a by a/k in Equation 3. Its reversed ordering makes the normalization dependence concrete. This changes the update scale and is not a measured ablation of the reported Adam experiments.

Local sensitivity at matched reward variance. Differentiating the exact log-gain at $a = 0$ gives

$$\frac{\partial}{\partial a} \log \frac{q_1}{q_0}\bigg|_{a=0} = k(1-1/c) = \|\nabla_\theta \log \pi_\theta(y^*)\|_2^2.$$

The equality follows because each component score has one coordinate $1-1/c$ and $c-1$ coordinates $-1/c$. For binary reward the initial variance is $q_0(1-q_0)$; every R24 factorization shares this variance while the displayed score norm differs. This illustrates how a fixed initial reward distribution leaves parameter-space update sensitivity undetermined. It is compatible with a reward-variance upper bound on the expected gradient (Razin et al., 2024), which does not require equal gradients for equal reward variance.

A.7 Proof of Reachability Reduction

For every reachable state $q \in R$, each transition $\delta(q, a)$ also lies in R. If $\mathcal{A}'$ agrees with $\mathcal{A}$ on transitions from R *and on accepting-state membership within* R, induction on word length gives identical reached states and acceptance decisions for every word. Transitions and acceptance flags outside R therefore sum out of the correctness probability. The transition-only count is $s|R|$; when acceptance flags are independently generated, the full constrained count is $(s+1)|R|$. For a uniform policy over G destination states and two acceptance choices, a specified reachable labeling has probability $G^{-s|R|}2^{-|R|}$. Any additional accepted labelings must be summed under their actual probabilities.

A.8 Generality of the Dead Zone: Full Statement

A fixed baseline on an all-failure batch. Let $q(\theta) = \Pr_\theta[\phi(Y) = 1]$, with $0 < q < 1$, and let b be a fixed scalar baseline. For a regular score-function estimator with gradient-ascent sign,

$$\mathbb{E}[\widehat{\nabla J} \mid \text{all failures}] = -b\,\mathbb{E}[\nabla \log \pi_\theta(Y) \mid \phi(Y) = 0] = \frac{b\nabla q(\theta)}{1-q(\theta)}.$$

The equality follows from $\mathbb{E}[\nabla \log \pi_\theta(Y)] = 0$ and $\mathbb{E}[\phi(Y)\nabla \log \pi_\theta(Y)] = \nabla q$. For $b > 0$, this conditional expected gradient is a positive multiple of ∇q. This statement concerns the mean gradient, not every realized or optimizer-transformed update. The realized gradient need not vanish, and this estimator is outside the pre-arrival stasis class. Learned baselines, auxiliary objectives, and optimizer-state effects require their own analysis.

Non-uniqueness and the effective exponent.

Proposition 6 (Non-uniqueness bound)**.** *Let $k \to \infty$, let $M = M(k) \geq 1$, and define $\eta_M = M^{-1/k} = \exp(-\log M/k)$. If $\log M = o(k)$, then $\eta_M \to 1$. If $\log M = \Theta(k)$, then η_M is eventually bounded away from both 0 and 1. In particular, if $\log M/k \to a$ for some $a \in (0, \infty)$, then $\eta_M \to e^{-a}$. If $\log M = \omega(k)$, then $\eta_M \to 0$; this regime is impossible when all component arities are uniformly bounded in k, since $M \leq c^k$ for an arity bound c.*

Proof. Each conclusion follows from $\eta_M = \exp(-\log M/k)$. For $\log M = \Theta(k)$, there exist constants $0 < a \leq b < \infty$ such that $a \leq \log M/k \leq b$ for all sufficiently large k, which gives $e^{-b} \leq \eta_M \leq e^{-a}$. □

We define the *effective exponent* to unify reachability reduction and multiplicity:

Definition 7 (Effective exponent)**.** *Given constrained component count k_{reach} and multiplicity M,*

$$\boxed{k_{\text{eff}} = k_{\text{reach}} \cdot \frac{\log N}{\log N + \log M}}. \tag{6}$$

This maps the optimistic boundary $(NM)^{-1/k_{\text{reach}}}$ to $N^{-1/k_{\text{eff}}}$, expressing non-uniqueness as a reduction in effective complexity. (k_{eff} depends on N: an analysis parameter, not an intrinsic complexity measure; Algorithm 1 reports observed outcomes separately.)

Observation 8 (Zero reward gradient and zero parameter update)**.** *If every sample's contribution has the form $f(\phi(Y), \pi_\theta)\nabla_\theta \log \pi_\theta(Y)$ with $f(0, \cdot) = 0$, an all-failure batch has zero reward gradient. No-baseline REINFORCE satisfies this condition. Group-centered GRPO gives zero advantages to an all-failure group when its zero-variance convention returns zero or skips the group (Shao et al., 2024). The claim applies to a PPO or advantage-weighted implementation only when its specified advantages also vanish on that batch. It does not apply to arbitrary fixed or learned baselines. A frozen policy additionally requires no parameter change from momentum, weight decay, auxiliary losses, or updates on other prompts.*

B Transition Sharpness

Transition-width statement. Under $P_{\text{correct}} \approx Cp^k$ the transition width is $\Delta p = O(p^*/k)$ (Proposition 9): at k=30, N=1000, per-component accuracy is about 79% when the expected number of correct samples per batch reaches one. It is the trivial AND case of sharp-threshold theory (Friedgut, 1999); we have not fitted it to measured widths.

Proposition 9 (Transition sharpness)**.** *Under $P_{\text{correct}} \approx Cp^k$ for effective multiplicity $C \in [1, M]$, the width of the transition from $\Pr[\geq 1 \text{ correct}] = \epsilon$ to $1 - \epsilon$ is*

$$\boxed{\Delta p \approx \frac{p^*}{k} \ln\left(\frac{-\ln \epsilon}{-\ln(1-\epsilon)}\right) = O\left(\frac{p^*}{k}\right)}. \tag{7}$$

The transition sharpens as k increases: the shift from "no signal" to "reliable signal" is abrupt (Figure 5, Appendix I; for k=30 at N=1000, per-component accuracy is about 79% at one expected correct sample per batch). This connects to sharp-threshold theory (Friedgut, 1999): the AND function's total influence $I_p = kp^{k-1}$ grows at the threshold as $k \to \infty$ (Bourgain et al., 1992; O'Donnell, 2014); the $O(1/k)$ width is the *trivial* special case for AND (direct Taylor expansion), strictly sharper than the general BKKKL/Talagrand $O(1/\log k)$ bound (Talagrand, 1994).

C CORRELATED COMPONENTS

Let C_i be component-correctness events under one scoring distribution, $p_i = \Pr(C_i)$, $\bar{p} = k^{-1}\sum_i p_i$, and $q = \Pr(\bigcap_i C_i)$. For $0 < q, \bar{p} < 1$, the descriptive parameter $\gamma = k \log \bar{p} / \log q$ rewrites $q = \bar{p}^{k/\gamma}$. Under independence, AM–GM gives

$$q = \prod_i p_i \leq \bar{p}^k \quad \Longrightarrow \quad \gamma \leq 1.$$

Equality holds when the component probabilities are equal. Thus, at the population level, $\gamma > 1 \Rightarrow q > \bar{p}^k \geq \prod_i p_i$: the all-correct event has excess probability relative to the product of the marginals, which marginal heterogeneity alone cannot explain under independence. This is a sufficient condition for that joint-event excess, not a characterization of pairwise positive association. Conversely, $\gamma \leq 1$ does not imply independence: heterogeneity can keep q below $\bar{p}^k$ even when $q > \prod_i p_i$.

For a fixed fitted γ, the algebraic expected-hit boundary is $N^{-\gamma/k}$; no architecture-wide constant is implied. An empirical $\widehat{\gamma} > 1$ is subject to sampling uncertainty. When both marginals and joint correctness are measured over parseable outputs, the interpretation concerns that conditional distribution. These estimates do not identify a causal effect of pretraining or validate extrapolation of a fixed exponent to larger tasks.

D THE LE CAM LOWER BOUND

Binary-feedback query lower bound. Any algorithm observing only binary feedback needs $Q \geq (1-2\delta)c^k/(2M)$ queries in the worst case (Remark 10), so the exponential cost is an information barrier and not a REINFORCE pathology—though as a worst case over adversarially placed instances it does not locate the boundary for the structured instances of §4; that is Theorem 3's job.

Could a smarter algorithm circumvent the dead zone? In the worst case, no:

Remark 10 (Le Cam lower bound)**.** *Let $|\mathcal{C}_i| \geq c$ for all i and $c^k \geq 2M$. Any algorithm—adaptive, randomized, with unbounded computation—that observes only binary feedback needs a total query budget $Q \geq (1-2\delta)\, c^k/(2M)$ to succeed with probability $\geq 1-\delta$ over all instances with $|\mathcal{Y}^*| = M$ (Le Cam's two-point method with adversarial placement of $\mathcal{Y}^*$; the component-factored instantiation of classical combination-lock bounds (Krishnamurthy et al., 2016; Kakade, 2003; Angluin, 1981); proof in Appendix A).*

This is a worst-case statement over adversarially placed instances; it certifies that the exponential cost is an information barrier rather than a REINFORCE pathology, but it does not by itself locate the boundary for the structurally placed instances of §4—that is what the per-instance, per-batch boundary $p^* = N^{-1/k}$ of Theorem 3 does (the binary-vs-per-component feedback gap is discussed in Appendix A).

E EXTENDED RELATED WORK

Gradient starvation and RLVR dynamics. Razin et al. (2024) analyze suppression of the expected policy gradient by small reward variation, including under adequate exploration. Nie et al. (2026) analyze zero advantages under group-mean centering and introduce fixed-reference advantages that produce updates on all-failure groups. Their assumption $0 < p < 1$ includes arbitrarily small positive success probabilities, so it does not define a regime disjoint from our rare-arrival setting. Huang et al. (2026) analyze transformer training on compositional state-tracking tasks. Their Proposition 4.1 gives an exponentially flat region despite nonzero terminal reward, and their mixed-difficulty analysis explains an implicit curriculum through shared updates. Their theoretical update uses length-normalized policy gradients (Equation 11), whereas our joint-loss first-update illustration states its normalization explicitly (Appendix A.6). These are analytical antecedents, not solely empirical observations of failed rollouts. Our distinct evidence is the matched-factorization separation of first arrival from subsequent adaptation, followed by controlled interventions and explicit scope tests. The elementary p^k and maximum-to-sum calculations support that evidence.

Estimator and sampling interventions. Related RLVR methods modify advantages, rollout allocation, or the prompt distribution (Yu et al., 2026; Zhu et al., 2026; Rad et al., 2026; Yu et al., 2025; Le et al., 2026; Liang et al., 2026; Baroian & Berger, 2026; Pang et al., 2026; Xu et al., 2026). The all-correct group is also zero-variance under group centering, which distinguishes the first positive reward from a necessarily productive GRPO update. Algorithm 1 is a reporting protocol for observed runs; it is not an estimator repair or a validated automatic curriculum. Hindsight relabeling and archived-state exploration change the goal pairing or behavior distribution (Andrychowicz et al., 2017; Ecoffet et al., 2021), and sit outside a fixed-policy first-arrival calculation.

Binary reward, partial credit, and diversity. Dymetman (2026) studies an objective that cannot distinguish distributions supported on valid outputs by correctness alone, raising a diversity issue after success. This concerns a different measured outcome from arrival, without requiring disjoint ranges of nonzero initial success probability. Li et al. (2026) examines limitations of pass-rate rewards in critic-free training. VeRPO aggregates per-test outcomes into a weighted scalar reward and combines local and global advantages (Wang et al., 2026a). It is therefore inaccurate to explain its difference from our results by claiming that it uses independent component-wise policy gradients while ours uses a scalar reward. Our dense position control and graded LLM control also use a scalar fraction-correct reward. Their divergent outcomes indicate dependence on the policy, optimizer, task, and reward construction; the experiments do not isolate which difference explains the cross-paper comparison. Base-policy pass@k limits observed RLVR gains in related studies (Yue et al., 2025); our finite probes supply noisy estimates, not instance-exact certificates of learning.

Classical antecedents and formal-language structure. Independent correctness decay is standard in reliability theory and Boolean function analysis (Barlow & Proschan, 1975; O'Donnell, 2014). Combination-lock and query-learning lower bounds establish the cost of uninformative binary feedback in their respective access models (Mitzenmacher & Upfal, 2005; Krishnamurthy et al., 2016; Kakade, 2003; Angluin, 1981; 1990). We claim neither that decay nor the zero-advantage observation as new mathematics. Formal languages provide an instrument for exact verification and structural matching, complementing architectural studies (Delétang et al., 2023; Liu et al., 2023a; Butoi et al., 2025). Reward machines structure feedback (Toro Icarte et al., 2022; Li et al., 2024); differentiable automata provide continuous relaxations (Mordvintsev, 2022; Umili & Capobianco, 2024).

Other relevant distinctions. Exploration, shaping, curricula, code feedback, and process supervision address sparse-reward learning through different information sources (Pathak et al., 2017; Ng et al., 1999; Gupta et al., 2022; Bengio et al., 2009; Liu et al., 2023b; Lightman et al., 2024). Binary verifiers can also be combined with flexible human feedback (Wang et al., 2026b). Feature-level gradient starvation (Pezeshki et al., 2021) concerns a different mechanism from an all-failure reward estimator. Metric-induced sharpness (Schaeffer et al., 2023) is relevant because final exact-match thresholds can mask continuous accuracy gains, as Figure 4 illustrates. Our fixed-policy arrival probability is smooth in q_0; the exact zero is a property of a realized all-failure estimator, not a discontinuity in that population probability. Computational phase transitions (Cheeseman et al., 1991; Monasson et al., 1999) concern yet another object, the difficulty of solving problem instances.

F Scope Limits on the Levers

The scope limits stated in short form in §6's Limitations paragraph, in full.

Isolated and shared updates. Isolated-prompt training makes arrival rate-invariant only when the full update leaves the policy unchanged before its first positive reward. Shared updates remove this premise and change arrival in both directions in the tested two-prompt setting. The eight-prompt comparisons have positive observed directions but limited instance-level power; they do not establish preservation of effect magnitude. At thirty prompts, higher rates and longer continuations recover final success on subsets of the tested prompts. These comparisons have three independent training seeds per arm, with coupled prompt outcomes, and do not isolate prompt count, group size, or composition. The observed changes therefore support a scoped intervention result. An autoregressive control reverses the rate effect ($3/3$ to $0/3$ final successes with unchanged arrival), so neither the rate multiple nor improvement under a larger rate is a general prescription.

Open theoretical questions. The p^k model assumes independence; correlated components, a dynamic $M_{\text{eff}}(\theta)$, and what the theory does and does not say about curricula (Bengio et al., 2009) remain future work.

One model scale. The central rate interventions exist on both LLM task families at one model scale (0.5B). Selected-instance studies are supplemented by the uniform draw of 30 buildable regex targets. We claim neither the lever ordering for RLVR in general nor that the constants measured here transfer to open-ended generation (Appendix I.11).

Escape-time magnitudes are a boundary phenomenon. On the (k, c) grid the k-effect replicates in direction at every iso-p_0 slice but not in size, and the fixed-H ladders sit on the boundary by construction (Appendix I.10).

What the LLM ladder cost. The matched ladder could not be transplanted to the LLM at all: the cells it needs are exactly the ones where a 0.5B model stops emitting scorable output (Appendix I.10), so the instances we do report were selected around that failure. The k-dependence of escape is therefore a claim about the factored substrate, whose ladders are narrower than Proposition 4, and not about this LLM. At high learning rates, both optimizers exhibit rapid jumps and the observed k-trend disappears. The gradual-compounding interpretation is then uninformative; large observed gains alone do not establish a violation of the uniform multiplicative-bound assumption in Proposition 4.

Memorization. A pre-registered probe (exp21, $R10$) finds no evidence that memorization drives the regex split—a contamination ratio of 1.81 against a registered threshold of 3—but the registered rule was three-way, and "no evidence" is not exclusion; it is not excluded per target (Appendix I.11).

G Arrival and Final-Outcome Reporting

This protocol separates observed arrival from final success. Final success without observed training arrival is a distinct outcome; it alone establishes neither zero-shot capability nor an evaluation fault. The protocol does not infer zero success probability from an absent hit or assign a specific optimization mechanism to every trapped run. Task eligibility and parsing outcomes are reported with the protocol. Pre-training probes estimate the initial correctness probability separately, with sampling uncertainty.

Probe uncertainty is reported separately. For h correct outputs in n independent initial-policy samples, $\widehat{q}_0 = h/n$ estimates the probability of *any* accepted output. Neither a multiplicity factor nor an effective exponent should be applied to this total correctness estimate again. For fixed n and zero hits, the exact one-sided confidence limit is $q_U = 1 - \alpha^{1/n}$. At $\alpha = 0.05$, $n = 2048$, and $B = 19200$, $Bq_U = 28.1$. With $n = 64$ the corresponding bound is approximately 878. For a zero-hit probe to give $Bq_U < t$ requires $n > \log \alpha / \log(1 - t/B)$, hence at least 57,517 samples for $t = 1$ and 19,172 for $t = 3$ at this budget. These are fixed-sample guarantees; repeatedly probing until a desired bound appears requires a separate sequential-validity argument.

With one hit in 2,048 samples, the point estimate is $\widehat{H} = 9.375$ but the one-sided 95% lower bound is only approximately 0.481. Thus neither zero hits nor a positive point estimate supports a deterministic pre-training label of Dead or Alive. The uniform draw has 11 positive-hit and 19 zero-hit probes; these observed groups are distinct from the retrospective run outcomes in Algorithm 1.

Finite final-evaluation resolution. The LLM final-success criterion is an empirical threshold of at least 58 correct outputs among 64 post-training samples. For illustration, under independent samples and a fixed verifier, a policy with true joint accuracy 0.85 passes with probability $\Pr\{\text{Binom}(64, 0.85) \geq 58\} \approx 0.1365$. A pass is therefore an operational outcome, not a confidence certificate that population accuracy is at least 0.9. Continuous final accuracy accompanies the thresholded outcomes (§5.5). In the uniform-draw learning-rate comparison, a post-hoc retabulation of the saved 64-sample counts gives the same $9/90$ versus $27/90$ result, with 18 pairs moving up and none down, for every cutoff from 56 through 59 correct. This local sensitivity check does not replace the pre-specified 58/64 criterion.

Algorithm 1 Arrival and final-outcome reporting

Require: A run under the stated evaluation protocol; positive-reward counts $K_1, \ldots, K_E$; final accuracy $\widehat{J}_E$; success threshold τ

$T_A \leftarrow \min\{t : K_t > 0\}$, or ∞ if no such episode exists
if $T_A = \infty$ **then**
  if $\widehat{J}_E \geq \tau$ **then**
    status $\leftarrow$ FINAL SUCCESS WITHOUT OBSERVED ARRIVAL
  else
    status $\leftarrow$ NEVER ARRIVED WITHIN BUDGET
  end if
else if $\widehat{J}_E \geq \tau$ **then**
  status $\leftarrow$ ESCAPED AT FINAL EVALUATION
else
  status $\leftarrow$ ARRIVED BUT TRAPPED AT FINAL EVALUATION
end if
return $(T_A, \widehat{J}_E, \text{status})$

Observed arrival is distinct from useful updates. A first positive reward establishes arrival. A group containing only correct outputs also has zero group-centered advantages, so arrival does not imply that the same episode necessarily produces a reward-directed update. If no scorable output is generated, report that fact and the task's eligibility convention alongside arrival and final accuracy, rather than inferring an inability from a finite zero-hit probe. Training-group threshold crossings should be logged separately from the post-training evaluation criterion.

Partial-credit comparison. We no longer recommend replacing binary reward with a graded one on a dead instance. Paired over the 54 $(k, \text{target}, \text{seed})$ cells where both rewards were run on the same instance and seed, the sequence-level graded reward of §H rescues 1 of the 21 cells where binary failed and destroys 8 of the 33 where binary succeeded (McNemar $p = 0.039$; five go from joint accuracy 1.000 to 0.000), and the harm concentrates in the transition zone (6 destroyed, 0 rescued, $p = 0.031$). Both tests count seeds of one instance as independent units, and they should not be read as instance-level evidence: the 8 destroyed cells come from three instances and the 6 from two, so with the instance as the unit the exact tests give $p = 0.63$ and $p = 0.50$. The paired counts show harm in these runs, with limited instance-level evidence for its magnitude. The position-constrained trajectory in Appendix I.11 illustrates one failure mechanism. Successful factored and autoregressive controls appear in Table 7 and Figure 15. These experiments change policy, initialization, and optimization together; they do not isolate which difference causes the contrasting outcomes. The conclusion concerns the tested scalar rewards and configurations, and does not establish the behavior of token- or process-level credit assignment.

