Does This Gradient Spark Joy?

Ian Osband
Google DeepMind
iosband@google.com

Abstract

Policy gradient computes a backward pass for every sample, although the backward pass is expensive and most samples carry little learning value. The Delightful Policy Gradient (DG) scores learning value in the forward pass with *delight*, the product of advantage and surprisal (negative log-probability). We introduce the *Kondo gate*, which compares delight against a compute price and pays for a backward pass only when the sample is worth it; sweeping the price traces a quality–cost trade-off. On MNIST, keeping 3% of gradient terms learns faster than standard policy gradient even when backward passes are free, and nearly matches full DG. In bandits, zero-price gating preserves the expected gradient direction and removes its perpendicular noise, delight ranks samples more reliably than additive mixes of value and surprise, and a rare action with high reward variance marks where delight fails. On transformer token reversal, the gate's advantage grows with sequence length and vocabulary size. Because delight tolerates moderate noise, a cheap approximate forward pass could screen samples before expensive backpropagation, a training analogue of speculative decoding.

1 Introduction

Policy gradient methods compute a backward pass for every sample [21]. The backward pass is often substantially more expensive than the forward pass, yet most samples carry little learning value: a sample that confirms an action the policy already prefers, or punishes one it already avoids, contributes little to progress. The Delightful Policy Gradient (DG) provides a forward-pass signal of learning value [14]. DG weights each gradient term by a sigmoid of *delight*, the product of advantage and surprisal, so rare successes are emphasized and uninformative outcomes are suppressed. Delight is available from the forward pass before any gradient computation. DG improves over PG and PPO across staleness, actor bugs, reward corruption, and rare discovery [14, 12]. This suggests a sharper question: if the forward pass already identifies which samples are worth learning from, should we compute every backward pass at all?

We propose the *Kondo gate*[1]: keep the samples that spark joy, skip the rest. For each sample, the learner compares delight against a compute price λ and draws a Bernoulli gate. The probability of a backward pass increases with delight and decreases with price. Sweeping the price traces a quality–cost trade-off; in practice, we set λ adaptively to target a fraction ρ of backward passes. In distributed RL, where actors generate more trajectories than the learner can backpropagate through, Kondo can serve as admission control for the learner queue: cheap scoring decides which samples deserve learner time.

On MNIST at $\rho = 3\%$, the Kondo gate learns faster than standard PG per forward pass, before any backward savings are counted, and nearly matches full DG while using only 3% of backward passes; in backward-pass space it wins by two orders of magnitude (Section 3). In tabular bandits, we show that gating preserves useful gradient direction while eliminating perpendicular noise, and explain why

[1] Named for Marie Kondo's organizing principle: keep what sparks joy, discard what does not.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

delight is a better screening signal than additive combinations of value and surprise (Section 4). The same analysis exposes the main limitation: a gambling regime in which high reward variance creates false delight on rare suboptimal actions (Section 4.2). On transformer token reversal, the Kondo gate solves harder problems at equal backward compute, and its advantage grows with sequence length (Section 5).

2 The Kondo Gate

Standard policy gradient uses per-sample updates $g_t = U_t \nabla_\theta \log \pi_\theta(A_t \mid \mathcal{H}_t)$, computing a backward pass for every sample regardless of learning value [21]. The Delightful Policy Gradient (DG) scores each sample by delight $\chi_t = U_t \cdot \ell_t$, the product of advantage and surprisal $\ell_t = -\log \pi_\theta(A_t \mid \mathcal{H}_t)$ [14]. Delight is largest when a rare action succeeds, negative when a rare action fails, and small for actions the policy already expects.

DG uses delight to *weight* gradient terms, but it still computes every backward pass. The Kondo gate takes the next step: if delight already scores how much the learner can gain from a sample, then delight should also decide whether that sample deserves a backward pass at all.

2.1 Implementation

Suppose each backward pass carries a price $\lambda \geq 0$. For a single sample, we choose a gate probability $w \in [0, 1]$ by maximizing

$$\max_{w \in [0,1]} \underbrace{\chi\, w}_{\text{learning value}} - \underbrace{\lambda\, w}_{\text{compute cost}} + \underbrace{\eta\, H(w)}_{\text{uncertainty}}, \tag{1}$$

where $H(w)$ is binary entropy and $\eta > 0$ is a temperature (derivation in Appendix B). The unique maximizer is $w^* = \sigma\big((\chi - \lambda)/\eta\big)$: the probability of paying for a backward pass increases with delight and decreases with price. On a real computer the cost is all-or-nothing, so the Kondo gate samples $G_t \sim \mathrm{Ber}(w_t^*)$ and computes the backward pass only when $G_t = 1$. Two limits anchor intuition: at $\eta \to 0$ the gate is a hard threshold $\mathbb{I}\{\chi > \lambda\}$, keeping only the most informative samples; at $\eta \to \infty$ the gate is constant, recovering standard PG up to a uniform rescaling. In practice we set λ adaptively to target a gate rate ρ, the fraction of samples that receive backward passes.

Algorithm 1 Delight-based Kondo gating for policy gradient

Require: Batch $\mathcal{B}$, policy π_θ
Hyperparameters: gate rate $\rho \in (0, 1]$ **or** price λ and temperature $\eta > 0$

for $t \in \mathcal{B}$ **do**
  $\ell_t \leftarrow -\log \pi_\theta(A_t \mid \mathcal{H}_t)$ ▷ Surprisal (forward pass)
  $\chi_t \leftarrow U_t \cdot \ell_t$ ▷ Delight (forward pass)
if gate rate ρ given **then**
  $G_t \leftarrow \mathbb{I}\{\chi_t \text{ is among the top } \lceil \rho |\mathcal{B}| \rceil\}$ for all t ▷ Fixed-rate gate
else
  $G_t \sim \mathrm{Ber}\big(\sigma((\chi_t - \lambda)/\eta)\big)$ for all t ▷ Fixed-price gate
$\Delta\theta \leftarrow 0$
for $t \in \mathcal{B}$ **do**
  if $G_t = 0$ **then continue** ▷ Skip backward pass
  $\Delta\theta \leftarrow \Delta\theta + U_t \nabla_\theta \log \pi_\theta(A_t \mid \mathcal{H}_t)$ ▷ Backward pass
return $\Delta\theta$

In this method, which we call DG-K, delight decides which terms to compute but does not reweight them. Each retained term is the plain PG term, so relative to PG the gate only removes terms, and relative to DG it replaces continuous weighting with binary selection. In expectation over the gate, $\mathbb{E}[\Delta\theta \mid \mathcal{B}] = \sum_{t \in \mathcal{B}} w_t^*\, U_t \nabla_\theta \log \pi_\theta(A_t \mid \mathcal{H}_t)$ is a DG-style weighted gradient. Our fixed-rate experiments use the hard version, keeping the top-ρ fraction of each batch by delight. Both are deliberately biased relative to PG whenever the gate is selective. On a single device this requires sparse or packed backpropagation; in distributed training the same rule is simpler: use χ_t as the priority for the learner queue and admit only the samples that exceed the current price.

2.2 Why Delight, Not Simpler Priority Signals?

A backward pass should be spent on samples that are both useful and non-redundant. Advantage alone measures usefulness but does not identify how likely the sampled action was: two samples with equal advantage can have very different probabilities under the current policy, and the common one changes the policy little. Surprisal alone measures rarity but ignores value: it prioritizes novelty for its own sake, including surprising failures that the learner has already learned to avoid. Additive combinations $\alpha U + (1-\alpha)\ell$ interpolate between these two mistakes and require regime-dependent tuning of α.

Because it multiplies advantage and surprisal, delight is large only when a sample is both valuable and unexpected under the current policy. This makes it a natural screening signal for backward compute: keep the rare successes that teach the learner something new, skip the samples whose gradient is either redundant or actively unhelpful. Section 4 makes this precise in tabular bandits: it shows why delight is more reliable than additive alternatives and identifies the gambling regime in which it fails.

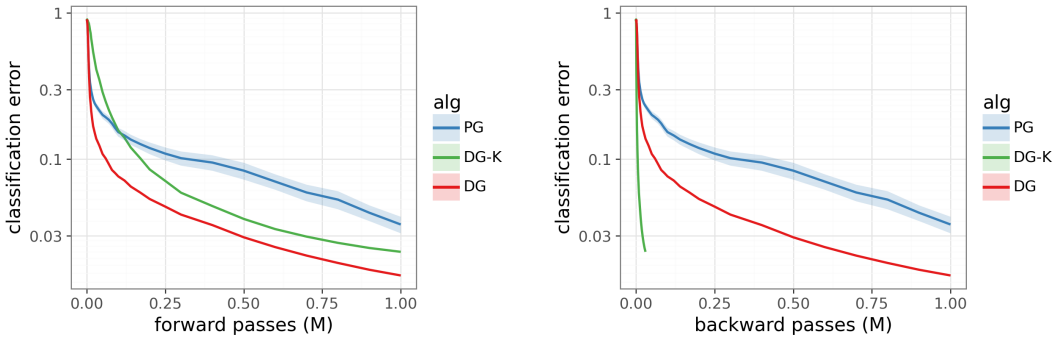
3 MNIST Diagnostic

We begin with MNIST [9], the simplest neural-network RL problem: a contextual bandit with ten actions, immediate reward, and a two-layer MLP. The setup follows Osband [14]: an MLP with two hidden layers of 100 units (about 90K parameters) trained with Adam on batches of 100 images; only the gradient gating differs. We parameterize the Kondo gate by a target *gate rate* $\rho \in (0, 1]$. On each batch, we retain the top ρ fraction of samples by delight. Setting $\rho = 1$ recovers standard PG; small ρ skips most backward passes.

3.1 Core Results

Figure 1 shows the main result at $\rho = 0.03$. In forward-pass space (a), the Kondo gate nearly matches full DG and both outperform PG, so gating preserves nearly all of the useful learning signal. In backward-pass space (b), the Kondo gate reaches the same error using two orders of magnitude fewer backward passes. The green curve ends earlier, reaching nearly the same quality at a small fraction of the backward cost.

Why should a deliberately biased update learn faster than the unbiased policy gradient, even per forward pass? Supervised learning makes the same choice: image classifiers are trained with cross-entropy, a surrogate that emphasizes hard examples, instead of the gradient of expected accuracy. Osband [14] shows that across contexts DG moves the expected update toward the cross-entropy direction; the Kondo gate keeps that reallocation while computing only 3% of the terms.



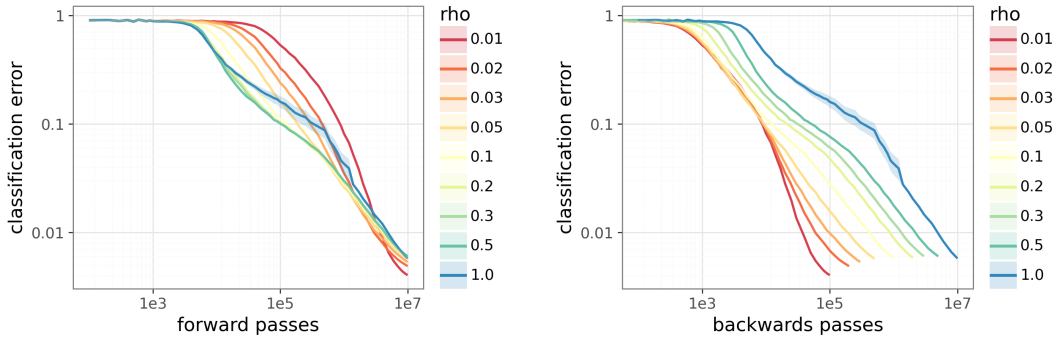
(a) Forward passes: DG-K $\approx$ DG $\gg$ PG.