H REWARD AND CONTROL SIGNAL DEFINITIONS

Parameterization, control, and metric. We sweep signal strength by $\theta = \alpha T^* + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I)$ and T^* the target transition tensor, which walks the convergence path without training-dynamics confounds, and use REINFORCE (Williams, 1992) as a transparent controlled estimator (Observation 8 states the precise extension conditions). Two non-binary *controls*, neither a contribution of this paper, isolate the reward formulation as the experimental variable (Appendix H). We report gradient signal-to-noise S/N $= |\mu_i|/\sigma_i$ averaged over parameters, with zero reported in the all-failure cases examined here; S/N alone is not a general certificate of no positive rewards.

Soft state-overlap control (signal-to-noise sweep, §4.2). Let $T \in \mathbb{R}^{|\Sigma| \times n \times n}$ be the (softmax-relaxed) transition tensor of a DFA, s_0 its initial state distribution, and $\mathbf{a} \in \{0,1\}^n$ its accepting-state indicator. On a string $x = x_1 \cdots x_{|x|}$ the soft run is $s_{t+1} = \sum_{i,j} [x_{t+1}]_i \, [s_t]_j \, T_{ij\cdot}$ and the per-step acceptance probability is $a_t(x) = \langle s_t, \mathbf{a} \rangle$, $t = 0, \ldots, |x|$. For a string set S, $\text{SSO} = \frac{1}{|S|} \sum_{x \in S} \left(1 - \frac{1}{|x|+1} \sum_{t=0}^{|x|} |a_t^{\text{cand}}(x) - a_t^{\text{tgt}}(x)|\right) \in [0,1]$; loss $1 - \text{SSO}$. In the reported per-point S/N sweep, each Monte-Carlo draw uses 50 strings. Lengths are sampled uniformly from the integers 1

through 20 and each symbol is sampled independently and uniformly from the binary alphabet. The fixed seed schedule is $42 + 1000j$ for draw j; the same draw-indexed sets are reused across targets, so the reported S/N varies the string set but not its sampling law. This control measures overlap under the string-sampling distribution; it does not certify language equivalence.

Product-construction random-walk control (training curves, Figure 14). With $T_\sigma^\times = T_\sigma^{\text{cand}} \otimes T_\sigma^{\text{tgt}}$, $s_0^\times = s_0^{\text{cand}} \otimes s_0^{\text{tgt}}$, and $\Delta = \mathbf{a}^{\text{cand}} \otimes (1 - \mathbf{a}^{\text{tgt}}) + (1 - \mathbf{a}^{\text{cand}}) \otimes \mathbf{a}^{\text{tgt}}$ the indicator of symmetric-difference product states, let $s_{t+1}^\times \propto \frac{1}{|\Sigma|} \sum_\sigma s_t^\times T_\sigma^\times$ (renormalized) be a uniform random walk on the product; then $d_{\text{prod}} = \frac{1}{K} \sum_{t=1}^{K} \langle s_t^\times, \Delta \rangle$ with $K = 30$, and the control reward is $1 - d_{\text{prod}}$. This finite-horizon distributional language distance is used only as a control; no new property of the distance is claimed. The test-string disagreement rate $\hat{d}$ reported in §4.2 is a separate, non-differentiable evaluation metric.

All reward signals used.

Name (as used in text)	Where	Definition
Binary	§4.2, §4.2, §5	1 iff all k components match (factored/transformer); exact minimal-DFA identity (regex DFAs); exact match on all reachable components (product); exact constrained-string match (GRPO, position-constrained); minimal-DFA equivalence of the parsed regex (GRPO, regex)
Fraction-correct (dense)	factored and transformer policies; regex DFAs; product	fraction of correct components (k_{trans}-weighted for regex DFAs; reachable components only for product)
Sequence-level graded (GRPO, position-constrained)	§5	fraction of constrained positions correct, one scalar per completion
DFA-agreement graded (GRPO, regex)	§5	agreement rate on one fixed per-instance 200-string set (100 sampled from the target language and 100 from its complement; seed 0)
Soft state-overlap (control)	S/N sweep	SSO above
d_{prod} (control)	training curves	$1 - d_{\text{prod}}$ above

For each half of the DFA-agreement set, a feasible length is sampled uniformly from $0, \ldots, 12$, then a word of that length is sampled from the corresponding DFA; the empty string can occur exactly when length zero is feasible. The set is constructed once before training and reused for every reward call in that run. Agreement on this fixed finite set does not certify language equivalence and permits optimization against the proxy; this is a design limitation, not an established cause of the observed failures. The position-constrained near-miss trajectory uses position scores and does not depend on this set.

I Extended Experimental Data

Success criteria across experiments. Each experiment fixes its own success criterion, stated where used: gradient S/N (§4.2), per-component accuracy > 0.9 (autoregressive comparison, Table 7), greedy exact match on all (reachable) components (regex corpus §4.2, product training), and joint accuracy ≥ 0.9 over 64 post-training samples (LLM GRPO). Criteria are fixed within each experiment before its runs; no criterion is compared across experiments.

Is the success bar load-bearing? The factored-ladder robustness analysis uses $\bar{p} > 0.99$. Moving it to 0.9 changes the side of 1 of the $1{,}500$ base-arm ladder runs, and moving it to 0.5 changes 2; only 2 runs finish anywhere in $(0.5, 0.99]$. Final $\bar{p}$ is bimodal—a run either converges or stays near its initialization—so there is almost nothing in the middle for a bar to carve, and every k-trend in Appendix I.10 is numerically unchanged at all three bars. Recomputed from the run artifacts by `experiments/robustness_for_appendix.py`.

Problem	k	M	k_{reach}	Transition type
DFA equivalence	sn	1	sn	k-determined
DFA minimization	sm	$(m-1)!$	sm	Symmetry-gov. (M_{eff})
Product constr.	sn_1n_2	1	$s\lvert R_\times\rvert$	Distributional

Table 2: FA problem taxonomy (idealized: all states reachable; deviations in Appendix I). $s = |\Sigma|$, n states, m minimal size, $|R_\times|$ reachable product states.

I.1 Analytic Phase Transition Curves

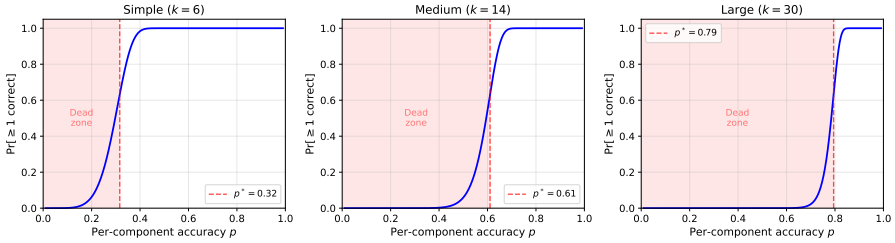


Figure 5: Phase transition of binary reward for three complexity levels ($N = 1000$). As k increases, the dead zone expands and the transition sharpens. For $k = 30$, per-component accuracy is about 79% at one expected correct sample per batch.

I.2 DFA Minimization: Full Table and Figures

DFA minimization and unrealized multiplicity. Corollary 5 predicts a non-monotone $p^*_{\text{dead}}(m)$ from the $(m-1)!$ relabelings; the measured boundary is monotone and tracks p^*_{alive} (corr. 0.994; 25 minimal DFAs per $m \in \{3, 4, 5, 6, 7, 8, 9, 11, 14\}$), because a policy converging to one canonical answer has $M_{\text{eff}} \approx 1$. The transition is governed by policy symmetry, not instance size.

Empirical minimization boundary. Corollary 5 predicts a *non-monotone* $p^*_{\text{dead}}(m)$; but Theorem 3 only brackets the true boundary between p^*_{dead} ($M_{\text{eff}} = M$) and p^*_{alive} ($M_{\text{eff}} = 1$), and the widening gap between them (Table 3, Appendix I) makes minimization a direct measurement of *which end a real policy occupies*. We measured the boundary: for each $m \in \{3, 4, 5, 6, 7, 8, 9, 11, 14\}$, 25 distinct minimal DFAs, sweeping signal strength α and recording the per-component accuracy at the first correct sample among N=1000 (Figure 6). The result is unambiguous: the empirical boundary is *monotone*, tracking p^*_{alive} (corr. 0.994) rather than the non-monotone p^*_{dead} (RMSE 0.283)—the factorial multiplicity is *not realized*, because a policy converging toward one canonical answer concentrates mass on a single labeling ($M_{\text{eff}} \approx 1$): for every $m \geq 4$, at least 22 of 25 first hits are that labeling, while at m=3, where relabelings are cheap, non-identity hits dominate (14/25) and pull the boundary toward the $1/m$ floor. The transition is thus governed by *policy symmetry*, not instance size: only a permutation-symmetric policy can reach the non-monotone p^*_{dead} (Appendix I).

Table 3 shows the complete phase boundary data for all m values tested.

For random 20-state DFAs over $\Sigma = \{0, 1\}$, the mean compression ratio is $m/n \approx 0.77$ (median $m = 16$), with most minimal DFAs having 14–19 states.

Small-m boundary measurement. The targeted small-m experiment behind Figure 6 uses 25 distinct minimal DFAs per $m \in \{3, 4, 5, 6, 7, 8, 9, 11, 14\}$ (generated by random-generate-then-minimize, deduplicated up to relabeling; correctness checked against the reported $(m-1)!$ start-preserving relabelings, with acceptance information carried by the relabeling, validated by brute-force enumeration at $m \in \{3, 4\}$). Per-m medians of the per-component accuracy at first hit: $0.37, 0.42, 0.55, 0.61, 0.65, 0.68, 0.70, 0.76, 0.80$—monotone, matching $p^*_{\text{alive}} = N^{-1/(2m)}$ (corr. 0.994) and not the non-monotone p^*_{dead}. Non-identity first hits by m: 14/25 (m=3), 3/25 (m=4), 2/25 (m=6), 0/25 at every other $m \geq$

m	k	$(m-1)!$	p^*_{dead}	p^*_{alive}	Inst.
3	6	2	0.282	0.316	1
6	12	120	0.377	0.562	1
9	18	4.0×10^4	0.378	0.681	3
10	20	3.6×10^5	0.373	0.708	5
12	24	4.0×10^7	0.362	0.750	8
14	28	6.2×10^9	0.349	0.781	20
16	32	1.3×10^{12}	0.337	0.806	39
18	36	3.6×10^{14}	0.325	0.825	23
20	40	1.2×10^{17}	0.315	0.841	8

Table 3: DFA minimization: complete theoretical phase boundaries. Instance counts reflect the natural distribution of minimal DFA sizes among 200 random 20-state DFAs ($s = 2$).

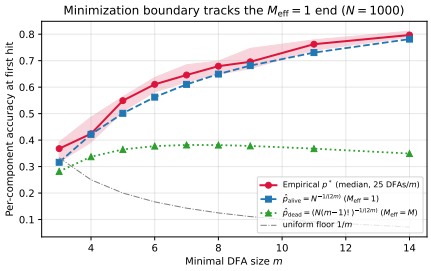


Figure 6: Minimization boundary, measured (median ± IQR, 25 minimal DFAs per m) vs. the two ends of the Theorem 3 interval: it tracks p^*_{alive} (corr. 0.994), not the non-monotone p^*_{dead}—concentrated policies forfeit the $(m-1)!$ multiplicity; at m=3, the only regime where relabelings are hit, the boundary is pulled toward the uniform floor.

5; at m=3, 12/25 DFAs already succeed at $\alpha = 0$ (the uniform floor). Raw data: `experiments/exp6_fa_expansion/results/phase2_small_m.json`.

I.3 Product Construction: Reachability and Jensen Effect

Product construction and per-instance boundaries. Only the k_{reach} transitions from reachable product states matter, and k_{reach} varies across instances ($\sigma = 15.9$ at 7×7); binary REINFORCE goes 50/50 on budget-alive instances and 2/55 on budget-dead ones, and success tracks k_{reach} and H, never nominal k.

Reachability and training outcomes. Product construction—produce the product DFA for $L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$—has nominal complexity $k = sn_1n_2$, but only $k_{\text{reach}} = s|R_\times|$ transitions from reachable product states matter, and k_{reach} *varies across instances*: for 7×7 products the reachability ratio falls to 0.43 with $\sigma(k_{\text{reach}}) = 15.9$ (30 random instances; Appendix I). Convexity of p^k then lets low-k_i instances dominate the mean reward (a ∼10× Jensen gap at $p = 0.85$), so aggregate success rates look easier than the average instance is. Training validates the instance-exact boundary: on 24 product instances spanning $k_{\text{reach}} \in [2, 82]$ (N=256, E=3,000, 5 independent seeds each; thresholds pre-fixed as in §4.2), binary REINFORCE goes 50/50 on the 10 budget-alive instances, 2/55 on the 11 budget-dead (both hits marginal-band, $H \in [0.03, 0.2]$, where the geometric model expects ≈1; deep-dead 0/40), and 10/15 on the three transition instances (model: 9.3); success tracks k_{reach} and H, never nominal k, and a dense control solves six deep-dead instances 18/18 (App. I).

Training validation (exp15). We train factored policies to emit the product DFA table for 24 random DFA pairs (s=2, state counts n_1=$n_2 \in \{2, 3, 4, 5, 7\}$), REINFORCE with mean baseline, N=256, E=3,000, 5 seeds per instance (independent RNG stream per instance–seed pair; an initial run that shared seed streams across same-shape instances is archived alongside). Binary success requires agreement with the target on *all reachable* components ($s|R_\times|$ transitions + $|R_\times|$ acceptance

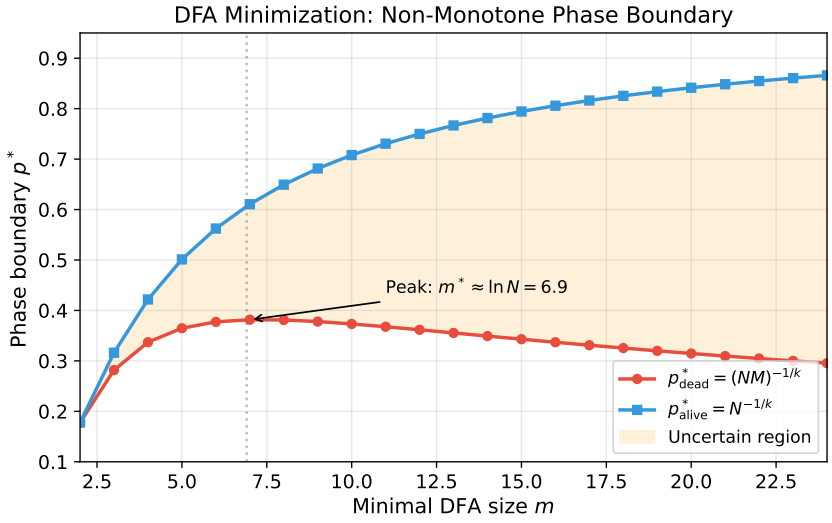


Figure 7: DFA minimization, *theoretical* boundaries: the optimistic end p^*_{dead} ($M_{\text{eff}} = M$) is non-monotone in m, peaking near $m^* \approx \ln N$, and the uncertain region (shaded) widens monotonically. As §4.2 shows, concentrated policies empirically realize the p^*_{alive} ($M_{\text{eff}} = 1$) end, not this curve.

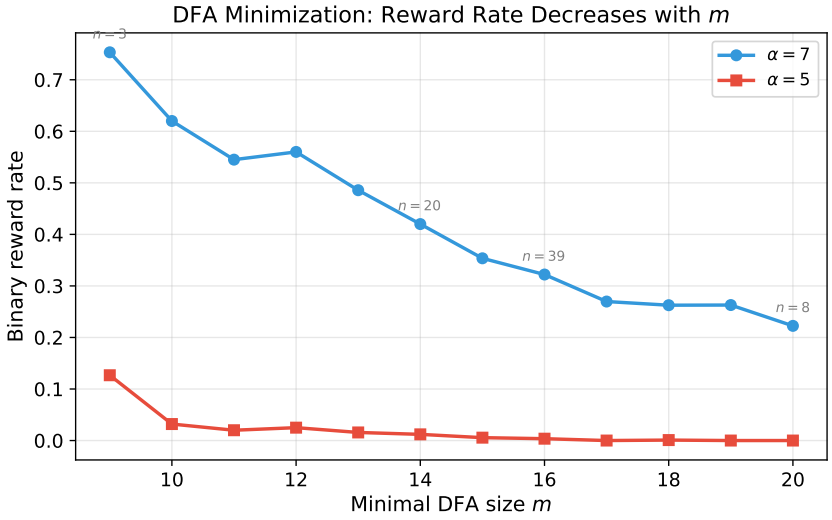


Figure 8: DFA minimization: empirical binary reward rate vs. minimal DFA size m at signal strengths $\alpha = 5$ and $\alpha = 7$ (200 random 20-state DFAs, $s = 2$).

bits); unreachable components are free, so $M{=}1$ over the constrained part and $p_0 = G^{-s|R_\times|}2^{-|R_\times|}$ with $G = n_1 n_2$ choices per transition, giving $H = ENp_0$ exactly as in §4.2. Results at the pre-fixed thresholds: all 10 budget-alive instances ($H \in [23, 2.4{\times}10^4]$) succeed 5/5; the 11 budget-dead ($H \leq 0.18$, down to 10^{-146}) yield 2/55—both late first hits (episodes 308 and 782) on the two *marginal*-dead instances ($H = 0.18, 0.03$), where the geometric model expects ≈1 success across the band ($\Pr[\geq 2] \approx 0.26$); the deep-dead instances ($H < 10^{-3}$) are 0/40; the three transition instances ($H \in [0.49, 2.9]$) hit 10/15 versus a model expectation of 9.3. Within every fixed nominal k, success tracks k_{reach}; within fixed k_{reach}, it tracks H through the alphabet factor G: at $k_{\text{reach}}{=}4$, success falls from 5/5 ($G{=}4$, $H{=}750$) through 10/10 ($G{=}9$), 5/5 ($G{=}16$), 3/5 ($G{=}25$, $H{=}0.49$) to 1/5 ($G{=}49$, $H{=}0.03$)—a dose–response in H, not a property of k_{reach} alone. A dense per-component control solves six deep-dead instances ($H \leq 1.6{\times}10^{-12}$, down to $9{\times}10^{-146}$; k_{reach} up to 82) 18/18, with first correct samples by episode 526. Grid note: at $n_1{=}n_2 \in \{5, 7\}$ only $|R_\times| \leq 2$

$n_1 \times n_2$	Full k	Mean k_{reach}	ρ	$\sigma(k_{\text{reach}})$	$k_{\min}$	$k_{\max}$
2×2	8	5.6	0.70	2.0	2	8
3×3	18	11.1	0.62	4.0	4	18
5×5	50	24.6	0.49	9.1	4	44
7×7	98	42.3	0.43	15.9	12	92

Table 4: Product construction: reachability reduction for random DFA pairs ($s = 2$, 30 instances each).

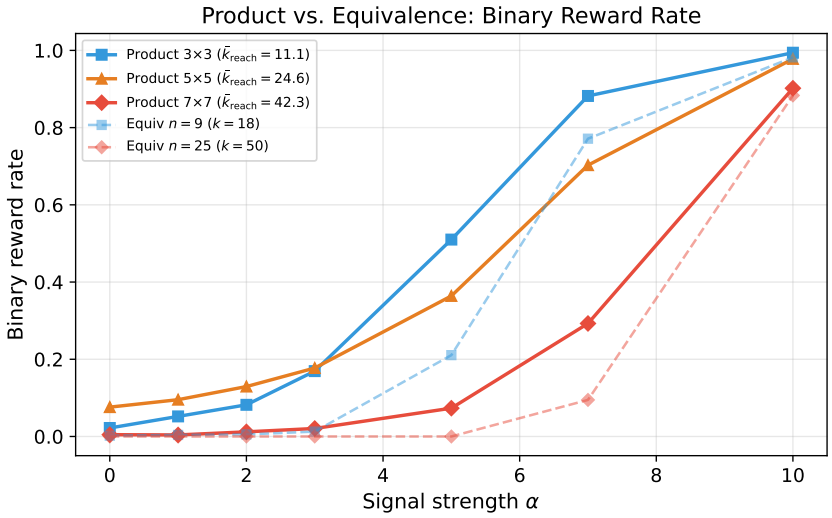


Figure 9: Binary reward rate across signal strengths for product construction (solid) vs. DFA equivalence (dashed) at comparable k values.

can be non-dead at this budget (each extra reachable state multiplies $1/p_0$ by $G^2 \cdot 2$), so smaller sizes populate the alive and transition bands; instances, per-run results, and the design document frozen before training will accompany the code release (`exp15_product_training`).