(b) Backward passes: DG-K $\gg$ DG $\gg$ PG.

Figure 1: PG, DG, and Kondo gate (DG-K) at $\rho = 0.03$ on MNIST. (a) The Kondo gate matches DG despite computing 3% of backward passes. (b) It dominates by two orders of magnitude in backward-pass space. Averaged over 30 seeds; shading shows ± 1 standard error.

Figure 2 sweeps the gate rate ρ from 0.01 to 1.0, with learning rate tuned per ρ. In forward-pass space (a), all gate rates converge to nearly the same final error ($\sim 0.5\%$): over this range, aggressive

gating costs little in quality. In backward-pass space (b), $\rho = 0.01$ reaches low error with $\sim 100\times$ fewer backward passes than $\rho = 1.0$.



(a) Forward steps: all ρ converge to similar error.

(b) Backward steps: $100\times$ fewer at $\rho = 0.01$.

Figure 2: Gate rate sweep ($\rho \in \{0.01, \ldots, 1.0\}$), learning rate tuned per ρ. (a) All gate rates converge to $\sim 0.5\%$ error eventually. (b) In backward-step space, smaller ρ reaches any error with orders-of-magnitude fewer backward passes.

Figure 3 shows which samples spark joy late in training. The gate keeps digits that the policy classifies correctly but not yet with certainty, which we call the learning frontier. It skips digits that are already solved, along with mistakes whose negative advantage makes delight negative. The selection does not fade as training proceeds; at step $10{,}000$ kept samples still have systematically lower $\pi(y^*)$ than skipped ones (Appendix A.4).

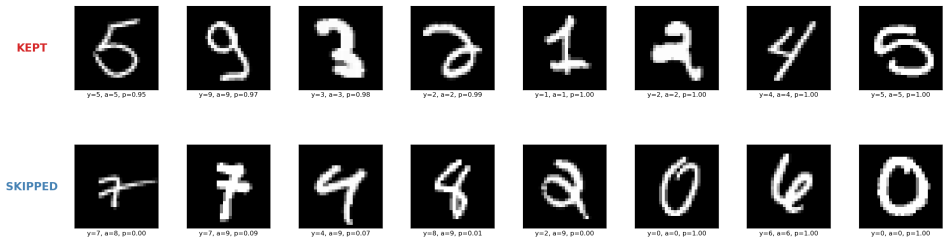


Figure 3: Images kept (top) and skipped (bottom) by the Kondo gate at step $10{,}000$ ($\rho = 0.03$), annotated with label y, action a, and $p = \pi(y^*)$. The gate keeps the learning frontier; it skips solved digits and confident mistakes.

3.2 Compute Efficiency and Approximate Delight

The practical value of skipping backward passes depends on their cost. Our experiments count selected gradient terms and do not measure wall-clock time. Figure 4 combines the forward and selected-term counts from the learning curves under an assumed backward/forward cost ratio, and reports the modeled compute (forward + backward $\times$ ratio) to reach 5% test error, normalized to PG. This ratio is an assumed input: a backward pass costs about twice a forward pass in FLOPs [7], and activation recomputation raises the practical ratio [11]. Even at ratio 0 (backward passes are free), DG-K improves over PG through better per-sample learning. As the assumed ratio grows, the modeled speedup increases: at a ratio of $4\times$, DG-K needs $6\times$ less modeled compute than PG to reach the same error. DG's speedup is constant ($\sim 2\times$) because it still computes every backward pass.

The gate's decision requires only a forward pass, which need not be computed at full precision. If delight can be approximated, whether via quantized inference, a distilled model, or cached values, the *effective* backward/forward cost ratio is much larger than $2\times$. A cheap screening pass followed by an expensive full pass mirrors speculative decoding [10], applied to training. Figure 5 tests this by injecting noise into the delight signal (a) and the forward-pass logits (b). DG tolerates roughly 50% relative delight noise and logit noise up to $\sigma_Z \approx 1$ before degrading; DG-K is more fragile in both cases. These results suggest that screening need not be exact to capture much of the compute savings.

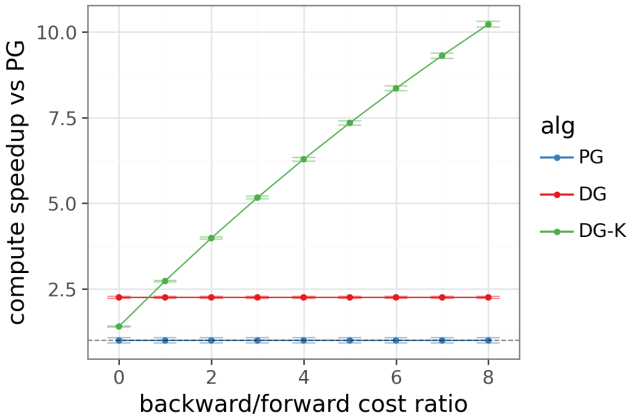
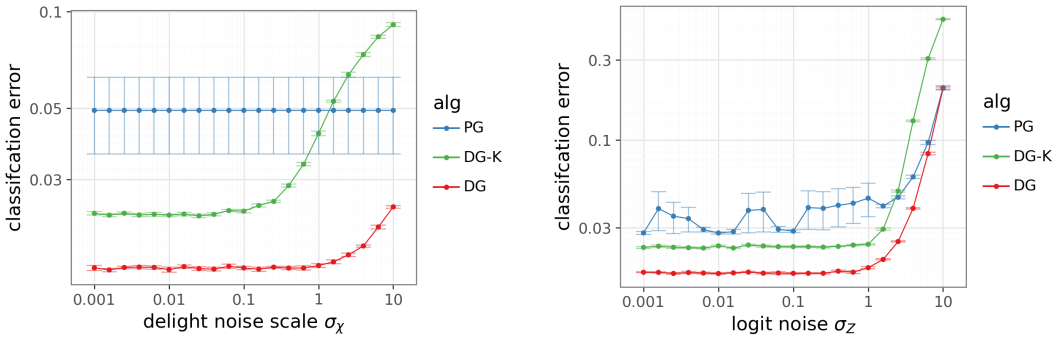