I.4 Unified Phase Diagram

I.5 Neural Experiments: Scaling Law and LLM Probe

Factored-policy k-sweep and architecture comparison. **Factored policy.** We train k independent categorical distributions over an alphabet of size $c = k/2$ to generate a fixed target sequence, sweeping $k \in \{6, 8, 10, 14\}$ with batch size N=256 (5 seeds per configuration, 50 at $k \in \{8, 10\}$; 5,000 episodes; binary and per-component dense reward). Binary reward succeeds reliably (50/50) for k=8, enters the transition zone at k=10 (9/50 seeds succeed, 18%), and fails completely for $k \geq 14$ (0/5); dense reward succeeds for all k. The expected correct samples per batch drops from 0.35 (k=6) to $4{\times}10^{-3}$ (k=8) to $3{\times}10^{-5}$ (k=10), explaining the sharp cutoff.

Components synchronize after the dead zone, and first-signal timing matches $\tau_{\text{first}} \approx 1/(Np_0^k)$ at a pooled observed-to-predicted ratio of 0.99 ($[0.83, 1.17]$) across batch sizes spanning 16× (Appendix I).

Architecture, and what actually moves the boundary. A small Transformer decoder (∼400K parameters) under identical REINFORCE shows no architecture dependence detectable at the ±25pp resolution of a success-rate comparison (k=10: 18% against 14%, Fisher $p = 0.79$, 50 seeds each), alongside the small measured factor $\gamma_{\text{arch}} = 1.12$ (below). The pretrained-model probe gives a different descriptive value: on one joint-vs-per-position estimator the random-init transformer gives $\gamma_{\text{arch}} = 1.12$ $[1.06, 1.14]$ at k=8 against $\gamma \approx 2.3$ for a probe of Qwen3-8B. The arms use different

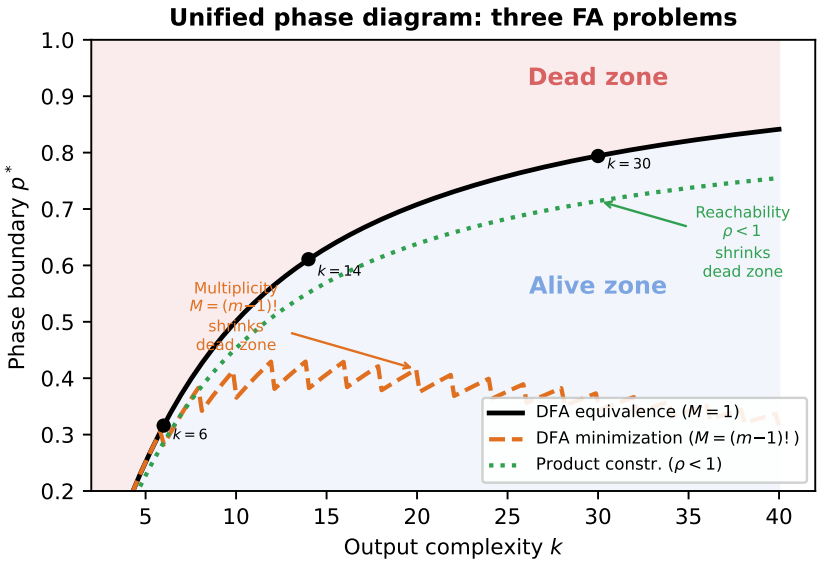


Figure 10: Unified phase diagram of *theoretical optimistic* boundaries (N=1000). Black solid: DFA equivalence (base boundary). Orange dashed: DFA minimization with full multiplicity ($M_{\text{eff}} = M$; §4.2 shows concentrated policies realize the base boundary instead). Green dotted: product construction (reachability lowers boundary). Dots mark tested k values.

tasks and our two estimators of γ disagree by $3\times$, so the observed ordering and values above one are descriptive; they do not identify a causal pretraining effect (Appendices I, I.9, I.11).

Synchronization detail (Figure 11). Under binary reward at k=8, the dead zone lasts 200–850 episodes of zero gradient, after which all k components converge within 0–1 episodes of each other; under dense reward, convergence spread grows with k.

LLM probe methodology (exp11). Both models are queried through the same local serving stack at temperature 1.0 (up to 1,200 new tokens). The prompt asks for a string of exactly L lowercase letters (a–z) with K position constraints ("position i must be 'c' "), L chosen per K; a response parses iff an exact-length lowercase string can be extracted. Per trial ($K \in \{3, 6, 9, 12, 15\}$, 3 trials each, 200 samples): parse rate, per-position accuracy over parsed samples, and joint accuracy P_{joint}; γ solves $P_{\text{joint}} = \bar{p}^{K/\gamma}$, with the bootstrap CI over trials. Raw per-trial records will accompany the code release (`exp11_llm_phase_transition`). The exploratory Exaone3.5-7.8B run gives median $\gamma = 2.4$ but from only 3 estimable trials, with bootstrap CI $[0.9, 3.9]$ including 1—direction, not magnitude.

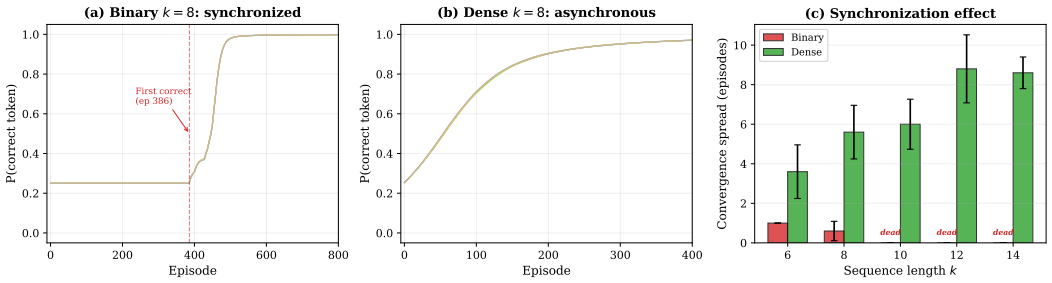


Figure 11: Convergence dynamics at the phase boundary (k=8). **Left:** Binary reward forces synchronized convergence after a dead zone. **Center:** Dense reward allows asynchronous convergence. **Right:** Binary convergence spread stays near zero; dense spread grows with k.

Matched-estimand dependence factor for the random-init transformer (pre-registered). For each checkpoint of the exp12 transformer with mean per-position accuracy $\bar{p} \in [0.3, 0.9]$, we sampled $\geq$ 2,048 outputs and computed $\gamma = k \log \bar{p} / \log P_{\text{joint}}$, the estimator used for the LLM probe. At k=8 (8 in-window checkpoints, 3 seeds) the median is $\gamma_{\text{arch}} = 1.12$, bootstrap 95% CI $[1.06, 1.14]$; the factored policy gives 1.01 $[0.97, 1.09]$ as the γ=1 sanity reference. At k=10 no checkpoint of any seed or architecture landed in the window (the escape is a rare, near-critical event), so γ_{arch} at k=10 is unmeasured. The original exp12 runs saved no weights, so the transformer cells were re-executed (seeds 42–46) with checkpointing; replaying a run under the same seed can reproduce a different escape outcome (GPU nondeterminism in the sampling path), so the checkpointed re-executions analysed here are not bit-identical to the exp12 runs behind the success-rate comparison, which we disclose as a reproducibility caveat.

Dependence factor and the boundary. Our LLM probe yields the window-sensitive estimate $\gamma \approx 2.3$ (median; see Appendix I.11 for the training-side values 1.46/4.4) for $K \geq 6$ (95% bootstrap CI: $[1.5, 3.7]$; 11 of 12 Qwen3-8B trials estimable), which algebraically gives a fitted expected-hit boundary at $\bar{p} \approx 0.44$ rather than 0.70 for $K = 15$ (the largest constraint count we probe) at $N = 200$.

Neural-policy experimental details. First-arrival timing is compared with the first-signal prediction $\tau_{\text{first}} \approx 1/(Np_0^k)$: across batch sizes spanning 16× with 30 independent seeds each, the pooled ratio of observed to predicted first-arrival time is 0.99 (95% bootstrap CI $[0.83, 1.17]$; per-batch-size means drift from 0.88 to 1.25 across the 16× range, a residual N-dependence the pooled constant averages over; Figures 11 and 12, Appendix I)—delayed-then-sudden like grokking (Power et al., 2022; Huang et al., 2026), but via reward starvation, not representation learning.[1] A secondary finding is *optimizer fragility*: the transformer needs 10× lower learning rate, distributing each rare gradient across 400K parameters rather than 32 logits. Two caveats: γ is computed over *parseable* outputs (parse rates 0.01–0.90), plausibly biasing it upward; and the Exaone3.5-7.8B run corroborates direction, not magnitude.

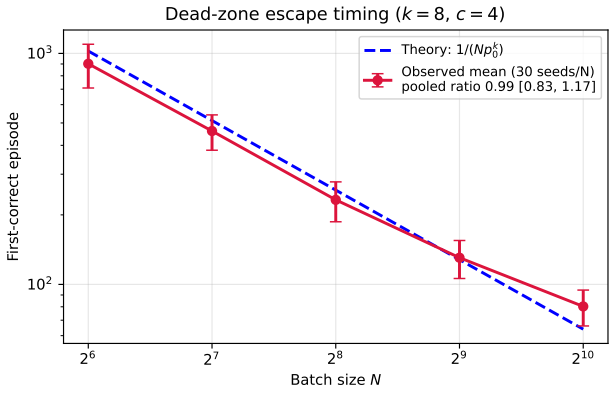


Figure 12: Dead-zone escape timing scales as $1/(Np_0^k)$. Observed mean first-correct episode (red) tracks the theoretical prediction (blue dashed) across batch sizes $N \in \{64, 128, 256, 512, 1024\}$ with pooled observed/predicted ratio 0.99 (95% bootstrap CI $[0.83, 1.17]$; 30 independent seeds per batch size; per-seed variance is high, as expected for geometric first-passage times).

I.6 DFA Equivalence: Training Curves and Full Gradient S/N Table

DFA equivalence at three sizes. At $k = 30$ the binary estimator's signal-to-noise ratio is exactly 0.000 for $\alpha \leq 3$—not one positive reward in 1,000 samples—while a soft state-overlap control

[1] An earlier version of this experiment reused the same random seeds across batch sizes, which produced a spuriously precise constant (1.31 ± 0.01) via common random numbers; the numbers reported here use an independent seed per run.

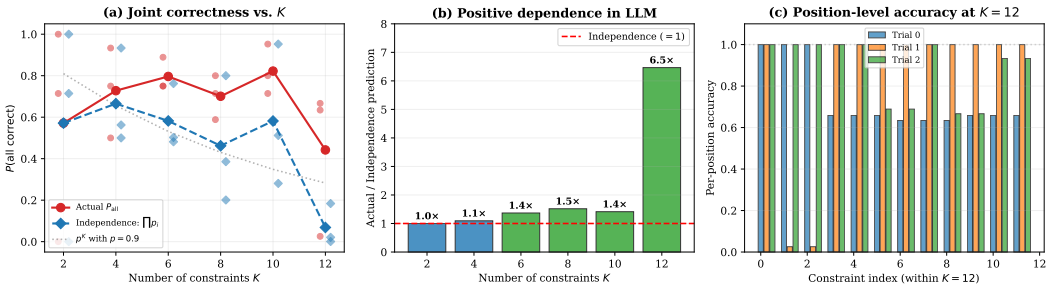


Figure 13: LLM inference probe (Qwen3-8B, n=50). **(a)** Joint correctness exceeds the independence prediction across all K. **(b)** The ratio grows with K. **(c)** Per-position accuracy at K=12: a few "weakest links" drag down joint correctness. An expanded experiment (n=200, K up to 15, two models) yields $\gamma = 2.3$ [1.5, 3.7] (see Appendix I.5).

keeps S/N ≥ 3.7 in every condition; REINFORCE finds equivalence in $27/30$ runs at $k = 6$ and produces zero gradient in $30/30$ at $k = 30$ (Figure 14).

DFA-equivalence results. We test three complexity levels—simple ($n = 3$, $k = 6$), medium ($n = 7$, $k = 14$), large ($n = 15$, $k = 30$)—with 5–6 signal strengths each (Appendix I), observing three regimes: at $k = 6$, binary S/N spans 0.08–0.81 (noisy but feasible); at $k = 14$, S/N stays below 0.1 for $\alpha \leq 3$ (predicted boundary $p^* = 0.61$); at $k = 30$, binary S/N is **exactly 0.000** for $\alpha \leq 3$—not one positive reward in 1,000 samples. The soft state-overlap control maintains S/N ≥ 3.7 across all 17 reported conditions (an 18th, simple/α=5, has a degenerate near-zero-variance binary estimate and is reported in Appendix I): the failure is a property of the reward formulation, not the task. Actual REINFORCE training (10 random targets $\times$ 3 seeds $\times$ 500 episodes; Figure 14, Appendix I): at $k = 6$, binary reward finds equivalence in 27/30 runs; at $k = 30$ it produces **zero gradient in all 30 runs**—the test-string disagreement rate $\hat{d}$ (fraction of held-out strings on which candidate and target DFAs disagree) stays at $\sim$0.50 (random) while the d_{prod} control reaches 0.20.

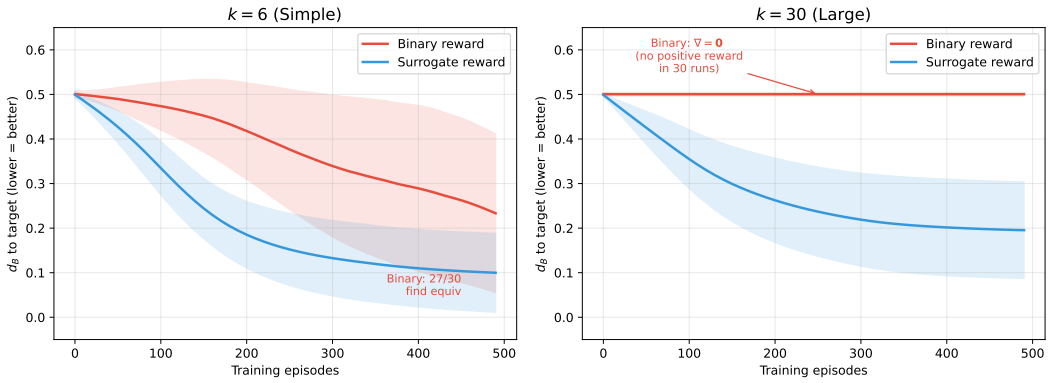


Figure 14: Training curves: test-string disagreement rate $\hat{d}$ to target over 500 REINFORCE episodes (mean $\pm$ std, 30 runs each). **Left** ($k = 6$): both rewards learn; binary finds equivalence in 90% of runs. **Right** ($k = 30$): binary is completely flat ($\nabla = \mathbf{0}$); the d_{prod} control makes steady progress.

Manufactured transition band. To test the boundary *where it is predicted*, we built ten instances with $H \in [0.52, 15.4]$ (varying the per-component choice space at fixed k_{reach}; Appendix I), prediction $1 - e^{-H}$ and thresholds frozen beforehand: success over 10 seeds rises across the predicted band (Table 5) at $\rho = 0.81$, but sharper than and shifted from the first-signal prediction—fitted scale 0.79, a rightward shift—which is what escape needing more than one hit looks like (Proposition 4). The informative cells are $H \leq 1.65$; above $H = 2$ the Wilson intervals admit any prediction over 0.72, so the calibration count is the weak half of this result and the rise is the strong half.

Table 5: **The boundary tested where it is predicted** (exp22; thresholds and the $1-e^{-H}$ comparison frozen before the runs). Binary REINFORCE, 10 seeds per instance; full table with n_{total} and Wilson intervals in Appendix I.

H	0.52	0.97	1.65	2.00	3.05	3.83	4.88	6.32	10.24	15.43
success / 10	2	2	9	10	10	10	10	10	10	10
$1 - e^{-H}$	0.41	0.62	0.81	0.86	0.95	0.98	0.99	1.00	1.00	1.00

Manufactured transition band (exp22, pre-registered). Ten DFA-equivalence instances ($|\Sigma| = 2$) were constructed so that the budget-aware expected hit count $H = ENp_0^{k_{\text{reach}}}$ ($E = 5{,}000$, $N = 256$; p_0 the uniform-initialization per-component accuracy) lands near the targets $\{0.5, 1, 1.5, 2, 3, 4, 5, 7, 10, 15\}$; because fully reachable two-symbol DFAs give only coarse H values, all ten instances use unreachable-state padding ($n_{\text{total}} > n_{\text{reach}}$), so H varies through the per-component choice space at fixed $k_{\text{reach}} = 6$, exactly as the reachability reduction (Appendix A) prescribes. Binary REINFORCE as in §4.2, 10 seeds per instance.

H	n_{total}	k_{reach}	success	Wilson 95% CI	$1 - e^{-H}$
0.52	28	6	2/10	[0.06, 0.51]	0.41
0.97	24	6	2/10	[0.06, 0.51]	0.62
1.65	21	6	9/10	[0.60, 0.98]	0.81
2.00	20	6	10/10	[0.72, 1.00]	0.86
3.05	18	6	10/10	[0.72, 1.00]	0.95
3.83	17	6	10/10	[0.72, 1.00]	0.98
4.88	16	6	10/10	[0.72, 1.00]	0.99
6.32	15	6	10/10	[0.72, 1.00]	1.00
10.24	5	9	10/10	[0.72, 1.00]	1.00
15.43	12	6	10/10	[0.72, 1.00]	1.00

Pre-registered verdict: PASS ($\rho = 0.81 \geq 0.8$; 9/10 calibrated). Two qualifications apply: the calibration count is weak evidence because the 10/10 cells' Wilson intervals admit any prediction above 0.72; the informative cells are $H \leq 1.65$, where the observed rise ($0.2 \to 0.9$ between $H \approx 1$ and 1.65) is sharper than $1 - e^{-H}$, consistent with success requiring more than one productive hit. The band is tested along the p_0 axis at (almost) fixed k_{reach}, not along k.

Random-DFA deviations from the idealized taxonomy. Random DFAs may have unreachable states, giving effective $k_{\text{reach}} < sn$ and $M > 1$. This makes the dead-zone observation *conservative*: it persists despite more favorable effective parameters than the $M{=}1$, $k{=}sn$ analysis predicts.

Regex-benchmark validation results. We trained factored policies on 100 DFAs derived from three regex benchmarks (KB13 (Zhong et al., 2018), NL-RX-Synth, NL-RX-Turk (Locascio et al., 2016)), sampled across k-ranges so that Np_0^k spans from 128 down to below 10^{-200} ($N{=}256$; $E{=}5{,}000$ episodes; 5 seeds each; binary vs. dense; $k = k_{\text{total}} = ns + n$ counts transition and acceptance components; correctness is exact identity of the minimal DFA, so $M = 1$). Since training runs E batches, the consistent criterion is *budget-aware* (cf. Proposition 4): budget-alive if the expected correct samples over the whole run, $H = E \cdot Np_0^k$, exceed 3; budget-dead if $H < 0.3$. Binary reward succeeds on **all** 45 budget-alive instances (225/225 runs) and on **1 of 275** runs across the 55 budget-dead instances (0.4%)—but 41 of the 45 are one-state DFAs (the re-draw off that stratum is the paragraph below), so the non-degenerate separation is $20/20$ runs on four instances against $1/275$; dense reward succeeds on all 100 (no significance test: instances share training seeds; duplicate corpus DFAs collapse to 23 distinct alive and 48 distinct dead trajectory groups, same separation, 23/23 vs. 1/48). Across all 20,901 successfully parsed instances (of 21,030 total), **74%** are budget-dead even at $E{=}5{,}000$, $N{=}256$ (76% under the per-batch criterion; the corpus is sharply bimodal, with only 0.2% in transition), explaining why Park et al. (2019) found binary equivalence insufficient. The 74% figure is specific to the from-scratch policy and output convention. On a uniform 200-instance corpus draw, 132 targets are buildable and 48 of those produce at least one correct pretrained-model probe output (36.4% of buildable targets; 24.0% of the full draw). The 22 targets classified as low-arrival by the from-scratch calculation but hit by the probe have a reported

median discrepancy of 79 orders of magnitude. These are probe observations under a different representation, not measured fine-tuning-success rates or confidence-certified learning categories (Appendix I.13).

Alive arm re-drawn off the degenerate stratum (exp45, pre-registered). Of the $20{,}901$ profiled corpus DFAs, $5{,}305$ are budget-alive under the criterion of §4.2 and $4{,}957$ of those (93%) have a one-state minimal DFA; the remaining 348 (1.7% of the corpus) are two-state (308) or three-state (40), with $k \in [4, 18]$ and H from 4.9 to $80{,}000$ (median 312; 55 in $[3, 10)$). We drew 45 uniformly from the 348 (`default_rng(450001)`, written before any run, 0 overlap with the original 100) and trained them with exp13's `run_single` unmodified—$N = 256$, $E = 5{,}000$, binary and dense, 5 seeds—with one disclosed deviation: seeds are independent per (instance, seed index) rather than shared, as this programme's random-stream rule requires. Registered P80 (success $\geq 4/5$ on ≥ 40 instances) and P81 (arrival $\geq 4/5$ on ≥ 43) both PASS at $45/45$: binary $225/225$, arrival $225/225$, dense $225/225$; the four marginal-band instances $20/20$; observed over predicted first-correct episode, median 1.00 over 225 arriving runs. No instance was trapped. The result and the corpus counts are recomputed by `experiments/exp45_nondegenerate_alive/score.py`.

Where H beats a size proxy, and where it does not (exp25, pre-registered). The corpus above spans H over two hundred orders of magnitude, so any monotone predictor separates it; the registration for this analysis said in advance that the resulting null would be stated rather than dropped. Per-instance binary success against each predictor, paired bootstrap over instances ($10{,}000$ resamples): on the 100 corpus DFAs H reaches AUC 0.999 against 0.992 for nominal k and 0.997 for the state count, and the gap to nominal k is *not* resolved ($\Delta = +0.006$, 95% CI $[-0.0004, +0.021]$). Where structure and size decorrelate by construction it is resolved and large: 1.000 against 0.763 on the product construction (Δ CI $[+0.069, +0.437]$), 1.000 against 0.438 for k_{reach} on the manufactured band, 1.000 against 0.948 and 0.500 on the two LLM grids. Two further nulls belong here rather than in a footnote: on the band the total state count also reaches 1.000 (Δ exactly 0), and on both LLM grids so does the raw probe hit count (Δ exactly 0)—where a probe exists, H adds nothing to what the probe already says, the same cancellation Appendix G reports inside the diagnostic. So the claim the design supports is the narrow one: H's structure-exactness is testable only where structure and size decorrelate, and this programme contains such cells (`experiments/exp25_predictor_decorrelation/`).

Regex-benchmark validation. The lone budget-dead success is a first-hit tail event (episode 3,037, $H = 0.004$; the uniform model expects 0.07 such runs, $\Pr[\geq 1] \approx 7\%$)—a plausible tail draw; a per-batch criterion ($Np_0^k < 1$) would instead misclassify 3 budget-alive instances ($H \in [4.9, 19.5]$, all 5/5) as dead. (Two caveats: SoftRegex rewards *equivalence*, not minimal-DFA *identity*—rescaling H by $M \geq 1$, immaterial given the orders-of-magnitude bimodality; and the regex→DFA derivation treats some metacharacters literally, so derived DFAs are training targets, not faithful NL semantics—on the 1,581 metacharacter-free (faithfully parsed) regexes the dead fraction is *80%*, so 74% is conservative; §5's regex experiment uses a faithful translator, and Synth/Turk share one regex population.)

I.7 Product Construction: Reachability Statistics

For random DFA pairs, the reachability ratio $\rho = |R_\times|/(n_1 n_2)$ decreases with DFA size. The standard deviation $\sigma(k_{\text{reach}})$ increases, amplifying the Jensen effect for larger DFA pairs.

I.8 Additional FA Problems

We briefly assess two additional FA problems not included in the main experiments:

Mealy/Moore machines would introduce heterogeneous components (transition targets and output labels have different arities), potentially creating a 2D phase transition. This is a natural next step but requires extending Theorem 3 to non-uniform component arities.

Büchi automata (for ω-regular languages) introduce acceptance conditions that change the multiplicity structure. This is a more distant extension.

Size	α	Bin. S/N	SSO S/N	Rew.
Simple (k=6)	0.0	0.077	8.94	4.8%
	0.5	0.111	8.74	7.0%
	1.0	0.178	8.58	11.2%
	2.0	0.439	8.46	23.6%
	3.0	0.814	9.15	44.0%
Medium (k=14)	0.0	0.012	7.86	0.7%
	0.5	0.014	7.82	1.0%
	1.0	0.014	7.63	1.5%
	2.0	0.030	6.79	1.7%
	3.0	0.093	5.77	4.0%
	5.0	0.796	5.31	42.1%
Large (k=30)	0.0	**0.000**	8.06	**0.0%**
	0.5	**0.000**	8.02	**0.0%**
	1.0	**0.000**	7.90	**0.0%**
	2.0	**0.000**	7.22	**0.0%**
	3.0	**0.000**	5.94	**0.0%**
	5.0	0.101	3.76	2.0%

Table 6: Gradient S/N across the training trajectory for DFA equivalence. For $k = 30$, binary S/N is exactly 0.000 at $\alpha \leq 3$: not one positive reward in 1,000 samples. The soft state-overlap control maintains S/N $\geq$ 3.7 across all 17 conditions shown; the omitted simple/α=5 run has binary S/N 3.3×10^6 (near-zero-variance artifact, reward rate 0.87) and control S/N 9.4.

I.9 Autoregressive Policy: Full Results

We train a small Transformer decoder (2 layers, 128-dim, 4 heads; $\sim$400K parameters) on the DFA equivalence task with REINFORCE, comparing against the factored policy across $k \in \{6, 8, 10, 14, 20, 30\}$ (N=256; 5 seeds per configuration, 50 seeds at k=8 and k=10[2]). The transformer uses lr=0.001 (tuned via LR sweep; lr=0.003 fails at k=8). The factored policy uses lr=0.01 and is robust across learning rates.

k	Factored bin.	Transf. bin.	Factored dense	Transf. dense
6	5/5 (100%)	5/5 (100%)	5/5 (100%)	5/5 (100%)
8	50/50 (100%)	46/50 (92%)	5/5 (100%)	5/5 (100%)
10	9/50 (18%)	7/50 (14%)	50/50 (100%)	48/50 (96%)
14	0/5 (0%)	0/5 (0%)	50/50 (100%)	48/50 (96%)
20	0/5 (0%)	0/5 (0%)	5/5 (100%)	5/5 (100%)
30	0/5 (0%)	0/5 (0%)	5/5 (100%)	5/5 (100%)

Table 7: Success rates (per-component accuracy >0.9) for factored vs. autoregressive transformer policies, all seeds included. At k=10 (50 seeds), Fisher exact $p = 0.79$ (two-sided): no detected difference; 90% CI for the success-rate difference is $[-0.08, 0.16]$. At k=8 (50 seeds), Fisher exact $p = 0.12$: transformer failure rate (8%) not significantly higher. TOST equivalence test with Δ=0.25 rejects non-equivalence ($p = 0.002$); narrower margins (Δ=0.15) do not reach significance ($p = 0.067$), reflecting limited statistical power at the transition boundary.

Optimizer fragility. At k=8 binary, the transformer's success depends critically on learning rate: lr=0.001 succeeds (3/3 seeds), lr=0.003 fails (0/3), lr=0.01 fails (0/3), lr=0.03 fails (0/3). All configurations observe correct samples at similar episodes (the same random seed produces the same first-correct timing regardless of LR); the difference is whether the optimizer can exploit the rare gradient signal without overshooting. This is a counterexample to the escape lever of §5 and we report it as one: arrival is unchanged, so the reversal is entirely in escape, and raising η makes

[2]Counts at k=8 and k=10 pool a 5-run pilot with the 45-seed extension (seeds 5–49) for both architectures; the pilot was run under a separate script whose seed labels (42–46) overlap the extension's range, but the two use different RNG streams, so no run is duplicated (verified: outcomes at the shared labels differ). Earlier drafts reported only the extension seeds.

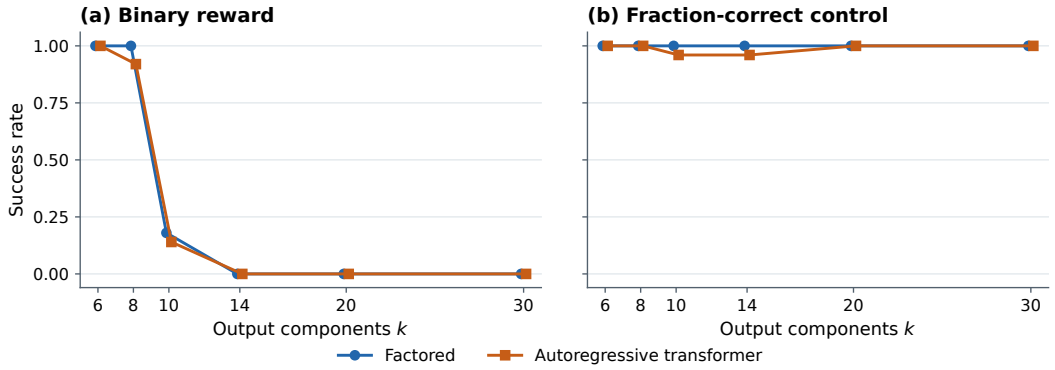


Figure 15: Factored and autoregressive success rates from the complete counts in Table 7. **Left:** Binary reward. **Right:** Fraction-correct control. The horizontal axis preserves the numerical spacing of k; exact successes/runs and the differing sample sizes are given in Table 7. At k=10, binary success is 9/50 versus 7/50 (two-sided Fisher $p = 0.79$). Markers are offset by 0.14 in k for legibility.

escape *worse* on this substrate. The lever is therefore not monotone in η—it is bounded above by whatever the optimizer can absorb per update, which for a 400K-parameter transformer receiving one rare gradient is a much lower ceiling than for 32 logits. What §5 claims is a comparison at two rates inside that ceiling, not a law.

How far the 74% transfers, measured on our own instances. For a specified reachable DFA labeling under a uniform factored policy, $q_0 = (1/|Q|)^{|Q||\Sigma|}(1/2)^{|Q|}$ includes transition choices and acceptance flags. On the six selected regex targets trained with an LLM, that from-scratch calculation labels five low-arrival at $E = 5{,}000$, $N = 256$, while four targets achieve 12/12 binary-reward training successes with probe estimates $\widehat{H}$ of 112, 178, 2,344, and 1,547. The two zero-hit targets remain unsuccessful in those runs. This small selected comparison does not imply that 74% upper-bounds the failure fraction of every pretrained model. Appendix I.13 supplies the larger uniform probe and reports disagreements in both directions.

I.10 Matched-Budget Ladders, the k-versus-c Grid, and the Two Levers

Exactly matched ladders. Restricting c to a power of a base makes $H = EN\,b^{-K}$ with $K = k \log_b c$, so every factorization of one K is an *exactly* matched instance rather than an approximately matched one. At $K = 24$ (base 2) this gives $(k, c) \in \{(3, 256), (4, 64), (6, 16), (8, 8), (12, 4), (24, 2)\}$—an 8× range in k at literally identical H—and base 3 gives two further ladders ($3^{15} = 27^5 = 243^3$; $81^4 = 9^8 = 3^{16}$). Escape is reported as a *duration* with right-censoring (Kaplan–Meier, log-rank trend), not as a fixed-budget probability: the probability version does not survive a fair-race control in which each rung is given at least the slowest rung's p_{90} duration (Spearman $+0.655$, $p = 0.158$), while the duration version gives Spearman$(k, \text{median escape time}) = -1.000$ on all three ladders, spreads of 49.5×, 35.5× and 47.1×, and log-rank $z = +18.2, +14.5, +7.2$ (Figure 2).

An analytic ϱ for a factored softmax: direction yes, magnitude no (exp26, pre-registered). Proposition 4 leaves ϱ free, and one registered attempt to close it belongs in the record with what it did and did not achieve. The registered heuristic for a factored softmax under REINFORCE used $\varrho_{\mathrm{SGD}} = \exp(\eta k(1-\bar{p})h/N) - 1$; under Adam the per-coordinate step is $\approx \eta$ regardless of gradient scale, so the h/N factor normalises away and $\varrho_{\mathrm{Adam}} = \exp(c\eta k) - 1$ with c an $O(1)$ constant, set to 1 rather than fitted. These are growth-rate heuristics, not uniform per-step bounds or the exact softmax update of Appendix A.6. We compared both against the exp24 estimator over an η sweep of more than one order of magnitude ($k \in \{6, 9\}$, $c = 3$, $N \in \{64, 256, 1024\}$; 240 runs, plus a 60,000-episode SGD extension). The frozen prediction—a ratio $\hat{\varrho}/\varrho_{\mathrm{pred}} \in [0.5, 2]$ for a majority of cells and an η log–log slope within $[0.7, 1.3]$—is met (Adam median 0.83, 80.8% in band, slope 0.81; SGD median 1.24, 65.7%, slope 0.93), but these aggregate criteria do not establish quantitative

calibration across parameter settings. Stratified, the ratio drifts systematically on every axis the registration did not score. Under Adam it falls monotonically with the rate—1.13, 0.90, 0.67, 0.43 at $\eta = 0.003, 0.01, 0.03, 0.1$, with $30/30$ cells in band at the two low rates and $8/30$ at the top—because the measured rate grows sub-exponentially in η where $\exp(c\eta k)$ grows exponentially; the in-band slope of 0.81 is that same shortfall, read as a success. Under SGD the ratio drifts with the batch, 0.61, 1.69, 5.68 at $N = 64, 256, 1024$: the derived $1/N$ is too strong, and the measured rate falls with batch size far more slowly than the derivation says. Across k the SGD ratio swings 1.94 to 0.57 between the only two values run. So this is a derivation that gets three directions right and no magnitude: we report ϱ as an empirical quantity throughout, exactly as §3.3 does, and make no claim that finite automata compute the escape stage the way they compute arrival. In exp26, the estimated compounding rate decreases with batch size under plain SGD (log–log slope -0.34 in N) and is approximately flat under Adam (slope -0.006), the contrast singled out by the registration (`experiments/exp26_analytic_rho/`).

Escape time under two optimizers. On a fixed-H ladder, the log–log slope of post-arrival escape time with k is -1.57 under plain SGD and -1.88 under Adam, both at Spearman -1.000. These are escape-time slopes, not fitted growth-rate slopes; the observed direction is not exclusive to Adam.

Does the arrival law need c to be a power of a base? No; powers exist only so that H matches exactly across rungs. On two dense scans with arbitrary integers—$k = 4$ with $c \in \{32, 36, \ldots, 113, 128\}$ and $k = 6$ with $c \in \{10, 11, \ldots, 22, 25\}$, in which 10 of 13 and 10 of 11 values are not powers of two—arrival is ordered by H across the 26 and 22 (cell, budget-arm) combinations at Spearman $+0.996$ and $+0.994$, and at $+0.993$ on each scan when the power-of-two values are dropped. The parameter-free prediction $1 - e^{-H}$ tracks the observed arrival rate to a mean absolute error of 0.022 and 0.019 (worst cell 0.095 and 0.088, against a binomial standard error of 0.03–0.05 at 100 seeds per cell). The restriction is a scope choice for the ladders, not a generator of the effect.

Separating k from c. A fixed-H ladder holds $k \log c$ constant, so raising k *is* shrinking c and the ladder cannot distinguish them. We therefore executed the full 8×5 grid with $k \in \{2, 3, 4, 6, 8, 12, 16, 24\}$ and $c \in \{2, 4, 8, 16, 64\}$, 60 seeds per cell, from a start conditioned on arrival. The scorer admits a cell only when a Kaplan–Meier median exists, at least half its runs escape, and at least one escaped run has a usable compounding-rate estimate; 27 cells pass and 13 do not. The post-hoc interior is the 19 admitted cells with $k \log_2 c \leq 16$. Table 8 lists these analysis sets. The shortcut preserves the uniform policy but also matches optimizer age: for each run it draws $T_A - 1$ from that cell's geometric arrival law (capped at 50,000), materializes Adam with zero first and second moments, and sets its step counter to that draw before replaying a forced-arrival batch. This mirrors the honest runner, which calls Adam on every all-failure episode even though the centered reward gradient is zero. Adam's bias correction can scale the first productive update by up to $3.16\times$. For a unique target under a uniform factored policy that remains fixed before arrival, $q_0 = c^{-k}$. With a fixed correctness predicate and independent samples across and within episodes, the one-based first-arrival episode satisfies

$$T_A \sim \mathrm{Geom}(p_A), \qquad p_A = 1 - (1 - c^{-k})^N, \qquad T_A \in \{1, 2, \ldots\}.$$

Thus $\mathbb{E}[T_A] = 1/p_A$; the approximation $\mathbb{E}[T_A] \approx c^k/N$ requires $Nc^{-k} \ll 1$. There are $T_A - 1$ all-failure episodes before first arrival. This count equals the number of preceding optimizer calls only when the runner executes one call per failed episode; first arrival and the first productive update need not coincide under centered advantages. The waiting-time distribution varies with both k and c, so a constant optimizer age does not preserve the pre-arrival history of each cell. We instead draw each run's optimizer age from its own arrival law; validated against runs that pay the real wait, this gives Kolmogorov–Smirnov $p = 0.93$, 0.99 and 0.23 on three reachable cells (a fixed constant fails at $p < 10^{-3}$).

All three of this grid's pre-registered predictions **failed as written**, and they failed because each posits a *global* log-linear law. In the post-hoc interior ($n = 19$ cells), the fitted escape-time coefficient on $\log k$ is -0.090 (95% CI $[-0.155, -0.024]$), while the coefficient on $\log c$ is close to zero (-0.002, $[-0.044, +0.040]$). The latter estimate does not establish an absence of c-dependence beyond this subset. Sliced along iso-p_0 lines, the direction replicates everywhere while the magnitude does not: the spread across k is $1.0\times$ at $K = 4, 6, 8$, then $1.2\times$, $1.2\times$, $1.9\times$, and $19.1\times$ at $K = 24$. The large numbers are a *boundary* phenomenon, and the fixed-H ladders sit on that boundary by construction. We therefore do not claim "escape is governed by k" as a global law. The ladders

are also narrower than Proposition 4, whose premises are general: every rung was run on a factored policy with uniform initialization at a single batch size, so we do not know how the escape-time law moves with $M > 1$, non-uniform initialization, an autoregressive policy, or N.

Table 8: **Executed grid and reported analysis sets.** All 40 (k, c) cells were run with 60 seeds each. The scorer admits 27 cells under the stated gate and excludes 13. The 19-cell interior is the post-hoc subset with $k \log_2 c \leq 16$. The other eight admitted cells form the boundary stratum. Entries list component arities c at each k; a dash denotes an empty set.

k	Admitted c	Interior c	Excluded c
2	2, 4, 8, 16, 64	2, 4, 8, 16, 64	–
3	2, 4, 8, 16, 64	2, 4, 8, 16	–
4	2, 4, 8, 16, 64	2, 4, 8, 16	–
6	2, 4, 8, 16	2, 4	64
8	2, 4, 8	2, 4	16, 64
12	2, 4	2	8, 16, 64
16	2, 4	2	8, 16, 64
24	2	–	4, 8, 16, 64

The compounding rate on the same grid. The fitted coefficients of $\log(1+\varrho)$ on the interior subset of the compounding-rate analysis (19 of 27 analyzed cells, stratified after the fact) are $+0.045$ (95% CI $[+0.032, +0.059]$) on $\log k$ against $+0.017$ ($[+0.008, +0.026]$) on $\log c$. Within this post-hoc subset, the fitted coefficient on $\log k$ exceeds that on $\log c$. The registered fit was global, P43, and failed (Appendix I.12); the $\log k$ coefficient is $+0.045$ in *both* strata, while the magnitude of the escape-time spread differs. The plain-SGD ladder above also exhibits the escape-time trend, so its direction is not exclusive to Adam.

Why the ladder could not be transplanted to the LLM. Holding H fixed shrinks the alphabet as k grows, so the matched-ladder cells the LLM would need are exactly the cells where a 0.5B model stops following the output format: a 220-prompt probe finds 1% of answers inside the declared alphabet at k=20, c=2, with $\bar{p}$ below the random floor. A ladder built there would compare "hard to explore" against "cannot emit a scorable output", which is why the LLM lever arm uses instances fixed by an escape-limited rule instead (§5).