Figure 4: Modeled compute speedup vs PG to reach 5% test error on MNIST, as a function of an assumed backward/forward cost ratio. DG's advantage is constant ($\sim 2\times$, better learning). DG-K's advantage increases with backward cost (fewer backward passes). At a ratio of $4\times$, the Kondo gate needs $6\times$ less modeled compute than PG.



(a) Delight noise (relative): DG tolerates $\sim 50\%$.

(b) Logit noise: DG robust to $\sigma_Z \approx 1$.

Figure 5: Noise robustness on MNIST. (a) Delight noise scaled relative to $\mathrm{std}(\chi)$: DG tolerates $\sim 50\%$; DG-K degrades earlier. (b) Logit noise: DG is robust until $\sigma_Z \approx 1$; DG-K degrades faster. Both validate that approximate forward passes and approximate delight preserve the gate's value.

The MNIST results show that most backward passes can be removed with little loss in learning quality, and that the resulting speedup grows with backward-pass cost; tabular bandits (Section 4) explain why.

4 Tabular Analysis

We now answer the three questions raised by MNIST in a setting with exact gradients and no function-approximation error. First, what useful part of the gradient survives the gate? Second, why is delight the right priority signal rather than a simpler combination of value and surprise? Third, when should delight-based screening fail? Bandits make all three questions analytically tractable, while remaining close enough to MNIST that the predictions can be checked empirically.

4.1 Pareto Improvement and Priority Signal

The first question is what the gate keeps. In a single-context softmax bandit, correct-action gradients are parallel to ∇J and carry no perpendicular noise. Incorrect-action gradients contribute $\Theta(1)$ perpendicular component and only $\Theta(p)$ cosine with ∇J. With a baseline bounded away from zero, the batch cosine scales as $\Theta(p\sqrt{B})$ when $p^2 B \ll 1$: unless $B \gg 1/p^2$, the policy-gradient estimate

is nearly random. In this setting, zero-price gating is a Pareto improvement in gradient geometry: it preserves the useful signal, eliminates perpendicular noise, and reduces backward-pass cost.

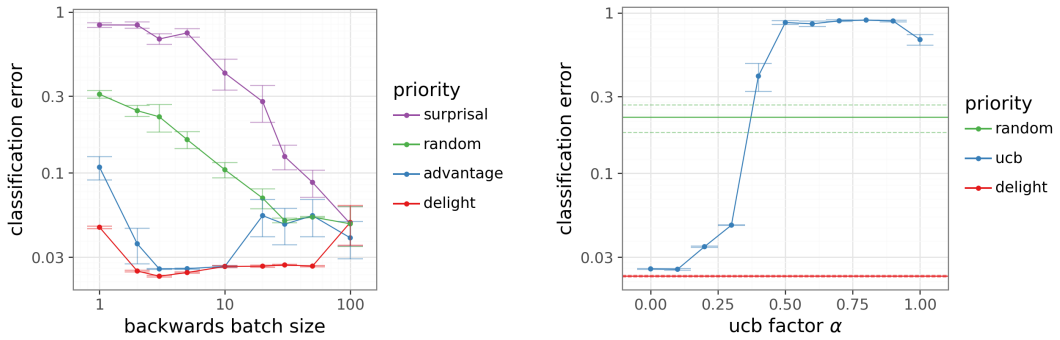
Proposition 1 (Kondo gate Pareto improvement)**.** *Under a K-armed bandit with softmax policy $\pi = \mathrm{softmax}(z)$, deterministic reward $R = \mathbb{I}\{A = y^*\}$, and correct-action probability $p = \pi(y^*)$, consider the zero-price hard gate that keeps samples with $\chi > 0$ and skips those with $\chi < 0$:*

1. *Direction preserved: $\mathbb{E}[g_{\mathrm{KG}}] \propto \nabla_z J$.*
2. *Perpendicular variance eliminated: $\mathrm{Var}_\perp(g_{\mathrm{KG}}) = 0$.*
3. *Compute reduced: pB backward passes instead of B.*
4. *Backward cost: for a baseline b bounded away from zero, PG needs $\Omega(1/p^2)$ backward passes for $\cos = \Theta(1)$; KG achieves $\cos = 1$ with probability $\geq 1 - \delta$ from $O(\log(1/\delta))$ backward passes, using about $\log(1/\delta)/p$ forward samples.*

This explains why aggressive gating can preserve learning quality. The next question is why to threshold delight instead of a simpler additive score built from advantage and surprisal. At $\lambda = 0$, gating on $\chi < 0$ is equivalent to gating on $U < 0$, so the product structure does not matter for sign alone. It matters once we must rank *among* positive-delight samples, as happens with $\lambda > 0$, multiple contexts, or approximate screening.