Optimizer generality. Repeating the $K = 24$ ladder with SGD, the registered predictions also failed—driven entirely by one arm. At $\eta = 1000$ every rung finishes in 3–4 episodes at final $\bar{p} = 1.0000$ with ϱ of order 10^2 and no k-trend (slope -0.091). These rapid jumps make the gradual-compounding interpretation uninformative; their large gains alone do not establish a violation of Proposition 4's uniform multiplicative bound. In the two gradual-compounding arms, Adam at $\eta = 0.01$ gives slope -1.884 and SGD at $\eta = 100$ gives -1.569, both at Spearman -1.000, a ratio of $1.20\times$. Adam enters the same jump regime early: pooled median escape time falls $19.6\times$ at $\eta = 0.03$ and $741\times$ at $\eta = 3.0$, at 100% convergence throughout.

The two levers. Splitting the ladder runs on the first-signal episode separates the interventions. The learning rate leaves arrival at $30.7\% \to 30.2\%$ while lifting escape given arrival from 94.8% to 99.3% ($p = 3.9\times10^{-5}$); an $8\times$ budget lifts arrival from 30.7% to 49.3% and also lifts escape where escape has room ($45.2\% \to 88.7\%$, $p = 1.5\times10^{-17}$, on the ladder whose base escape rate leaves room for it). So the learning rate is escape-only and the budget acts on both. This is why the same learning-rate arm reads as a null ($+0.9$ pp, $p = 0.60$) on unconditional success: only 5.2 pp were available on escape and 70% of runs never arrived.

A pre-registered replication on a disjoint seed stream paired the learning-rate arms with shared randomness. Across $2{,}400$ paired runs, the first-signal episode matched in *every* case (0 mismatches), consistent with pre-arrival policy stasis. Exact stasis follows from the update conditions in Remark 2; the timing agreement provides a consistency check. Escape given arrival rose (unconditionally, pooling both budgets, $11.0\% \to 15.5\%$) $50.5\% \to 93.9\%$ at $E = 5{,}000$ ($p = 3.0\times10^{-12}$) and $76.2\% \to 98.6\%$ at $E = 20{,}000$ ($p = 1.6\times10^{-17}$), pooling to $69.6\% \to 97.4\%$—91% of the available headroom, $p = 7.5\times10^{-28}$—with the larger gain at the budget with the larger headroom, as the ceiling account requires.

I.11 LLM GRPO EXPERIMENT: DETAILS

Grid summary. Qwen2.5-0.5B with GRPO on position-constrained generation, $k \in \{4, \ldots, 16\} \times 3$ instances, binary vs. graded reward, 3 seeds per cell (108 runs), with the probe $\hat{P}_0$—though not the success threshold—frozen before training and $\hat{H} = EG\hat{P}_0$ from a pooled probe (Figure 3; Appendix I.11): the binary outcome is seed-stable in every cell ($18/18$ unanimous), monotone in the pooled probe hit count (0 hits: $0/6$ instances; 1: $1/2$; ≥ 3: $10/10$; permutation $p < 10^{-4}$), and set by the pre-measured count rather than by k—$3/3$ on all ten instances with $\hat{H} \geq 14$, $0/3$ on all six zero-hit instances (a 95% upper bound of 17: consistent with, not certifying, budget-death).

Scope of the substrate. The factored testbed draws $N = 256$ samples per episode while GRPO draws $G = 64$ per group. H is always defined with whichever the run actually draws, but the two substrates are never matched on that number, so no constant measured on one is expected to transfer numerically to the other. The task itself is a combination lock with nothing to compose or transfer, chosen because k and p_0 are settable; it is an instrument, not a proxy for reasoning.

Which lever rescues a trapped LLM instance. Instances were frozen by a rule applied before the arms were run: at the baseline (E=300, η=10^{-6}) an instance qualifies if at least one seed produces a positive reward and no seed succeeds—arrived but trapped, by construction, so this is a test of the escape lever and not of arrival. Seven instances qualified. The four arms cross $\eta \in \{1, 3\} \times 10^{-6}$ with $E \in \{300, 1200\}$, six seeds each, and the arms share a random stream per (instance, seed) so that arrival is identical across them; a built-in audit confirms it (42/42 first-signal steps agree wherever both arms reach that step).

instance	E=300, η=1	E=300, η=3	E=1200, η=1	E=1200, η=3
k=6, t6	0/6	6/6	4/6	6/6
k=9, t0	0/6	6/6	6/6	6/6
k=9, t3	0/6	5/6	5/6	6/6
k=9, t4	1/6	6/6	6/6	6/6
k=12, t2	0/6	0/6	0/6	2/6
k=16, t0	0/6	6/6	6/6	6/6
k=16, t2	0/6	4/6	0/6	6/6
instances rescued	1/7	**6/7**	5/7	7/7
runs succeeding	1/42	**33/42**	27/42	38/42

The instance counts run $6/7$ for tripling the learning rate against $5/7$ for quadrupling the budget (78.6% against 64.3% of runs, at a quarter of the compute), with one discordant pair: that orders nothing at this n, and we report the two arms descriptively, as §5.3 does. The two are complementary—together they rescue all seven. The one instance no single lever rescues, k=12 t2, is the one whose arrival is marginal rather than complete (2/6 seeds arrive at baseline), which is what the two-stage account predicts: a lever aimed at escape cannot help a run that has not arrived. This is the LLM counterpart of the factored-policy result in Appendix I.10, and it was pre-registered with the opposite expectation—that the budget would dominate—which is how the ceiling artifact in the earlier fixed-budget comparison came to light.

The scope of this table is narrow and should be read as such: it is one model (Qwen2.5-0.5B) and one instance-selection rule, on the position-constrained task family. The regex task now carries the same arm (Appendix I.16), so both task families have been measured; neither orders the two levers at its own n, both are at 0.5B, and we claim no ordering for RLVR in general.

The collapse replicates across three estimators but at one model scale (0.5B), so it is a measurement and not yet a general theory. A pre-registered control with REINFORCE+baseline and σ-free GRPO fails 12/12 on the two cells where binary wins (graded reward plateaus at 0.76–0.83): the collapse is not a normalization artifact.

Figure 3 in the main text reports the position-constrained grid. The binary outcome is monotone in pooled probe hits in this grid: all instances with at least three hits succeed, all zero-hit instances fail, and the marginal zone splits one–one. The graded arm fails in 25 of 27 runs at $k \geq 10$ (seven of

nine instances fail in all three seeds; two $k = 10$ instances split across seeds). The setup and probe accounting follow.

Setup. Prompts list all k position constraints ("Position i must be the letter 'c_i' "); the target string is fully determined, so $M = 1$ and p_0 is directly measurable. Transparent single-file GRPO: group-relative advantage $A_i = (r_i - \bar{r})/\sigma$, one update per group (importance ratio $\equiv 1$, clipping inactive), $\beta = 0$ (no KL anchor), AdamW, $\mathrm{lr} = 10^{-6}$, temperature 1.0, bf16. Every LLM run in this paper is a full fine-tune: all parameters of the policy are in the optimizer, and no adapter, LoRA, or other parameter-efficient method is used at either model scale. All-identical-reward groups ($\sigma = 0$) are skipped, per Observation 8. Success = joint accuracy ≥ 0.9 over 64 generations sampled after training at the training temperature (1.0).

Probe accounting and what was fixed in advance. $\hat{P}_0$ pools two independent 2,048-sample probes per instance (four late-added instances have one); at the resolution floor ($P_0 \sim 5 \times 10^{-4}$), single instances flip between 0 and 1–2 hits across probes, which is why we report pooled estimates with Poisson CIs rather than point calls. On a 4,096-sample pooled probe $\hat{H} = EG\hat{P}_0$ is therefore a lattice of multiples of 4.7, and the main text's "$\hat{H} \geq 14$" cell is exactly the set of instances with ≥ 3 probe hits. What was fixed before training is the *measurement* ($\hat{P}_0$) and the task grid; we did not pre-register a numeric success threshold—the design document's per-batch heuristic ($G\hat{P}_0 \gtrsim 1$) is inconsistent with the budget-aware theory (it would predict failure everywhere here), and the post-hoc rule $\hat{H} \geq 1$ scores 12/14 on the first probe alone. The main text reports the monotone association: no threshold choice is needed for it; an exact permutation trend test (Kendall statistic between pooled probe hits and per-instance success fraction, 10^5 label permutations over the 18 instances) gives $p = 6 \times 10^{-5}$, with the caveat that the three training seeds are shared across instances, so instances are exchangeable only up to that shared-seed dependence. The coarse 64-sample screening also covered $k \in \{20, 24\}$ (all zero hits); those cells were not trained, to spend GPU budget near the boundary—so the grid itself concentrates instances away from (below) the deep-dead interior, and "16 of 18 outside the marginal zone" partly reflects that selection.

Escape is bimodal. Successful binary runs (33 of 54) have 146–276 productive (non-skipped) groups out of 300—once escape begins, hits compound; failed runs have 0–10, and their mean reward never exceeds $1/64$. Isolated hits do not escape: twelve failed runs across the 54 receive first signals (productive-group counts 1–10) that never cascade, matching the birth-process picture of Proposition 4 at $\mathrm{lr} = 10^{-6}$, where a single hit's multiplicative gain ϱ is, on the productive-group counts, insufficient to cover $\log(1/(GP_0))$ in the remaining budget. **Marginal-pair lr ablation.** Raising ϱ directly tests this: the trapped k=16 instance ($\hat{H}$=4.7; 5–10 isolated productive groups at $\mathrm{lr}\, 10^{-6}$) succeeds 3/3 at $\mathrm{lr}\, 3{\times}10^{-6}$ (66–78 productive groups) and 3/3 at 10^{-5} (16–21, faster convergence needs fewer), while k=8 stays 3/3 at all three lr values. Within each seed the first-signal step is *identical* across lr (112/16/25 at k=16)—mechanically so: all-failure groups skip the update, so the pre-hit sampling stream is seed-determined; this verifies the zero-gradient-stasis premise of Proposition 4(c) at the implementation level rather than testing it independently. The empirical content is the escape side: the same first hits that never cascade at $\mathrm{lr}\, 10^{-6}$ convert to 6/6 success at larger ϱ (Part d). The realized alive threshold is therefore lr-specific: at $\mathrm{lr} \geq 3{\times}10^{-6}$ it moves down to include $\hat{H} \approx 5$; the zero-hit training instances remain unchanged across the tested rates under the audited pre-arrival stasis condition.

Partial credit and final correctness. The position-constrained control uses a scalar fraction of correct constrained positions; the regex control uses DFA agreement on a balanced test set. On the position task, the graded control fails in 28/54 cells and binary reward in 21/54. Near-miss outcomes include one position pinned at 0% and vanishing group variance. Across the 54 paired cells, partial credit recovers one binary failure and loses eight binary successes; the latter come from three instances, so seed-level tests do not establish a general instance-level disadvantage. On one selected regex target, the graded control restores gradient in every seed but ends at joint accuracy 0.0. These are counterexamples to a universal partial-credit remedy, not evidence that all scalar partial-credit rewards share the same behavior. The same scalar fraction-correct construction succeeds on the factored and small-transformer controls (Appendix H).

Near-miss mode collapse under graded reward. Diagnostic retraining of a failed graded run (k=6, target `fifxfk`) shows the mechanism: the group's modal output climbs to 5/6 correct positions while the wrong position's accuracy *decreases* (0.17 → 0.0)—early in training, samples correct at

that position happened to be wrong elsewhere, so the sequence-level advantage pushed it down; by step 250 the policy is deterministic on `fiixfk` (group std $= 0$) and frozen. Ablations on this instance, 3 seeds per cell: a KL anchor to the initial policy fails at both $\beta = 0.02$ and $\beta = 0.05$ (final joint accuracy 0.0, 6/6 runs); tripling the learning rate fails (0/3); raising the sampling temperature to 1.2 escapes in *every* run (6/6, joint accuracy ≥ 0.98 at evaluation temperature 1.0, with and without the KL anchor)—temperature is the causal variable: the trap is an exploration failure, not an artifact of the KL-free objective (a mild entropy bonus, $\lambda = 0.01$, reopens diversity slightly but does not escape). The same interventions do not generalize deeper into the dead zone: on collapsed k=10 and k=12 instances, neither the KL anchor at temperature 1.0 nor temperature 1.2 without the anchor escapes within 300 steps (joint accuracy ≤ 0.02 in all 12 runs, 3 seeds per cell, evaluated at temperature 1.0)—exploration-based escape is available only near the boundary. Binary reward is structurally immune to this trap: it performs *no* update until an exact hit occurs, so it cannot be steered into a near-miss mode; its risk is starvation, not misdirection. Collapse is itself stochastic: two k=10 graded cells split across seeds (1/3 success each), while eight cells fail and eight succeed unanimously. One threshold sensitivity: the two failed runs in one split k=10 cell end at joint accuracy 0.81, just under the 0.9 success criterion; every other graded failure ends at joint accuracy ≤ 0.02, so the 28/54 count moves by at most two runs under any reasonable threshold. The dependence-factor estimates from these instances ($\gamma \approx 1.9$–4.9 where estimable) are consistent with the exp11 probe on a $16\times$ larger model.

RLVR on actual regex generation (exp16). Six instances from the §4.2 benchmarks (KB13, NL-RX-Synth); prompt = the NL description plus the permitted alphabet; output = a regex; reward $= 1$ iff the output parses and its minimal DFA over the fixed per-instance alphabet equals the target's. (The §4.2 corpus derivation treats some regex metacharacters literally—a parser limitation harmless there, where the derived DFAs are themselves the training targets—so exp16 recomputes faithful targets with a documented, unit-tested translator; equivalent surface forms provably collapse to one language.) The limitation's scope is worth printing: $8{,}175$ of the $21{,}030$ corpus targets (38.9%) contain `&` or `~`, so the corpus figure of §4.2 is computed over derived DFAs that are the training targets by construction, and is not a claim about the benchmarks' intended languages. Same GRPO protocol as the position-constrained grid; $\hat{P}_0$ from 2,048-sample probes; alive/dead calls and the success criterion frozen in `predictions_pre_training.json` before any training. Results: the four alive instances ($\hat{H} \in [112, 2344]$; targets like `.*(dog).*`, `[0-9]`) succeed 12/12 (final joint accuracy 0.97–1.0; 192–245 productive groups; first signals within or near the $3/(G\hat{P}_0)$ bound); the two dead instances (0 probe hits; conjunctive targets—"contains dog and truck"—of 16–20 states) fail 0/6 with zero equivalence hits in all $19{,}200$ training samples per run and 0 productive groups. The marginal band (1–2 pooled probe hits) is empty partly by construction: the 128-sample screen cannot resolve $P_0 \sim 5 \times 10^{-4}$, so any candidate with nonzero screening hits lands deep-alive—exp16 therefore replicates the *separation and starvation mechanism* at the extremes, not the location of the realized boundary (which only the position-constrained grid resolves); candidate screening and tie-breaks among 1-hit candidates are recorded in `candidates.json` and the design document. $\hat{P}_0$ on this task also conflates combinatorial deadness with syntax compliance and expressibility; memorization of the public 2016–2018 source corpora was tested directly (contamination probe below) (conjunctive descriptions invite unsupported lookahead syntax): the monotone-in-$\hat{P}_0$ claim is unaffected, but a k-causal reading is not licensed here. Parse rate is a separate axis: dead instances parse at 9–38% initially (the model emits `dog|truck` for a conjunction) and binary training leaves parse rate unchanged (no gradient ever flows), while alive training lifts it to ≈ 1 as a side effect. A graded control (DFA membership agreement on a balanced 200-string test set, 3 seeds each, unanimous): the alive instance succeeds 3/3 (joint accuracy ≥ 0.98); the dead instance regains gradient in every seed (245–275/300 productive groups; agreement $0.24 \rightarrow 0.55$; parse $\rightarrow 0.98$) but ends at joint accuracy 0.0 in all three—graded reward un-starves the gradient yet lands on a deceptive optimum, the same dissociation as the position-constrained grid. Test-set sampling for the graded control was amended before running it (uniform Σ^* sampling yields no positives for sparse languages; fixed by sampling from the target DFA and its complement, dated addendum in the design document).

Both deaths on the LLM, pooled (exp34 re-analysis). Pooling the 128 binary GRPO runs of the position-constrained grid (exp14), its budget extension (exp27), the regex grid (exp16) and the Qwen3-8B cell (exp17), and classifying each run from its own log: 113 arrive, 94 escape, 19 arrive and stay trapped, 15 never arrive; the classification obeys its invariants on every run (0 violations).

The k-stratified table below covers the 110 runs from exp14, exp27 and exp17, with 101 arrivals; exp16's 18 regex runs carry no k and are reported separately, 12 arriving and satisfying the final criterion and 6 never arriving, so that $110+18=128$ and $101+12=113$ recover the pooled totals. A 10-step pre-flight run kept under exp17's `results/smoke/` (seed 999, outside the registered grid) was swept in by the re-analysis script's recursive glob and counted as a training run in earlier drafts; excluding it moves the pooled totals from $129/114/20$ to the figures above. Escape given arrival by k for the 110-run subset:

k	runs	arrived	escape given arrival
4	9	9/9	1.00
6	12	12/12	0.83
8	16	16/16	1.00
9	18	18/18	1.00
10	18	14/18	0.57
12	15	13/15	0.69
16	16	13/16	0.46
20	3	3/3	1.00
28	3	3/3	1.00

The fall from 0.83–1.00 at $k \leq 9$ to 0.46 at $k = 16$ is the direction of Figure 2, but this grid holds c fixed, so H falls with k and the k-effect and the H-effect are confounded here; it is reported as the count of both deaths on the LLM, not as a k-law. Generated from `experiments/exp34_llm_split/results/llm_split_reanalysis.json`.

Transition-band test on the LLM (exp23, pre-registered). Fifty-six position-constrained instances ($k \in \{6, \ldots, 13\}$, several constraint counts each) were probed with 32,768 samples; seventeen landed in $\hat{H} = EG\hat{P}_0 \in [1, 10]$, and the six nearest the pre-registered targets $\{1, 2, 3, 5, 7, 10\}$ were frozen before training (the other eleven were not trained) and trained with the exp14 GRPO protocol, 3 seeds each.

k	probe hits	$\hat{H}$	success	first signal seen	$1 - e^{-\hat{H}}$
10	2	1.17	0/3	2/3	0.69
6	3	1.76	0/3	3/3	0.83
9	5	2.93	0/3	2/3	0.95
9	9	5.27	0/3	3/3	0.99
9	13	7.62	0/3	3/3	1.00
9	17	9.96	3/3	3/3	1.00

Pre-registered verdict: FAIL for the first-signal prediction ($\rho = 0.66$; 1/6 calibrated). The first hit is not the binding step: in the five failing instances a first signal occurs in 13 of 15 seeds, but productive groups remain rare afterwards (1–7 of 300 steps in 13 runs) and joint accuracy ends at 0.000; only the $\hat{H} = 5.27$ instance sustains 78–97 productive groups in two seeds and reaches 0.28–0.42, while the successful instance has 96–128. Within this grid the transition is sharp (0/3 at 7.6, 3/3 at 10), but it sits where the post-hit cascade, not the first hit, is satisfied; the $\hat{H}$ values here (32,768-sample probes) are not on the same scale as the 4,096-sample grid above, where one $\hat{H} \approx 4.7$ instance escapes.

Non-GRPO graded-reward control (pre-registered). On the cells $(k{=}6, t{=}0)$ and $(k{=}12, t{=}0)$, where binary reward succeeded 3/3 and graded reward failed, we re-ran graded reward with (a) REINFORCE and a moving-average baseline (no group normalization) and (b) GRPO with advantage $r - \bar{r}_{\text{group}}$ (no σ division, no $\sigma{=}0$ skip), same G, E, learning rate, temperature, no KL, 3 seeds each. All 12 runs end at joint accuracy 0.000 while the graded reward plateaus at 0.76–0.83: the policy optimizes partial credit and never locks in the exact string, under every estimator. Pre-registered verdict: GENERAL (failure in $\geq 2/3$ seeds under both estimators in both cells); the $\sigma{=}0$ mechanism described for GRPO is therefore one route into a collapse that does not depend on it.

Contamination probe (pre-registered). To test whether the alive/dead split reflects memorization of the public benchmark targets, we generated six fresh targets from random lowercase literals absent from KB13 and NL-RX (three matched to the alive instances by minimal-DFA size ± 1 and structural type—literal containment, character class; three matched to the dead instances—conjunctive

containment, 18 states), with NL descriptions from the same template family, and probed $\hat{P}_0$ with 2,048 samples on Qwen2.5-0.5B in the same session as a re-probe of the benchmark instances. Benchmark-alive hits: 8, 21, 241, 176 (of 2,048; the 241 is a fixed-length target `.{4}` with no fresh counterpart); fresh-alive: 1, 8, 176; benchmark-dead: 0, 0; fresh-dead: 0, 0, 0. The pooled benchmark-to-fresh alive ratio is 1.81 (pre-registered threshold for "no evidence": ≤ 3; "contamination likely": > 10), with per-pair ratios from 1 (character class) to 8 (single-literal containment); all dead-matched targets are at zero hits. Verdict under the pre-registered rule: no evidence of contamination, with the per-pair spread above.

Cross-scale cell: Qwen3-8B (exp17). Same task, identical instances (shared generator seeds), frozen criterion as above; protocol deltas, all disclosed in the design document: 8-bit AdamW, chunked logprob backward (gradient-identical), Qwen3 thinking mode off. Probe (2,048 samples $\times$ 12 instances): the 8B model's $\hat{P}_0$ is orders higher than 0.5B's, as pre-training softening predicts, and remains instance-dependent rather than k-monotone—$k{=}10, t{=}0$ has zero probe hits (the model systematically drops one letter: 1,143/2,048 outputs are the 9-letter truncation of the 10-letter target, so $\hat{P}_0$ here too conflates combinatorial deadness with format compliance; the monotone claim is unaffected, a k-causal reading is not licensed) while $k \in \{12, \ldots, 28\}$ instances are alive. No probed instance has 1–2 hits and all trained alive instances are deep-alive ($\hat{H} \geq 422$), so the 8B cell tests the alive side and the probe calibration, not the dead-zone location. Training grid (6 instances $\times$ 3 seeds): all five alive predictions confirmed 15/15 (joint accuracy ≥ 0.97; first signals within or near the $3/(G\hat{P}_0)$ bound—one of 15 runs at step 2 vs. bound 1.2). The frozen dead call (0 hits $\rightarrow$ 0/3) was *refuted*: 2/3 seeds succeeded (first signals at steps 13/128/228; the failed seed's single hit never cascaded—the trapping signature). Refutation accounting: the three first-signal times give $\hat{P}_0^{\mathrm{MLE}} \approx 1.3{\times}10^{-4}$ ($\widehat{H} \approx 2.4$, transition band, $\sim 4\times$ below the probe's $1/2048$ resolution); a 2,048-sample probe sees zero hits with probability 0.77 at that rate, and the Poisson 95% upper bound for $0/2{,}048$ is $H \leq 35$ (two-sided convention; one-sided gives $H \leq 28$)—the dead call was never certified, only consistent. (Provenance note: the design document's probe-extension addendum carries an erroneous self-stamp; filesystem timestamps establish addendum $<$ extended probe (10:56) $<$ freeze (10:57) $<$ training (10:59), and a dated correction is recorded in the design file.) The design lesson is quantitative: probes must resolve P_0 down to $\sim 1/(EG)$, i.e. probe size on the order of the training budget, before "zero hits" certifies budget-death; at 0.5B this gap did not bite because true P_0 sat orders below both scales (zero hits in $3 \times 19{,}200$ training samples), at 8B pre-training lifted this instance into the gap. Training-side γ from the probe (joint vs. per-position accuracy over parsed outputs) is *window-sensitive*: within exp11's probed range $K \leq 15$ ($n{=}5$ instances) the median is 1.46, *below* the inference CI 2.3 $[1.5, 3.7]$; over all $k \geq 6$ ($n{=}10$) it is 4.4, driven by deep-k instances outside that range. Per-instance values overlap the inference estimates and $\gamma > 1$ in all 10 estimable $k \geq 6$ instances, but the two estimators agree in direction, not in a single number—recorded as a growing-dependence-with-k tension, unresolved.

Interpreting $\gamma > 1$. For $0 < q, \bar{p} < 1$ under a fixed scoring distribution, the chain rule gives $q = \prod_i \Pr(C_i \mid C_1, \ldots, C_{i-1})$. If every conditional correctness probability is at least its marginal and one is strictly larger, then $q > \prod_i p_i$. With heterogeneous marginals this alone need not imply $\gamma > 1$, which requires the stronger condition $q > \bar{p}^k$ (Appendix C). The empirical values concern the stated parse-conditioned estimators and retain their sampling and window sensitivity. Cross-position transfer of correctness information is a possible explanation to investigate; these comparisons do not identify its representational mechanism or a causal effect of pretraining.

I.12 Every Registered Prediction, and What Happened to It

The word "pre-registered" appears throughout this paper, and a reader is entitled to ask how many registered predictions there were in total and how many of them survived. This appendix answers that. The counts below are produced by a script over the scoring artifacts (`experiments/ledger/`), not transcribed: the programme stores verdicts in nine different shapes, and our own first attempt at this table read only one of them, found 7 of the 15 experiments, and undercounted the failures by more than half.

	what it asked	registered	pass	**fail**	other
exp 27	Is exp23's failure just budget truncation?	P1–P3	1	**2**	—
exp 28	The oracle ladder: information, not density	P4–P7	1	**1**	2
exp 29	Why did the counterexample fail?	P8–P10	0	**0**	3
exp 30	Two deaths, two prescriptions	P11–P13	1	**2**	—
exp 31	Locating the prescription's boundary	P14–P17	2	**2**	—
exp 32	Separating arrival from escape at fixed H	P18–P23	4	**2**	—
exp 33	Is the split real, or a property of one grid?	P24–P27	4	**0**	—
exp 35	Does the prescription transfer to the LLM?	P33–P36	2	**2**	—
exp 36	The mirror test: hold k fixed, vary H	P37–P40	2	**2**	—
exp 37	Is escape governed by k or by c?	P41–P44	0	**4**	—
exp 38	Is the escape law an Adam artifact?	P45–P47	0	**3**	—
exp 39	Is the CPU lr null a range artifact?	P51–P54	2	**2**	—
exp 40	Replicating the lr effect, disjoint stream	P55–P57	3	**0**	—
exp 41	The lever arm on the regex task	P58–P63	5	**0**	1
exp 42	The unselected draw	P68–P73	3	**2**	1
exp 43	Does the corpus figure survive a real model?	P64–P67	3	**1**	—
exp 44	The mixed batch and the KL anchor	P74–P79	4	**1**	1
exp 45	Alive arm re-drawn off the degenerate stratum	P80–P84	2	**0**	3
exp 46	Partner swap, and GRPO's own KL estimator	P85–P90	4	**1**	1
exp 47	The width cell: 8 and 30 prompts per step	P91–P96	1	**2**	3
exp 48	The thirty-prompt null: step size or budget	P97–P99	2	**0**	1
total	21 experiments	91 predictions	**46**	**29**	16

Twenty-nine of the seventy-five decided predictions failed as written — 39%**.** The ledger is not a hypothesis-testing family and is not corrected for multiple comparisons: each prediction targets a distinct pre-specified question. The ledger combines statistical tests, deterministic checks, and descriptive thresholds, so the 46 passes do not admit a single false-positive interpretation. The 29 failures include both limited-power outcomes and incorrect predictions; passing a registered criterion is not itself evidence that every broader interpretation is supported. For the one family the paper leans on, the eight registered escape-lever tests (P77, P78, P85, P87, P91, P94, P97, P98), Holm at 0.05 keeps only P87 and Benjamini–Hochberg at $q = 0.05$ keeps five of eight (P77, P85, P87, P97, P98; `experiments/ledger/multiplicity.py`). One UNTESTABLE hides a null rather than thin data: P95 is $0/39$ escape at both rates, the programme's largest zero, routed out of the decided denominator by the discordance floor and decided afterwards by exp48. Twenty-nine is the number this appendix exists to publish. Two patterns cover seven of them; a third kind of failure, in the other namespace and so outside this count, mattered more than any of them. The twenty-two that remain are spread over thirteen experiments; we name below the two of those that bear on the thesis rather than leave them inside a count:

(i) A prediction posed as a global law that holds only stratified. exp37's P41–P43 and exp38's P45–P47 — six of the twenty-nine — each assumed a single log-linear law in $(\log k, \log c)$ or across optimizers. Stratified, the picture is clean (Appendix I.10); globally, it is not, and we report the registered failure rather than the stratified rewrite.

(ii) Our own instrument. exp37's gate P44 failed because `run_grid.py` paid the arrival wait without stepping the optimizer, so it compared warm against cold rather than conditioned against unconditioned. The gate is recorded FAIL, the instrument was repaired, and the three predictions it gated then failed anyway on the repaired instrument.

(iii) A registered prediction that the data reversed, which is the useful kind—and is an R-series entry, counted in Table 9 rather than in the twenty-nine. exp23's transition-band test (Table 9) located the LLM boundary from arrival alone and was wrong; that failure is what forced the separation of arrival from escape, and hence the structure of this paper (§5).

The twenty-two the patterns do not cover. They are P2–P3, P4, P11–P12, P16–P17, P18–P19, P33–P34, P38–P39, P52–P53, P66, P68, P70, P78, P86, P91 and P94, all in the table above; most are prescriptions that a later grid refuted (P11–P12, P16–P17, P33–P34) or thresholds registered as majorities that came out pluralities (P68, P70), and two bear directly on the thesis. exp32's **P18 and P19** registered the two halves of §5.1—arrival flat in k, success monotone in k—on a 60-seed base-budget arm and both failed, P18 through a success proxy that conflated the two deaths it was meant to separate. P18's own falsification clause asked that H then be demoted from *the* predictor

to *a* predictor; what the figure plots instead is exp33's independently registered 100-seed replication on a disjoint stream, which isolates arrival by first-signal episode and passed (P24–P27), and §5.1 now prints both verdicts. exp44's **P78** and exp46's **P86** are the lever's own failures in the deployed regime, reported in §5.5.

Four entries need their own line, because a bare count would misrepresent them. exp28's P5 and P6 are recorded *not testable*, never FAIL: both are contrasts against an effect that the same experiment found to be zero, so there was nothing to attribute. exp41's P63 returned a registered null in the opposite direction to the prediction and is reported as a design limit (Appendix I.16). exp29's P8–P10 were frozen and the experiment was *never launched* — its pre-registration says so in its own header — so they have no verdict and are counted separately rather than quietly dropped. And the numbers P28–P32 and P48–P50 were never assigned to anything; we note this so that a reader who follows the numbering does not conclude that eight results are missing.

		what it asked	verdict
R2	exp 19	Matched-estimand γ_{arch} for the random-init transformer	**PASS**
R3	exp 20	Is the graded-reward collapse GRPO-specific?	**GENERAL**
R10	exp 21	Does memorization drive the regex split?	**NO-EVIDENCE**
R8'a	exp 22	Manufactured transition band on the factored policy	**PASS**
R8'b	exp 23	The same band on Qwen2.5-0.5B GRPO	**FAIL**

Table 9: Five additional checks were registered four days before the P-numbered programme. They use separate identifiers and three-way outcome rules, so their outcomes are reported separately from the table above. R8′b is exp23, whose failure the two-death split explains.

Two limits of this ledger, stated so it is not read as more than it is. It covers the pre-registered era only: exp24–exp26 carry frozen stances and falsification clauses but no prediction numbers, so their verdicts are reported in prose where they belong rather than counted here (exp24 above, exp25 in Appendix I, exp26 in Appendix I.10), and the experiments before exp19 predate the practice entirely. And a prediction count is not an error rate — these predictions are not independent, several are deliberately paired as a claim and its control, and the denominator was chosen by us. What the table supports is narrower and, we think, worth having: that the registered predictions in this programme were capable of failing, and did.

I.13 How Much of the 74% Survives a Real Model

The from-scratch calculation uses the probability of a uniform factored DFA output, whereas the model generates regex strings. Predictions P64–P67 were frozen before generation. This section retains their reported verdicts while separating observed probe hits from claims about eventual training success.

Design. We drew 200 instances uniformly without replacement from the 20,901-entry corpus frame and probed Qwen2.5-0.5B-Instruct with 2,048 samples per target at $T = 1.0$ (inference only, approximately 25 GPU-minutes). Sixty-eight target regexes could not be built by the parser and are reported as unbuildable rather than model failures. The median structural k is 96 in that excluded group and 154 in the buildable group. The from-scratch rule classifies 73.0% of the full draw and 76.5% of the 132 buildable targets as low-arrival under its specified policy.

Observed probe hits. At least one correct output appears for 48/132 buildable targets (36.4%), or 48/200 of the complete draw (24.0%). These proportions measure positive-hit probes, not successful fine-tuning or confidence-certified high-arrival targets. The registered P64 scorer reports PASS, with its reported 95% Clopper–Pearson lower bound of 0.294 applying to the positive-hit proportion as scored. A single hit gives $\widehat{H} = 9.375$ at $B = 19{,}200$, but its one-sided 95% lower bound is about 0.481. Zero hits give an upper bound about 28.1. Neither observation certifies a binary learning category.

Association and representation mismatch. In the rule's high- and low-arrival strata the positive-hit counts are 25/30 and 22/101 (one-sided Fisher $p = 1.4 \times 10^{-9}$; registered P65 PASS); the remaining buildable target lies between those strata. For the 22 low-arrival targets with probe hits, the reported discrepancy ranges from 4 to 994 orders of magnitude, with median 79 and eight exceeding 100. For

example, ([A-Za-z]){6,} has a 610-component transition representation and reported from-scratch $H = 10^{-992}$, while a regex model generates a compact description. Five rule-high targets yield zero probe hits (P66 FAIL), all involving an intersection operator. The median parse rate among zero-hit targets is 0.222, so absence of a correct sample is not equivalent to an absence of parsed samples (P67 PASS). This does not exclude format-related effects on correctness probability. The contrast changes policy and representation simultaneously and is not an isolated causal estimate of pretraining.

One-state stratum. Nineteen buildable targets have a one-state minimal DFA and all 19 yield positive probe hits; the rule classifies them all as high-arrival. After removing them, 29/113 targets yield positive hits (25.7%), while the from-scratch rule labels 89.4% low-arrival. The full and stratified values characterize the sample and probe; they are not bounds on the prevalence of eventual training failures. A repeated probe under the same seed, prompted by a parse-counter correction, gives identical hit counts on all 132 buildable targets.

Compounding rate ϱ (exp24), the artifact behind §3.3. Over 122 binary GRPO runs pooled from the position-constrained grid, the regex grid, the Qwen3-8B cell and the transition band, 78 escape and 44 are trapped. Two contingencies are reported because they answer different questions. *Censoring* — whether a growth rate can be fitted at all, i.e. whether the first hit repeats for at least three productive steps — is 75:3 estimable-to-censored among escaping runs against 14:30 among trapped ones (odds ratio 53.6, Fisher $p = 1.3\times10^{-14}$). That trapped arm is not clean by this paper's own taxonomy: 17 of the 44 never arrived, so they are censored by definition rather than by slow compounding. Restricted to the 27 that did arrive the table is 75:3 against 14:13, Fisher $p = 5.1\times10^{-7}$—the separation survives, and the printed 10^{-14} is the one inflated by the merge. We report both and use the arrived-only comparison to assess post-arrival censoring; on the estimable runs the medians are $\varrho = 0.106$ against -1.7×10^{-16}, rank-sum $p = 1.1\times10^{-8}$ ($n = 75$ against 14). *Compounding*, an indicator $\varrho > 0.02$, is $68/78$ against $0/44$ (Fisher $p = 7.3\times10^{-24}$); that threshold was fixed after we saw the censoring rate and is reported as a post-hoc test choice, as the measurement script itself records. The pre-registration fixed only that censored runs be reported rather than dropped.

I.14 An Unselected Draw

The preceding branch comparisons use targeted instance-selection rules. We therefore evaluate the learning-rate intervention on a uniform draw from the buildable corpus. Predictions P68–P73 and the scorer were frozen before the draw. Before any training job was created, the scorer was checked on synthetic grids representing the null cases, including a thin grid required to return UNTESTABLE rather than FAIL.

Frame, draw and streams. The frame is the 132 buildable instances of the uniform 200-instance corpus sample of Appendix I.13. The 68 excluded have targets our parser cannot build (\b, ∼); they skew *smaller* than the buildable set (median k 96 against 154), so the exclusion does not inflate the alive side. From those 132 we drew 30 uniformly without replacement and wrote them to disk before the job list was generated; no probe count, criterion call or training outcome was consulted. The seed handed to the runner is a hash of (instance, seed index), so streams are independent across instances, while the two rate arms of one (instance, seed index) deliberately share theirs. At $\beta = 0$, with a single prompt and no auxiliary loss, an all-identical reward group executes `continue` before `AdamW.step()`. An all-failure group therefore leaves the parameters unchanged, including through weight decay. The first-signal audit records 41 pairs arriving in both arms, 49 arriving in neither, no one-arm arrivals, and no first-arrival episode mismatches. These are checker-confirmed arrivals under the verification convention described below. Runner, reward and decoding are those of §4.2, unmodified: 30×2 rates $\times$ 3 seeds $\times E = 300$ at $G = 64$, 180 runs.

The two deaths, and the lever.

	never-arrived	trapped	escaped	escape given arrival
$\eta = 10^{-6}$	14	13	3	$9/41$ pairs
$\eta = 3\times10^{-6}$	14	4	12	$27/41$ pairs

Clopper–Pearson 95% intervals on the arrival-failure rate are $[0.28, 0.66]$ in both arms; the trapped fraction *of all thirty instances* — not escape given arrival, which is the last column — is $[0.25, 0.63]$ at 10^{-6} against $[0.04, 0.31]$ at $3{\times}10^{-6}$ (P73, reported and not tested). The rate moves escape and does not move arrival: the arrival classification is identical in both arms on all 30 instances, the first-signal episode is identical in all 41 arriving pairs (P71 PASS, 0 mismatches), and of the 18 discordant pairs all 18 favour the higher rate (P72 PASS, one-sided exact $p = 3.8{\times}10^{-6}$). The registered unit is the (instance, seed) pair, and we print the registered test; but those 18 pairs come from only nine instances, so that p is pseudo-replicated. With the instance as the unit all nine favour the higher rate, one-sided exact $p = 0.002$, and that is the figure the claim should be read at. This is the paper's central mechanism on instances nobody chose.

Verification convention and sensitivity. Across 180 runs, 12 verification timeouts were recorded in 10 runs (C007: 2; C034: 2; C060: 2; C081: 1; C177: 5). A timed-out check is treated as a failure and cached for subsequent calls; it is not a completed nonequivalence judgment. Training and the final 64-sample evaluation share one checker, and only a cumulative counter is saved after evaluation. The saved records therefore do not identify the stage of each event or the influence of cache reuse on final scores. All 41 recorded first-arrival pairs remain matched. In a post-hoc sensitivity analysis, we excluded all seeds and both rate arms for C060, C081, and C177, the three instances containing pairs with unequal timeout counts between arms. Among the remaining 27 instances, all six with discordant success counts favor the higher rate (one-sided exact sign test, $p = 0.015625$). This supports directional consistency under that exclusion; it does not exclude all five instances with recorded timeouts or establish the absence of verification effects.

Two registered failures. *P68 FAIL.* We registered arrival failure as the majority mode at $\geq 18/30$ and it is $14/30$: a plurality, near-even with the 13 trapped. The two-death split is not weakened by this—both branches being populated is what makes a two-way diagnostic worth having—but the emphasis is: the theory of this paper is spent on the arrival branch, which is under half of an unselected draw at the base rate. It is 14 of the 18 remaining failures once the escape lever is applied, which is a different quantity and is reported here as post-hoc.

P70 FAIL. The registered prediction was that no zero-hit-probe instance would succeed under either rate. C100, C118, and C177 do: each arrives at 10^{-6} and satisfies the final criterion at 3×10^{-6} despite zero hits in 2,048 initial-policy samples. At this probe depth the 95% upper bound is $H \leq 28.1$, so the observation leaves substantial arrival probability unresolved. The draw contains 11 positive-hit and 19 zero-hit probes; Algorithm 1 separately reports observed training arrival and final outcomes rather than reproducing the earlier prospective labels. The from-scratch calculations for these targets are $H = 1.4 \times 10^{-89}$, 3.8×10^{-33}, and a value numerically reported as zero for C177. For a finite uniform output space with at least one accepted output, the mathematical probability is positive; the reported zero should not be read as impossibility. As in Appendix I.13, the DFA calculation and regex probe concern different policies and output representations.

The probe still predicts, thinly. P69 PASS: escape at the base rate is $3/11$ among positive-hit-probe instances against $0/19$ among zero-hit-probe ones, one-sided Fisher $p = 0.041$. The margin is thin because only three instances escape at all in that arm, and it does not survive the stratification this paper performs one subsection earlier: three of the thirty have a one-state minimal DFA, all three have positive probe hits, and excluding them — as Appendix I.13 excludes their nineteen counterparts from the corpus figure — leaves $2/8$ against $0/19$ at $p = 0.080$. The registration did not name the stratum, so the PASS stands as registered and the stratified failure is reported beside it; this draw does not establish that the probe orders unselected instances. An unregistered sensitivity—scoring escape under either rate— gives $9/11$ against $3/19$ at $p = 0.0006$; it is reported as post-hoc and the registered verdict is the primary one.