Proposition 2 (Delight is sign-consistent; additive mixes can mis-rank)**.** *Under Assumption 1, delight $\chi = U \cdot \ell$ satisfies $\chi(y^*) > 0 > \chi(a \neq y^*)$ for all p, K. The additive family $f_\alpha = (1-\alpha)U + \alpha\ell$, with surprisal weight $\alpha \in [0, 1)$, always ranks the correct action above incorrect ones when $p \leq 1/K$; when $p > 1/K$ it does so only if $\alpha < \alpha^*(p, K) = \frac{1}{1+L}$, where $L = \log\left(\frac{p(K-1)}{1-p}\right) > 0$. The admissible surprisal weight shrinks as policy quality or action-space size grows.*

Multiplying the advantage by a positive surprisal preserves its sign, whereas adding surprisal can flip it. Figure 6 checks this prediction on MNIST. Delight remains robust across backward batch sizes and across the full sweep of additive-mix coefficients, whereas surprisal-only and additive priorities degrade or collapse.



(a) Error vs. backward batch size by priority.

(b) Error vs. surprisal weight α; delight is flat.

Figure 6: Priority signal comparison on MNIST. (a) Delight is robust across backward batch sizes; surprisal-only fails. (b) The additive mix collapses for $\alpha > 0.3$; delight (product) is α-independent. Consistent with Proposition 2.

4.2 The Gambling Pathology

The final question is when delight-based screening should fail. The failure mode is a specific gambling regime in which a rare suboptimal action has such high reward variance that lucky draws masquerade as breakthroughs.

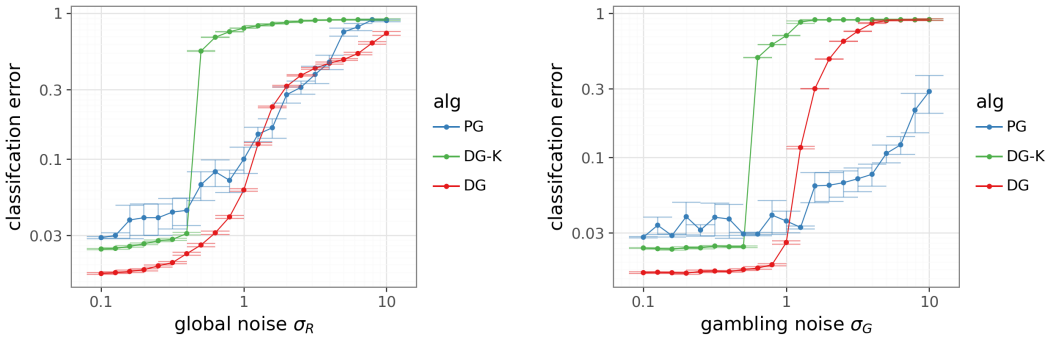
Consider a slot machine: arm 1 pays \$1 always; arm 2 pays \$0 with probability 0.99 and \$50 with probability 0.01. Gap $\Delta = 0.50$, noise $\sigma \approx 5$, ratio $\sigma/\Delta = 10$. When the slot machine hits, the learner observes $U > 0$, and because the policy rarely pulls arm 2, the high surprisal opens the gate wide.

Proposition 3 (Gambling pathology)**.** *Two-armed bandit: arm 1 (optimal) with deterministic reward μ^*, arm 2 (suboptimal) with Gaussian reward $R_2 \sim \mathcal{N}(\mu^* - \Delta, \sigma^2)$.*

1. *When $\sigma/\Delta \ll 1$:* $\Pr(U_2 > 0 \mid A{=}2) \leq \exp(-\Omega(\Delta^2/\sigma^2))$.
2. *When $\sigma/\Delta \gg 1$:* $\Pr(U_2 > 0 \mid A{=}2) = \Theta(1)$.
3. *Delight amplifies: $|\chi_2| = |U_2| \cdot \log(1/\varepsilon)$ grows as the policy avoids arm 2.*

No per-sample statistic computed from (R, π) can distinguish a genuine breakthrough from a lucky draw. The pathology is driven by *differential* σ_a/Δ_a; in our homoskedastic experiment, DG and PG degrade together. The same joint signal that makes delight valuable in normal learning becomes pathological in this regime, because a rare lucky draw on a suboptimal arm looks like a breakthrough.

We validate this on the MNIST bandit from Section 3. To inject differential noise, we designate action $a{=}0$ as the "gamble": whenever the agent predicts 0 (regardless of true label), its reward receives additive $\mathcal{N}(0, \sigma_G^2)$ noise. Figure 7 sweeps two noise regimes. Under homoskedastic noise σ_R, DG and PG degrade smoothly together (a). Under gambling noise σ_G on a single action, DG outperforms PG for $\sigma_G < 1$ but collapses sharply near $\sigma_G \approx 1$ while PG degrades gracefully (b), consistent with the regimes of Proposition 3.



(a) Homoskedastic σ_R: smooth joint degradation.

(b) Gambling σ_G: sharp DG collapse.

Figure 7: Gambling pathology on MNIST (Section 3). (a) Under homoskedastic noise, DG and PG degrade smoothly together. (b) Under differential noise on action $a{=}0$, DG collapses sharply near $\sigma_G \approx 1$ while PG degrades gracefully, consistent with the regimes of Proposition 3.

5 Token Reversal

The bandit analysis showed that the Kondo gate can preserve useful gradient signal while removing much of the backward computation. We now test whether the same mechanisms survive in sequence-model training, where compute efficiency matters most. Token reversal [14] asks a transformer to reverse a length-H sequence drawn from a vocabulary of size M. The task isolates autoregressive generation and token-level credit assignment, and lets us vary sequence length and vocabulary size independently. As either H or M grows, the task becomes harder and informative events become rarer, so selective backward computation should become increasingly valuable. The agent is a causal transformer with 3 layers, 4 heads, and embedding dimension 64 (about 100K parameters). Each gradient step processes a batch of 100 episodes (10 prompts $\times$ 10 responses); full experimental details are in Appendix D. Code is available at `https://github.com/google-deepmind/egg`.