I.15 The Mixed Batch and the KL Term

Every other run in this paper trains one instance per batch at $\beta = 0$, the regime in which the pre-arrival freeze of Remark 2 is exact. Deployed RLVR trains many prompts per batch with a KL coefficient, and §5.2 and Appendix F predicted that the escape-only lever should weaken there. This experiment is the smallest departure from the protocol that can test that prediction. Predictions P74–P79 and the scorer were frozen before any job was created

(notes/exp44-preregistration.md), and the scorer was dry-run on synthetic null, alternative and thin grids, the last of which must return UNTESTABLE rather than FAIL.

Design. Instances are fixed by rule from the unselected draw of Appendix I.14, with no outcome of this experiment consulted. *Trapped partners* are the six instances that arrive in all three seeds at $\eta = 10^{-6}$ with a first-signal step above 0 and never escape at that rate (C060, C071, C101, C118, C140, C151; two further qualifiers arrive at step 0, leaving arrival no room to move, and sit in the one-state stratum—excluded, and said so). *Alive partners* are two instances that escape $3/3$ at both rates and arrive by step 2 (C066, C185), assigned alternately in sorted order (C060, C101, C140 with C066; C071, C118, C151 with C185). An alive partner is reused across three pairs; we registered it as context, the results below show it is a nuisance factor with two levels, and the pre-registered follow-up at the end of this subsection exchanges the partners. *Never-arrived* instances, for P74 only, are the first three in sorted order of the fourteen (C002, C006, C007). Cells: mixed at $\beta = 0$ and at $\beta = 0.02$ (6 pairs $\times$ 2 rates $\times$ 3 seeds $= 36$ each); single at $\beta = 0.02$ on the trapped six (36) and on the never-arrived three (18); the single, $\beta = 0$ baseline is the draw itself. $\eta \in \{10^{-6}, 3{\times}10^{-6}\}$, $E = 300$, $G = 64$ per prompt, $T = 1$, Qwen2.5-0.5B-Instruct, a fresh checkpoint per run: 126 runs. The runner imports the reward, sampling and log-probability code of the single-instance runner unchanged. Per step each prompt gets its own group of G completions and its own group-relative advantage (zero if its rewards are all identical), and one optimizer step is taken over the union, with loss the mean over prompts of $-(A \cdot \log \pi_\theta)$ averaged over the group plus $\beta\,(\log \pi_\theta - \log \pi_{\mathrm{ref}})$ averaged over the group. At $\beta = 0$ a step on which every group is all-identical is skipped, which is what the single-instance runner does; at $\beta > 0$ the step always executes, so the KL gradient is applied on all-failure groups—which is what Remark 2 asserts. The reference policy is a frozen copy of the initial checkpoint; the estimator is the one of the KL ablation in Appendix I.11. Seeds are SHA-256 of (experiment, cell, pair, seed index): independent across pairs and cells, and shared across the two rate arms of one (cell, pair, seed) so that first-signal identity can be audited. Seven reward-evaluator timeouts occurred in 3,801,600 reward calls, in four runs (a timeout scores the sample 0, so it can only hide a hit): C060+C066 at $3{\times}10^{-6}$, $\beta = 0$, one timeout, first signal 13 in both arms; C151+C185 at 10^{-6}, $\beta = 0$, one, first signal 284 against 166 at the higher rate, so a hidden hit could only have moved a pair that already differs; C060+C066 at 10^{-6}, $\beta = 0.02$, one, first signal 6; and C071+C185 at $3{\times}10^{-6}$, $\beta = 0.02$, four timeouts in a run that never arrived and whose pair enters P78 through its 10^{-6} arm regardless. No registered verdict can change.

cell	η	arrives	escapes	partner escapes	steps run
single, $\beta = 0$ (exp 42)	10^{-6}	$18/18$	$0/18$	—	—
	$3{\times}10^{-6}$	$18/18$	$8/18$	—	—
mixed, $\beta = 0$	10^{-6}	$18/18$	$2/18$	$18/18$	217–290
	$3{\times}10^{-6}$	$14/18$	$9/18$	$18/18$	54–172
mixed, $\beta = 0.02$	10^{-6}	$18/18$	$2/18$	$18/18$	300
	$3{\times}10^{-6}$	$9/18$	$5/18$	$12/18$	300
single, $\beta = 0.02$	10^{-6}	$14/18$	$0/18$	—	300
	$3{\times}10^{-6}$	$6/18$	$1/18$	—	300
single, $\beta = 0.02$, never-arrived	10^{-6}	$0/9$	$0/9$	—	300
	$3{\times}10^{-6}$	$0/9$	$0/9$	—	300

"Arrives" and "escapes" are the trapped prompt's counts over 18 (pair, seed) units (9 for the never-arrived cell); "steps run" is the number of optimizer steps not skipped, out of 300. Rows are produced by experiments/exp44_mixed_batch/emit_table.py over the run files.

Registered verdicts. **P74 PASS**: the KL term produces no arrival on the never-arrived instances ($0/18$), though it moves the policy—their parse rate ends at 1.00 in eight of nine runs at $3{\times}10^{-6}$ and at 0.00 in the ninth. **P75 PASS**, **P76 PASS**: first-signal identity across η breaks under the KL term alone (4 of 6 arriving pairs differ) and in the mixed batch at $\beta = 0$ (12 of 14), where the single-instance runner gives $41/41$ identical (Appendix I.14); the implementation does what the paper says. The direction is null in both (P75b: 2 later, 2 earlier; P76b: 5 later, 7 earlier, two-sided $p = 0.77$). **P76c PASS**: sharing does not kill arrival at 10^{-6} ($18/18$). **P77 PASS**: among pair-seeds arriving at 10^{-6} in the mixed batch at $\beta = 0$, escape is $2/18$ at 10^{-6} and $9/18$ at $3{\times}10^{-6}$; 9 discordant, 8 favour the higher rate, one-sided exact $p = 0.0195$. Pair-seeds within an instance are not independent: at the instance level three instances move up (C060 $0/3 \to 3/3$, C101 $0/3 \to 1/3$, C140 $0/3 \to 3/3$), none moves down, and three are tied (C071 $2/3 \to 2/3$, C118 and C151 $0/3 \to 0/3$), a sign test of

$3/3$ at $p = 0.125$. **P78 FAIL**: the same test at $\beta = 0.02$ gives 7 discordant, 5 favouring the higher rate, $p = 0.23$. **P79** (reported): C071 escapes $2/3$ at 10^{-6} in both mixed cells, where alone at that rate it escapes $0/3$; no other instance does.

Two observations we did not register. *Arrival at the higher rate, and the partner.* At $3{\times}10^{-6}$ and $\beta = 0$, four pair-seeds that arrive at 10^{-6} never arrive (C118 $3/3 \to 0/3$, C151 $3/3 \to 2/3$), all four with partner C185. Recomputed from the trajectories (`emit_table.py`): at $3{\times}10^{-6}$ the partner's group is first all-correct at step 20–35 (C185, median 24) or 31–71 (C066, median 47), against 146–184 and 236–284 at 10^{-6}; after that its group carries variance only sporadically, so the run executes 54–147 optimizer steps with C185 (median 59) and 55–172 with C066 (median 115), against 217–290 at 10^{-6}. Sampling continues on every step regardless, so a frozen policy at an unchanged p_0 would have arrived at the same rate in both arms; it did not. C118 draws 0 hits in 19,200 samples in each of its three runs at $3{\times}10^{-6}$ against 9, 1 and 5 at 10^{-6}, while its parse rate over steps 60–69 is 56–57 of 64 against 23–28: the partner's updates moved C118's distribution further at the higher rate and moved its p_0 down. That is drift through the partner, and its sign is instance-dependent—or partner-dependent, which this design cannot tell apart: the three trapped instances paired with C066 escape at $3{\times}10^{-6}$ in $3/9$ runs alone and $7/9$ shared, the three paired with C185 in $5/9$ alone and $2/9$ shared, so every instance that moves up in the registered test is one partner's and every tie is the other's. The lever's total is nearly unchanged by sharing ($8/18$ alone, $9/18$ shared); who benefits is not. Whether a trainer skips or executes a zero-advantage step is an implementation choice; at $\beta = 0$ the two differ only by the optimizer's momentum tail and weight decay. *The KL term as estimated here is not an anchor.* Its all-failure gradient is the score function of the sampled outputs, zero in expectation and of order $\beta/\sqrt{G}$ per batch; the runs show it moving the policy a long way in 300 steps. Alone at $3{\times}10^{-6}$ the trapped instances' parse rate settles within 3 of $64/64$ or of $0/64$ between step 30 and step 66 in all 18 runs and stays there; at 10^{-6} it declines, as a per-run mean, from 10–16 of 64 over the first ten steps to 0–6 over the last thirty in 15 of 18 runs, and rises to 62 in the other three (one instance). Arrival falls with it ($18/18 \to 14/18 \to 6/18$ alone; $18/18 \to 9/18$ in the mixed batch at the higher rate, where even the alive partners fall to $12/18$). The one escape in the single KL cell is a run whose parse rate went to $64/64$ on the correct regex. The estimator GRPO uses (Shao et al., 2024), $r - \log r - 1$ with $r = \pi_{\text{ref}}/\pi_\theta$, has derivative $1 - r$ in $\log \pi_\theta$, which is 0 at the reference and a restoring pull once the policy has moved; under it the pre-arrival freeze of a never-updated policy is exact at any β (up to weight decay), and a KL term only matters for arrival once some prompt in the batch has moved the parameters. exp46, below, is the cell that tests it.

Partner swap, and GRPO's own estimator (exp46, pre-registered). The preceding comparison confounds the focal instance with the partner and uses a KL estimator that does not anchor the policy at the reference. We therefore test partner exchange and the reference-anchoring GRPO estimator. Predictions P85–P90 and the scorer were frozen before any job (`notes/exp46-preregistration.md`; scorer dry-run on synthetic null, alternative and thin grids); 108 runs, seeds disjoint from exp44. Three cells: the six pairs with partners *exchanged* (C060, C101, C140 with C185; C071, C118, C151 with C066) at $\beta = 0$; exp44's pairs at $\beta = 0.02$ under GRPO's per-token estimator $r - \log r - 1$, $r = \pi_{\text{ref}}/\pi_\theta$ (the runner's `--kl k3`; the k_1 path is byte-identical to exp44); and the six trapped instances alone under the same estimator. Two reward-evaluator timeouts in 3,456,000 calls, both in mixed-cell runs that arrive at 10^{-6} regardless; a timeout can only hide a hit, so no verdict can change.

cell	η	arrives	escapes	partner escapes	steps run
mixed, $\beta = 0$, partners exchanged	10^{-6}	18/18	6/18	18/18	187–291
	$3{\times}10^{-6}$	18/18	15/18	18/18	90–156
mixed, $\beta = 0.02$, GRPO's k_3	10^{-6}	18/18	0/18	18/18	300
	$3{\times}10^{-6}$	16/18	9/18	18/18	300
single, $\beta = 0.02$, GRPO's k_3	10^{-6}	18/18	3/18	—	300
	$3{\times}10^{-6}$	18/18	10/18	—	300

Rows from `experiments/exp46_partner_swap_k3/emit_table.py`.

Arrival direction (unregistered). Over the 36 $\beta = 0$ pair-seeds of exp44 and the swap, the trapped prompt's first signal comes earlier at $3{\times}10^{-6}$ in 21 and later in 7 (two-sided sign $p = 0.0125$; 4 suppressed): exp44's pairing alone gives 7 against 5 (P76b, null) and the swap 14 against 2 ($p =$

0.004, median ratio 1.9×); at the instance level five are earlier-dominant and one later-dominant, not significant. **P85 PASS.** Over the twelve (instance, partner) units of exp44 and the swap, six move up and none down (one-sided sign $p = 0.016$); pooled over both assignments escape is $8/36$ at 10^{-6} and $24/36$ at 3×10^{-6}. **P85b** (reported): the sign is concordant across the two partners in only 2 of 6 instances, and *up* is 5 of 6 instances with C066 against 1 of 6 with C185 (two-sided Fisher $p = 0.08$)—because with C185 as partner C060 and C101 already escape $3/3$ at 10^{-6} (alone: $0/3$), C071 $2/3$, so the rate has nothing left to move, C118 never arrives at 3×10^{-6}, and C151 loses one seed of three. The partner sets the baseline the lever acts from; the lever's sign is never negative. **P86 FAIL.** Arrival suppression at 3×10^{-6} is $4/18$ with C185 against $0/18$ with C066 ($p = 0.052$), and the swap adds none: C185's new partners arrive $18/18$. The suppression is two instances (C118, C151) with one partner, not the partner. **P87 PASS.** Under GRPO's own estimator the mixed batch at $\beta = 0.02$ gives escape $0/18 \rightarrow 9/18$, nine of nine discordant pair-seeds favouring the higher rate ($p = 0.002$), arrival $18/18$ and $16/18$ (the two misses are C101 with C066), the partner escaping $18/18$; P78's failure was the k_1 estimator's. **P88 PASS.** Alone under the same estimator, the first-signal step is identical across η in all 18 arriving pairs. This is observed timing agreement, not an audit of exact parameter stasis. The runner executes optimizer steps in these $\beta = 0.02$ cells. Although the k_3 derivative vanishes at the reference policy, that fact does not eliminate other optimizer effects, including nonzero weight decay. Effective optimizer settings must also satisfy Remark 2 before exact stasis can be inferred. **P89 PASS.** Arrival is $18/18$ at both rates (the k_1 term gave $14/18$ and $6/18$); the final parse rate stays near its starting value in runs that do not escape (median 0.22 at 10^{-6}) rather than collapsing to 0 or 1; escape is $3/18$ and $10/18$ against $0/18$ and $8/18$ for the same six at $\beta = 0$. These measurements describe arrival and final outcomes under the configured optimizer; they do not extend exact stasis beyond the conditions in Remark 2. In the mixed setting the observed final-success effect remains present with this estimator.

Units and evaluation conventions. For the eight- and thirty-prompt arms, each seed trains one shared policy; prompt outcomes within a seed are coupled. The 18 or 39 focal outcomes are therefore not 18 or 39 independent training replicates. The registered instance- and pair-seed tests below are retained as part of the experimental record; they do not establish generalization across independent shared training runs. We base the main-text interpretation on descriptive final accuracy, per-seed counts, and the tested settings. Here "escape" counts use the final post-training criterion unless explicitly identified as the first training-group threshold crossing. The longer-budget arm continues the same three policy trajectories, so its arrival and cumulative event counts are nested by construction.

Width: eight and thirty prompts per step (exp47, pre-registered). Rounds five to seven all named the two-prompt batch as the binding constraint. Predictions P91–P96 and the scorer were frozen before any job (`notes/exp47-preregistration.md`); 18 runs, seeds disjoint from exp44 and exp46, all at $\beta = 0.02$ under GRPO's own estimator. Three cells: the six trapped instances and both alive partners in one eight-prompt batch with $G = 16$ per prompt (samples per step fixed at exp44's 128) and with $G = 64$; and the whole exp42 draw—all 30 uniformly drawn instances, 14 never-arrived, 13 trapped, 3 escaped at 10^{-6} alone—in one thirty-prompt batch with $G = 16$. Each run trains every prompt of its batch, so one run yields one outcome per prompt. The instance-level sign test registered as primary can pass with six instances only at 5 or 6 up and none down; we registered it anyway, as the paper's own unit, and print the pair-seed count beside it.

cell	η	arrives	escapes	partners escape	steps run
$P = 8, G = 16$: the six trapped	10^{-6}	$16/18$	$0/18$	$0/6$	300
	3×10^{-6}	$15/18$	$9/18$	$6/6$	300
$P = 8, G = 64$: the six trapped	10^{-6}	$17/18$	$0/18$	$6/6$	300
	3×10^{-6}	$18/18$	$11/18$	$6/6$	300
$P = 30, G = 16$: the draw's 13 trapped	10^{-6}	$23/39$	$0/39$	$0/9$	300
	3×10^{-6}	$32/39$	$0/39$	$9/9$	300
$P = 30, G = 16$: the draw's 14 never-arrived	10^{-6}	$0/42$	$0/42$	$0/9$	300
	3×10^{-6}	$0/42$	$0/42$	$9/9$	300

Rows from `experiments/exp47_width/emit_table.py`; counts are over (instance, seed); "partners" are the two alive instances at $P = 8$ and the draw's three escaped-alone instances at $P = 30$.

P91 FAIL, P94 FAIL. At $P = 8$ none of the six instances moves down at either group size; with $G = 16$ three move up (C060, C071, C140, each $0/3 \to 3/3$) and with $G = 64$ four (C101 joins at $0/3 \to 3/3$, C060 at $0/3 \to 2/3$), for one-sided sign $p = 0.125$ and 0.0625—the same three or four instances as at $P = 2$, and the same shortfall: the registered test cannot reach 0.05 with two or three ties. The pair-seed counts are 9 against 0 and 11 against 0 ($p = 0.002$, 0.0005), pseudo-replicated. We print the registered verdicts as written. The frozen consequence table read a P91 FAIL as "the lever weakens with width". The observed directions are positive, but these tests neither establish weakening nor preservation of effect magnitude. Their limited units and shared-policy dependence constrain the conclusion. **P92 PASS.** At eight prompts arrival clears its floor: $16/18$ at 10^{-6} ($15/18$ at $3{\times}10^{-6}$; the $G = 64$ cell, not registered for this test, gives $17/18$ and $18/18$). Across widths, however, base-rate arrival on the same six instances falls $18/18 \to 16/18 \to 12/18$ (two against thirty prompts, one-sided Fisher $p = 0.0095$), and at thirty the higher rate restores it, $12/18 \to 18/18$; the direction of the first signal is null at eight (6 earlier, 6 later at $G = 16$; 10 against 3 at $G = 64$) and at thirty (8 against 8). **P93.** At $\beta = 0.02$ every step executes by construction ($300/300$). Every group was nonetheless all-identical on 70–172 of 300 steps at $G = 16$, 10^{-6} (18–25 at $3{\times}10^{-6}$; 0–16 at $G = 64$ and 2–15 at thirty prompts, both rates pooled): the $\beta = 0$ freeze would have occurred here too and is masked by the KL term, not removed by width. **P95 UNTESTABLE.** In the thirty-prompt batch no trapped instance escapes at either rate ($0/39$, $0/39$), so there is no discordance to test. The reason is visible on the instances that escape alone (unregistered): at 10^{-6} the three of them escape $0/9$ in the shared batch (final joint accuracy median 0.03) and at $3{\times}10^{-6}$ $9/9$; with thirty prompts sharing one update, no focal prompt satisfies the final criterion after 300 steps at the base rate, and tripling the rate restores escape for those with headroom. The trapped prompt's own budget is the same at eight and at thirty (300 steps, $G = 16$); what differs is realized signal—its group carries variance on a median of 47 steps at eight against 2 at thirty, and its total hits over a run fall from a median 686 to 2 at $3{\times}10^{-6}$—so the $1/30$ share of each update, the smaller group and the batch-mates' drift are not separated by this design, and we do not attribute the null to any one of them. The next block asks whether additional rate or budget changes recover final performance in this setting and finds recovery on a subset. Arrival on the trapped rises with the rate, $23/39 \to 32/39$ (unregistered direction; among pairs arriving at both rates 8 earlier, 8 later), and their final parse rate rises from a median 0.27 to 0.61. **P96** (reported): the fourteen never-arrived instances arrive in $0/42$ at either rate and escape in none; their parse rate rises ($0.25 \to 0.39$) without a hit—drift, not arrival.