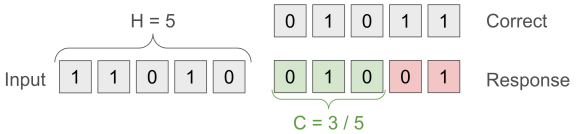


Figure 8: Token reversal ($M{=}2$, $H{=}5$): the agent must output the input in reverse; reward counts correct tokens up to the first error.

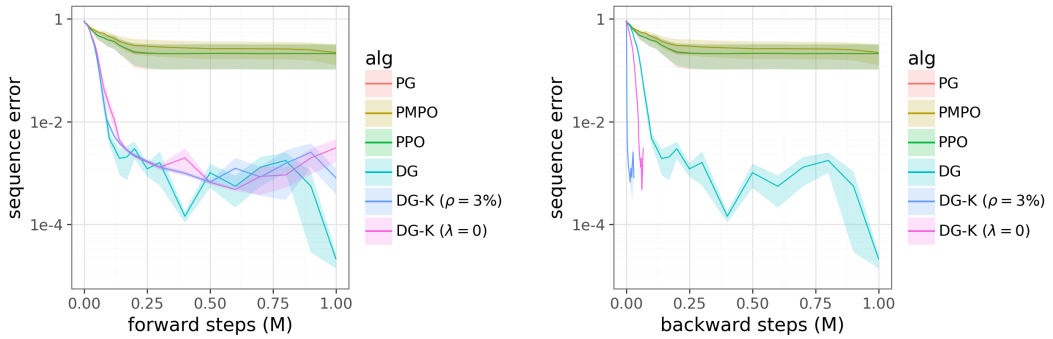
We compare PG, PPO, PMPO, DG (full delightful gradient), and two Kondo variants. DG-K ($\rho = 3\%$) imposes a fixed backward-compute budget, keeping only the top 3% of tokens by delight. DG-K

($\lambda = 0$) gates on the sign of delight and adapts automatically as the policy improves. We ask whether DG-K keeps DG's learning quality while collapsing the backward cost. PPO coincides with PG in this setting: each batch is fresh and receives a single update, so the PPO ratio is one and clipping never activates, and all methods share the same grouped baseline with no learned critic. PPO departs from PG under replay, multiple epochs, or stale actors [12]; we keep those complications out to isolate compute selection.

Figure 9 shows the representative result. In forward-pass space (a), DG and both DG-K variants outperform PG, PPO, and PMPO by over an order of magnitude. In backward-pass space (b), the DG-K curves reach the same quality as DG with orders of magnitude fewer backward passes.

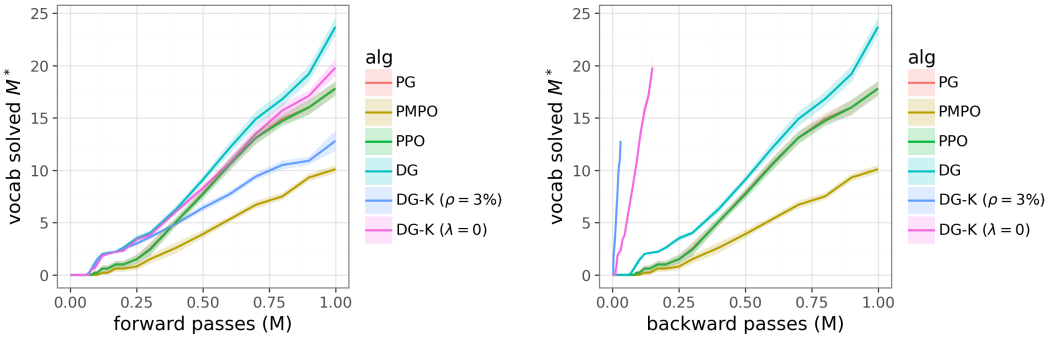
Figure 10 sweeps vocabulary size M, exposing the tradeoff between the two gates. In forward-pass space, the adaptive gate ($\lambda = 0$) tracks full DG and remains robust as M grows, whereas the fixed gate ($\rho = 3\%$) becomes too aggressive at large vocabularies. In backward-pass space, however, both Kondo variants still outperform all baselines. The adaptive gate is the safer default when difficulty is unknown; the fixed gate is attractive when backward savings matter most.

Figure 11 shows the main scaling result. As sequence length grows, PG, PPO, and PMPO scale sublinearly, while DG and DG-K ($\rho = 3\%$) solve longer sequences at every forward budget; DG-K slightly exceeds full DG at the longest lengths. In backward-pass space, DG-K ($\rho = 3\%$) solves $H^* \approx 29$ using a small fraction of the backward compute that DG needs to reach $H^* \approx 27$. The gap widens with sequence length as correct sequences become exponentially rarer, and here the fixed-budget gate gives both the best learning quality and the lowest backward cost.



(a) Forward passes: DG-K $\approx$ DG $\gg$ PG. (b) Backward passes: DG-K $\gg$ alternatives.

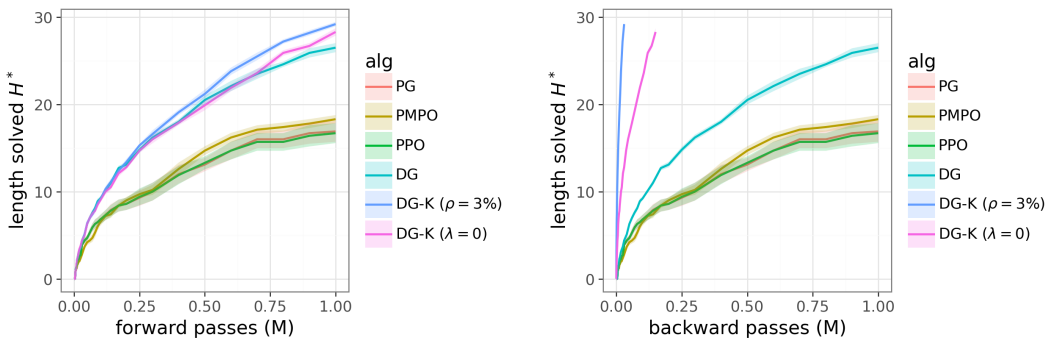
Figure 9: Token reversal learning curves ($H = 10, M = 2$). 10 seeds; shading ± 1 s.e.



(a) Vocab solved M^* (forward passes). (b) Vocab solved M^* (backward passes).

Figure 10: Scaling with vocabulary size: M^* = largest vocabulary solved vs. compute. Fixed $\rho = 3\%$ is too aggressive at large M; adaptive $\lambda = 0$ is robust and preserves backward-compute gains.

Across both scaling axes, the Kondo gate preserves DG's gains in forward-pass space and turns them into large gains in backward-pass space. As sequence length and vocabulary size grow, the gate solves harder problems at equal backward compute. This matches the bandit prediction that screening



(a) Length solved H^* (forward passes). (b) Length solved H^* (backward passes).

Figure 11: Scaling with sequence length: H^* = longest sequence solved (reward > 0.75) vs. compute. DG-K matches or exceeds DG and solves longer sequences at far lower backward cost.

becomes more valuable when useful learning events are rare. The adaptive gate ($\lambda = 0$) is reliable across problem difficulties; the fixed gate ($\rho = 3\%$) delivers the largest savings when tuned per task.

6 Related Work

We situate the Kondo gate among existing approaches to compute-efficient training, priority sampling, and robust policy gradients. The Kondo gate extends selective backpropagation to policy gradients, using advantage and surprisal instead of loss to decide which terms to compute.

Selective backpropagation and curriculum learning. Selective-Backprop [6] uses forward-pass losses to skip backward passes on easy examples, and importance sampling [8] favors high-loss samples, but loss is agnostic to gradient direction: high loss need not imply high learning value. Curriculum learning [3, 4] similarly prioritizes training examples by difficulty, but selects which data to present rather than which gradients to compute. Delight instead combines value and surprise, and Proposition 2 shows that its sign remains aligned with usefulness across regimes in which additive alternatives can mis-rank samples.

Prioritized experience replay. PER [17] and its distributed extension Ape-X [5] prioritize replay transitions by TD error, which conflates epistemic and aleatoric uncertainty. The Kondo gate prioritizes *gradient computation* instead of replay, and its priority signal comes from a single forward pass.

Speculative decoding. Speculative decoding [10] skips expensive inference steps using a cheaper draft model. The Kondo gate is the training counterpart, skipping expensive backward passes based on a cheap forward-pass signal. Section 3.2 shows that delight tolerates moderate noise, which supports the analogy.

UCB, active learning, and optimistic exploration. UCB-style methods [2] encourage exploration through an additive bonus. Delight's surprisal term is instead multiplicative: it prioritizes events that are both valuable and unexpected. Proposition 2 formalizes the difference: additive mixtures require regime-dependent tuning, while the product remains sign-consistent. The gate can also be viewed as within-batch active learning, but it requires no acquisition model: the selection criterion is a by-product of the forward pass.

GRPO, AWR, PMPO. These methods weight by advantage or exponentiated advantage but are surprisal-blind [20, 16, 1, 19, 18]. As M^H grows, gradient budget concentrates on predictable tokens whose advantage happened to be nonzero, starving rare informative events. The Kondo gate inherits DG's surprisal sensitivity and uses it to decide which backward passes are worth computing. The same principle applies to RLHF pipelines [15, 22], where backward passes through large models are especially expensive.

7 Conclusion

In the settings studied here, the forward pass is enough to decide whether a backward pass is worth computing. Delight, the product of advantage and surprisal, provides that signal, and the price λ sets the quality–cost trade-off. Across MNIST, bandits, and transformer token reversal, the Kondo gate skips most backward passes while retaining nearly all of DG's learning quality, yielding large gains over PG, PPO, and PMPO.

In our sequence tasks most gradient computations are redundant, and screening before backpropagation becomes more valuable as useful learning events become rarer and backward passes more expensive. The gate's robustness to approximate delight motivates a speculative-training direction: a cheap screening pass selects samples for a more expensive differentiable pass.

Because we count selected gradient terms instead of measuring wall-clock time, realizing the savings requires a two-stage implementation that scores samples without gradients and backpropagates only the selected ones, since a naive loss mask runs the same dense kernels; for token-level selection, a kept loss still backpropagates through its prefix, so selected terms do not translate directly into transformer FLOPs. The gated estimator is deliberately biased, and general convergence remains open, including under stochastic rewards, inaccurate advantages, and batch-dependent thresholds. Our tasks have unambiguous rewards and lack the ambiguity and stochasticity of preference-based learning, and the gambling regime of Proposition 3 marks where delight itself misleads. Exact delight needs the realized action and reward, so it can allocate gradient computation but cannot guide sampling; predicting delight before sampling defines a separate exploration method [13]. We also have not measured how gating affects the KL–performance trade-off or plasticity. The central open question is scale: do the same gains survive in modern large-model training?