Recovering final performance at thirty prompts (exp48, pre-registered). The thirty-prompt null does not determine whether a larger rate or longer budget can recover final performance: with the loss averaged over prompts, a trapped prompt at $P = 30$ carries $1/15$ of the gradient weight it carried at $P = 2$, its group is a quarter the size, and its batch-mates drift the shared parameters. Predictions P97–P99 and the scorer were frozen before any job (`notes/exp48-preregistration.md`); 6 runs on exp47's three thirty-prompt seeds. Two arms against exp47's $3{\times}10^{-6}$, 300-step arm (the base): $\eta = 9{\times}10^{-6}$ at 300 steps and $\eta = 3{\times}10^{-6}$ at 900 steps. Both triple ηE ($2.7{\times}10^{-3}$ against the base's $9{\times}10^{-4}$) and neither varies P or G, so the arms test recoverability of final performance in this setting; they do not separate the three batch changes above, and the registered rationale for the rate arm—that a second tripling returns the trapped prompt's share of each update, η/P, to roughly the two-prompt base arm's ($3{\times}10^{-7}$ against $5{\times}10^{-7}$)—assumes the loss's $1/P$ sets the step, which under AdamW with gradient clipping it does not. Sampling on this hardware reproduced the base runs exactly—all 90 prompt trajectories of the 900-step arm coincide with the base arm over the first 300 steps—so the budget arm is the base arm continued, not a second draw. The registered unit is the instance (the draw's 13 trapped at 10^{-6} alone): escape count over the three seeds in the new arm against the base, one-sided sign test over discordant instances, floor 3. Because the base arm escapes $0/39$, no instance can move down and the test reduces to a count of instances that escape at all ($p = 2^{-7}$ at seven); and the thirteen instances of one run share one policy trajectory, so the run (seed) is printed as a unit beside it, three per arm.

cell	η	arrives	escapes	partners escape	steps run
$P = 30, G = 16$: the draw's 13 trapped	$9{\times}10^{-6}$	$29/39$	$18/39$	$9/9$	300
	$3{\times}10^{-6}$	$36/39$	$13/39$	$9/9$	900
$P = 30, G = 16$: the draw's 14 never-arrived	$9{\times}10^{-6}$	$0/42$	$0/42$	$9/9$	300
	$3{\times}10^{-6}$	$1/42$	$0/42$	$9/9$	900

Rows from `experiments/exp48_width_decided/emit_table.py`; the base arm is the $3{\times}10^{-6}$, 300-step row of the table above ($32/39$ arrive, $0/39$ escape, partners $9/9$).

P97 PASS. A second tripling returns escape at thirty prompts within 300 steps: $0/39 \rightarrow 18/39$, seven of thirteen instances (one-sided sign $p = 0.008$; C017, C071, C081, C100, C140 each $0/3 \rightarrow 3/3$, C060 $0/3 \rightarrow 2/3$, C118 $0/3 \rightarrow 1/3$)—5, 6 and 7 of thirteen in the three seeds against 0 in every base seed, five instances escaping in every seed. The continuous outcome agrees: the 39 trapped runs' final joint accuracy rises from a mean 0.156 to 0.470 (paired Wilcoxon $p = 6.5{\times}10^{-5}$), and the verdict holds at a 0.8 or 0.7 bar (8 and 6 up, none down). Every escaping run's group reaches the bar by step 20–177, and escape is close to bimodal here: the 18 runs that escape carry variance on 77–244 of 300 steps, the 21 that do not on 0–119, median 1 (C111 ends at 0.81). The arm has a cost: trapped arrival falls $32/39 \rightarrow 29/39$ over four instances (next paragraph), of which C101 is the one that loses arrival outright: $3/3$ at $3{\times}10^{-6}$ against $0/3$ at $9{\times}10^{-6}$ (and $0/3$ at 10^{-6})—the mirror of C118's suppression at two prompts—and the trapped prompts' final parse rate falls, median $0.61 \rightarrow 0.42$. **P98 PASS as registered; final-evaluation recovery.** Continuing the same three shared policy trajectories to 900 steps increases final success: $0/39 \rightarrow 13/39$, six instances ($p = 0.016$; C071, C081, C100 each $0/3 \rightarrow 3/3$, C017 $2/3$, C060 and C140 $1/3$; 3, 6 and 4 per seed against 0), the same instances as at $9{\times}10^{-6}$ less C118. The registered PASS label records the specified scoring rule. We report the endpoint comparison descriptively because the 39 prompt–seed outcomes arise from three shared training trajectories, and the longer arm continues the same trajectories. These endpoint gains are distinct from additional training-threshold crossings. The final successes include both types of crossing history: eight of the thirteen reach the bar after step 300 (305–707), and five are the base runs that had touched the bar before step 300 and ended it at joint accuracy 0.72–0.84 (C081 in all three seeds, C071 in two), which 600 more steps carry to 0.94–0.98; excluding C071 and C081 leaves 4 up ($p = 0.0625$), and a 0.7 bar also gives 4 up ($p = 0.0625$). These post-hoc sensitivity comparisons do not meet the 0.05 significance threshold; the pre-registered primary comparison retains its PASS verdict. Across all 39 prompt–seed outcomes, mean final joint accuracy increases from 0.156 to 0.380 (paired Wilcoxon $p = 2.2{\times}10^{-4}$). Escape is not bimodal here: a non-escaping C017 run carries variance on 849 of 900 steps and ends at 0.84, two C140 runs at 0.77 and 0.86. Among escaping runs, first signal to the bar takes a median 51 steps (17–169) within the 300-step window at $9{\times}10^{-6}$ and 301 (153–634) within the 900-step window at $3{\times}10^{-6}$; the windows differ, so this is descriptive, not an estimate of the rate's effect on escape time. Trapped arrival reaches $36/39$ by step 900 ($32/39$ within the first 300, identical to the base). **P99** (reported): the three alive instances escape $9/9$ in both arms; the fourteen never-arrived escape in none, and one run of one (C073) arrives at step 543 with a single hit; the trapped prompts' productive steps have median 4 ($9{\times}10^{-6}$) and 7 (900 steps) against 2 in the base—for the budget arm, a longer window over an identical prefix makes cumulative counts non-decreasing but does not guarantee an increase; their change alone does not isolate a mediating effect. Reward timeouts: 1 in 437,760 and 4 in 1,301,760 evaluations. **Arrival cost of the rate arm, per instance.** Tripling again moves the arrival counts of four of the thirteen: C101 $3/3 \rightarrow 0/3$, C151 $3/3 \rightarrow 2/3$ and C177 $2/3 \rightarrow 1/3$ down, C018 $0/3 \rightarrow 2/3$ up, for $32/39 \rightarrow 29/39$ overall; the final parse rate of the trapped prompts falls from a median 0.61 to 0.42. The budget arm costs no arrivals ($32/39 \rightarrow 36/39$ by step 900).

Deviations from the frozen consequence table. Its row for this outcome read the null as "dilution and budget together" and its Conclusion phrase as "the rate and the budget both had to rise". Neither is what the arms license: matched in ηE and holding P and G fixed, they cannot attribute the null to a diluted step; and each alone suffices, so "both" is wrong. The paper therefore reports recoverable final-performance shortfalls: a higher rate or longer continuation each recovers final success on a subset. These arms do not identify which batch change caused the shortfall or establish a general absence of width-related limitations.

The same result as a dose, not a bar. Table 10 prints the quantity the escape bar is computed from—each trapped run's final joint accuracy—for the four doses run at thirty prompts, so a reader can see what a thresholded count hides. Three things are visible there and not in the counts. The response is ordered in the rate: mean final accuracy over the 39 runs is 0.010 at 10^{-6}, 0.156 at $3{\times}10^{-6}$ and 0.470 at $9{\times}10^{-6}$, and all 13 instances are monotone in the rate by seed median. The bar is doing work at the base rate: six runs of two instances sit in $[0.72, 0.84]$ at $3{\times}10^{-6}$, five of them having reached the bar at some step, which is why the registered $0/39$ is a statement about the final evaluation and not about a policy that never moved. And the lever's reach at this width is partial: 7 of 13 instances put a run past the bar at $9{\times}10^{-6}$ and 5 of those in all three seeds, while *five*

instances (C018, C054, C101, C151, C177) stay under 0.10 in every run of every dose—for those the recipe of §6 has nothing to offer at thirty prompts.

Table 10: Final joint accuracy of every trapped run at $P = 30$ ($G = 16$, $\beta = 0.02$, GRPO's own estimator, the same three seeds in every arm; the 900-step arm is the base arm continued, so its first 300 steps are the base arm's). Three cells per arm, one per seed. **Bold** = at or above the 0.9 escape bar (success as scored); † = the run's group reached the bar at some step but its final evaluation did not. Rows from `experiments/exp48_width_decided/emit_dose.py`.

instance	$\eta = 10^{-6}$ 300 steps	$\eta = 3{\times}10^{-6}$ 300 steps	$\eta = 9{\times}10^{-6}$ 300 steps	$\eta = 3{\times}10^{-6}$ 900 steps
C017	0.00 0.08 0.03	0.31 0.48 0.45	**0.91 0.98 0.97**	0.84† **0.97 0.94**
C018	0.00 0.00 0.00	0.00 0.00 0.00	0.00 0.00 0.00	0.00 0.00 0.00
C054	0.00 0.00 0.00	0.00 0.00 0.00	0.00 0.00 0.00	0.00 0.00 0.00
C060	0.02 0.00 0.00	0.00 0.00 0.00	0.00 **1.00 1.00**	0.00 **0.95** 0.00
C071	0.00 0.02 0.02	0.72 0.84† 0.72†	**0.94 0.97 1.00**	**0.97 0.97 0.94**
C081	0.05 0.11 0.05	0.77† 0.83† 0.83†	**0.98 0.97 0.97**	**0.98 0.94 0.95**
C100	0.00 0.00 0.00	0.00 0.02 0.00	**0.98 0.95 1.00**	**0.97 0.91 0.91**
C101	0.00 0.00 0.00	0.00 0.00 0.00	0.00 0.00 0.00	0.00 0.00 0.00
C111	0.00 0.00 0.00	0.00 0.00 0.00	0.00 0.00 0.81†	0.00 0.00 0.00
C118	0.00 0.00 0.02	0.00 0.00 0.00	0.00 0.00 **0.98**	0.00 0.00 0.00
C140	0.00 0.00 0.00	0.00 0.11 0.02	**1.00 0.95 0.95**	0.77† **0.97** 0.86†
C151	0.00 0.00 0.00	0.00 0.00 0.00	0.00 0.00 0.00	0.00 0.00 0.00
C177	0.00 0.00 0.00	0.00 0.00 0.00	0.00 0.00 0.00	0.00 0.00 0.00
runs ≥ 0.9	$0/39$	$0/39$	$18/39$	$13/39$
runs ≥ 0.7	$0/39$	$6/39$	$19/39$	$16/39$
mean	0.010	0.156	0.470	0.380

What in a wider batch slows escape (descriptive, unregistered). The two arms above vary η and E, not the batch, so they cannot say which change of the thirty-prompt batch made 300 steps too few. The cells already run answer a narrower version of it. At one rate ($3{\times}10^{-6}$), one β and GRPO's own estimator they form a partial factorial in prompts per step P and generations per prompt G—$(2, 64)$, $(8, 64)$, $(8, 16)$, $(30, 16)$—and six trapped instances appear in all four. Taking time to the success bar as the outcome, so that a slowdown is visible where the escape count is 0: at fixed $G = 16$ the step $P = 8 \to 30$ multiplies the median bar-step over the runs that reach it, $126 \to 349$, and cuts the fraction reaching it by step 300 from $9/18$ to $2/18$; at fixed $P = 8$ the group size $G = 64 \to 16$ multiplies it $62 \to 126$; and at fixed $G = 64$ the step $P = 2 \to 8$ costs little in the reported median ($63 \to 62$, and $9/18 \to 12/18$ reach the bar), while generations per step also quadruple. At approximately matched generations per step—128 and about 500 each occur twice—the observed cross-cell median ratios are $2.0\times$ from two to eight prompts and $5.6\times$ from eight to thirty. These ratios describe the tested batch configurations. Prompt count, group size, and batch composition are not fully separated. Every caveat of an unregistered cross-cell read applies: the cells have disjoint seed streams, the horizons differ (300 against 900 steps) so these are medians over the runs that reached the bar rather than survival medians—the Kaplan–Meier median is not identified at thirty prompts, where under half the runs reach it. These six-instance, eighteen-outcome comparisons do not estimate an isolated prompt-count effect (`experiments/exp48_width_decided/emit_scaling.py`).

What this does and does not cover. Two prompts is the smallest mixed batch and the most favourable one for the lever: the trapped prompt's own group is half of every update; eight and thirty are one step each toward a deployed batch. The pairs are fixed by rule from a draw that was uniform, so the pairs are not; the alive partners are two instances, exchanged once; GRPO's own estimator was run at one β; at thirty prompts $G = 64$ was not run, no rate above $9{\times}10^{-6}$ was run (the transformer cell of Appendix I.9 reversed escape at a high rate), and 900 steps is three times the budget, not a sweep; and all of it is at one model scale. These results show that the learning-rate effect on final success extends to the tested shared-prompt settings. Our arrival-invariance guarantee uses the sufficient pre-arrival stasis conditions in Remark 2, which updates from other prompts can violate.

I.16 The Lever Arm on the Regex Task

This experiment tests the learning-rate intervention on the NL → regex task, extending the comparisons on the factored testbed (Appendix I.10) and the position-constrained LLM task (Appendix I.11). Predictions P58–P63 were frozen before any training run. The scorer was written and checked on synthetic null grids before the training data were collected. The predictions and scorer were revised before launch to address a non-falsifiable criterion, a ceiling effect, and a scorer error.

Instrument and instances. The runner is that of §4.2, unmodified: Qwen2.5-0.5B-Instruct, binary minimal-DFA equivalence reward, $G = 64$, $T = 1.0$, $\beta = 0$, one instance per run from a fresh checkpoint. Nine instances, fixed by rule: two *trapped* (arrive, but do not clear the bar within $E = 75$ at the base rate), two that escape early, two that never arrive, and three from a transition band located by a 32,768-sample arrival probe. That depth is the point: at 2,048 samples a single hit already implies $\hat{H} = 9.4$, so the entire band around the $H \geq 3$ threshold collapses into "zero hits". Over the 24 candidates $\hat{H}$ spans 0 to 2,193, nine previously unselected instances fall in $[0.6, 7.0]$, and four still have zero observed hits after 32,768 draws, leaving their initial correctness uncertain rather than proving it is zero. Band instances run at $E = 600$, the rest at $E = 300$; six seeds per rate, three at 10^{-5}.

Pooled escape time. Kaplan–Meier over the 40 runs per rate (24 escape at 10^{-6}, 32 at $3{\times}10^{-6}$): median escape time 117 steps against 16; among the 24 pairs that escape at both rates the paired difference has median -49.5 steps (`experiments/exp41_regex_lever/score.py`).

The estimator, audited before it was used. Because the arms share a stream and no optimizer step executes while every reward in a group is zero, arrival *must* be identical across rates. That is a property of the estimator, not a finding, so it was audited first: three separate processes on a never-arriving instance, at 10^{-6}, at 10^{-5}, and a repeat, produce element-identical trajectories. In the grid, arrival matched in all 75 paired runs and so did every pre-arrival trajectory entry—27 of those comparisons on band instances whose pre-arrival windows run to hundreds of steps, and 12 on complete 300-step trajectories where no update ever executes. A caveat that belongs with it: the reward is not a deterministic function of the sample, because every equivalence check runs under a 2 s wall-clock alarm whose result is cached (one firing appears in the earlier logs); the scorer therefore reports whether a mismatching pair's timeout counts differ, so an instrument event is not read as a theory failure.

instance	role	E	η=1	η=3	ratio	η=10
A5 `.{4}`	escapes early	300	35	7	5.0×	1
A6 `only a number`	escapes early	300	51	8	6.4×	1
A4 `ends with dog`	trapped at 75	300	90	13	6.9×	2
A1 `has dog`	trapped at 75	300	151	21	7.2×	3
D8	band, $\hat{H}$=0.6	600	>budget	151	—	3
D6	band, $\hat{H}$=0.6	600	>budget	>budget	—	3
D4	band, $\hat{H}$=1.8	600	>budget	>budget	—	3
D1, D3	never arrive	300	—	—	—	—

Kaplan–Meier median escape time in steps from arrival, right-censored at the budget each run actually had; rates in units of 10^{-6}. Among the instances that escape at the base rate the ratio is flat across a 4.3× range of baseline, which is what a compounding escape predicts and an additive one does not. Two of the three band instances have mean arrival at step 378 (D4) and step 290 (D6), and then do *not* escape in the budget that remains at $\eta \leq 3{\times}10^{-6}$; D8 escapes at $3{\times}10^{-6}$ with a median of 151 steps, and all three escape in three steps at 10^{-5}.

The asymmetry. Tripling the rate moves arrival on 0 of 75 paired runs and lifts escape given arrival from 13/26 to 25/26 at $E = 75$ (paired sign test over the 26 complete pairs: 12 discordant, all 12 favouring the higher rate, one-sided exact $p = 2.4{\times}10^{-4}$). The escape-time separation is a stratified paired log-rank $z = -5.66$, $p = 1.5{\times}10^{-8}$, and is separately significant in each stratum (PRIMARY $p = 5.3{\times}10^{-4}$, CTRL_ESC $5.3{\times}10^{-4}$, BAND $4.7{\times}10^{-3}$); the pooled test is stratified because the arms share a random stream, so the unit of inference is the (instance, seed) pair and

not the run. Quadrupling the budget moves arrival from $26/42$ to $40/42$—14 independent streams, all on band instances—and also lifts escape where escape has room. So on this task family too the rate is escape-only and the budget acts on both. Two instance sets fixed before any outcome was seen keep this from being a property of what we picked: the two chosen for never being trapped give the same escape-time ratio as the trapped ones ($5.0\times$, $6.4\times$ against $6.9\times$, $7.2\times$), and the three band instances, drawn by rule from the 32,768-sample probe with no training outcome consulted, go $0/18 \to 8/18$ over the two rates the comparison is scored on ($7/9$ at 10^{-5}, reported but not tested on).

Arrival follows the probe's law with no fitted parameter. Pooling 42 independent streams (the three rates share a stream, so runs would multiply-count) and comparing the observed first-signal step against $\mathrm{Geometric}(1-(1-\hat{P}_0)^G)$ with $\hat{P}_0$ from the probe, a randomised probability-integral-transform test gives KS $p = 0.75$ and 0.66 at two randomisation seeds, with a sign test over instances at $p = 0.38$. A plain Kolmogorov–Smirnov test against the geometric CDF is *not* usable here and was not used: on synthetic data drawn from that very null it rejects at $p = 10^{-54}$, because the distribution is discrete and heavily tied.

Two things this does not show. First, the 10^{-5} arm reaches the measurement floor—escape times of 1–3 steps against a one-step loop resolution. This resembles the rapid jumps in Appendix I.10, for which a gradual-compounding interpretation is uninformative. Large gains alone do not establish a violation of Proposition 4's uniform multiplicative bound. The scored rate comparison uses 1 versus 3×10^{-6}; the 10^{-5} arm is reported separately and its short escape times are limited by the loop resolution. Second, the pre-registered compliance prediction returned a null in the opposite direction: final parse rate *rises* from 0.599 to 0.994 at $10\times$. The design cannot resolve that cleanly—a run either escapes, after which all-correct groups are skipped and parse rate freezes at 1.0 in both arms, or never updates and is identical in both arms by construction, leaving 14 of 27 pairs exact ties. We report it as a design limit and not as evidence that compliance is safe at higher rates.

I.17 Evidence for the Main Claims

Table 11 connects each central interpretation to its identifying comparison. It separates observations that establish the interpretation from controls that delimit its scope. All experimental evidence comes from the runs reported elsewhere in this manuscript. The first-update normalization comparison is an analytical addition, not a new experiment.

Table 11: Claim, supporting comparison, and remaining scope.

Interpretation	Comparison and existing control	Remaining scope
Initial correctness does not determine adaptation	R24 fixes q_0, initial entropy, reward variance, and cardinality; escape times differ $49.5\times$. Base-3 ladders replicate the direction.	k and c change together. Initial reward matching does not match parameter gradients or effective step sizes.
The trend is not exclusive to Adam	Escape-time slopes are -1.57 under SGD and -1.88 under Adam. Crossed-grid controls sample the appropriate optimizer age.	Interior spreads are 1.0–$1.9\times$; a high-rate SGD arm jumps to convergence. No universal scaling law follows.
The isolated rate effect occurs after arrival	The uniform draw preserves first-arrival episodes for the same 41/90 runs; final successes increase 9/90 to 27/90.	One central 0.5B model; coverage is conditional on 132/200 buildable targets. Higher rates do not always help.
Selection is not the sole source of the rate effect	Uniform sampling replaces the selected trapped-instance rule; all nine instance-level discordances favor the higher rate.	Coverage is conditional on buildability. Verification conventions and sensitivity are detailed in Appendix I.14.
A success threshold can hide progress	At $P = 30$, both lower rates have 0/39 final successes, but mean final accuracy is 0.010 versus 0.156.	There are only three shared policy trajectories per arm. Arrival changes, so conditioning would select different cohorts.
Zero probe hits do not certify low arrival	The exact zero-hit upper confidence bound is $H_U = 28.1$ at the tested probe size and training budget; three zero-hit instances succeed.	This limits prospective classification. It does not invalidate retrospective measurement of first arrival.