Acknowledgements and Disclosures

We thank Ben Van Roy and Satinder Singh for detailed feedback on earlier drafts. We are grateful to Raia Hadsell, Zoubin Ghahramani, Demis Hassabis, and Satinder Singh for building the research environment at Google DeepMind that made this work possible. This work was funded by Google DeepMind, where the author is employed. The author has no other competing interests to declare.

References

[1] Abbas Abdolmaleki, Bilal Piot, Bobak Shahriari, Jost Tobias Springenberg, Tim Hertweck, Rishabh Joshi, Junhyuk Oh, Michael Bloesch, Thomas Lampe, Nicolas Heess, et al. Preference optimization as probabilistic inference. *arXiv preprint arXiv:2410.04166*, 2024.

[2] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

[4] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. *arXiv preprint arXiv:1704.03003*, 2017.

[5] Daniel Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. Distributed prioritized experience replay. In *International Conference on Learning Representations*, 2018.

[6] Angela H. Jiang, Daniel L.-K. Wong, Giulio Zhou, David G. Andersen, Jeffrey Dean, Gregory R. Ganger, Gauri Joshi, Michael Kaminsky, Michael Kozuch, Zachary C. Lipton, and Padmanabhan Pillai. Accelerating deep learning by focusing on the biggest losers. *arXiv preprint arXiv:1910.00762*, 2019.

[7] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[8] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International Conference on Machine Learning*, pages 2525–2534, 2018.

[9] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[10] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286, 2023.

[11] Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. Efficient large-scale language model training on GPU clusters using Megatron-LM. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021.

[12] Ian Osband. Delightful policy gradient under distributed friction. In *Advances in Neural Information Processing Systems*, 2026.

[13] Ian Osband. Delightful exploration, 2026. URL `https://arxiv.org/abs/2605.13287`.

[14] Ian Osband. Delightful policy gradient. *Reinforcement Learning Journal*, 7, 2026.

[15] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.

[16] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

[17] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

[18] John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proc. of ICML*, 2015.

[19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[20] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, 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.

[21] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[22] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A MNIST Diagnostic

We provide experimental details and supplementary figures for the MNIST experiments in Section 3.

A.1 Experimental Details

Architecture. The policy is a two-layer MLP with 100 hidden units per layer and softmax output over 10 actions (digits 0–9). The environment is an MNIST contextual bandit: the agent observes an image, selects an action, and receives reward $r = \mathbb{I}\{a = y\}$ where y is the true label.

Optimization. All methods use Adam with a learning rate swept over $\{10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 10^{-2}\}$. Each gradient step uses a batch of $B = 100$ samples drawn with replacement from the training set. Training runs for 10,000 gradient steps with validation every 100 steps on the full 10,000-image test set. All main-body figures average over 30 seeds; shading shows ± 1 standard error.

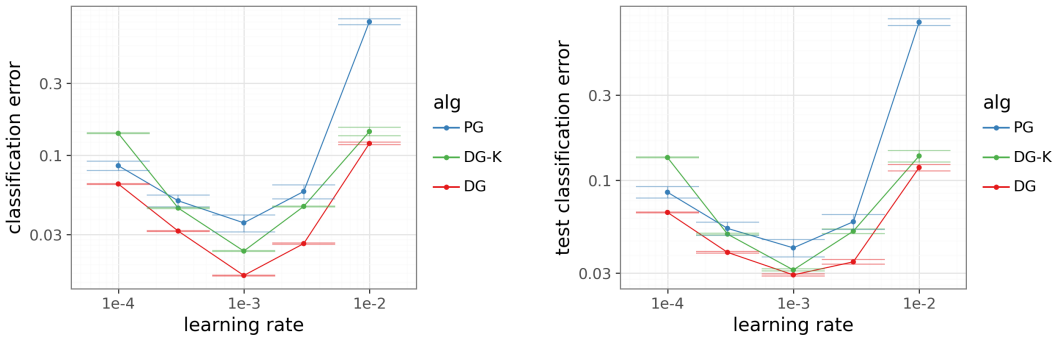
Baseline and advantage. All methods use the expected-confidence baseline $b = \sum_a \pi(a)^2$, the reward the policy would expect if its own sample were the label; it needs no access to the true label. Since $b \in (0, 1)$, this gives advantage $U(y^*) = 1 - b > 0$ for the correct action and $U(a \neq y^*) = -b < 0$ for incorrect actions.

Kondo gate. The gate rate ρ is implemented by keeping the top ρB samples by delight within each batch. We sweep $\rho \in \{0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.5, 1.0\}$; $\rho = 1$ recovers standard PG. For the priority comparison (Figure 6), we additionally test advantage-only, surprisal-only, absolute-advantage, uniform (random subsampling), and additive $(1-\alpha)U + \alpha\ell$ with $\alpha \in \{0, 0.1, \ldots, 1\}$.

Gambling experiment. For the gambling pathology (Figure 7), action $a = 0$ is designated the gamble: whenever the agent predicts 0 (regardless of true label), its reward receives additive $\mathcal{N}(0, \sigma_G^2)$ noise. Homoskedastic noise σ_R is applied to all actions. We sweep $\sigma_G \in \{0, 0.5, 1.0, 1.5, 2.0\}$ and $\sigma_R \in \{0, 0.5, 1.0, 2.0, 5.0\}$.

A.2 Learning Rate Sensitivity and Test Error

Figure 12 sweeps the learning rate for PG, DG, and DG-K ($\rho = 0.03$). All three methods share the same optimum at $\mathrm{lr} = 10^{-3}$, with DG dominating across the entire range and DG-K close behind. The pattern is nearly identical for training error (a) and test error (b): the Kondo gate is not exploiting a train/test gap.

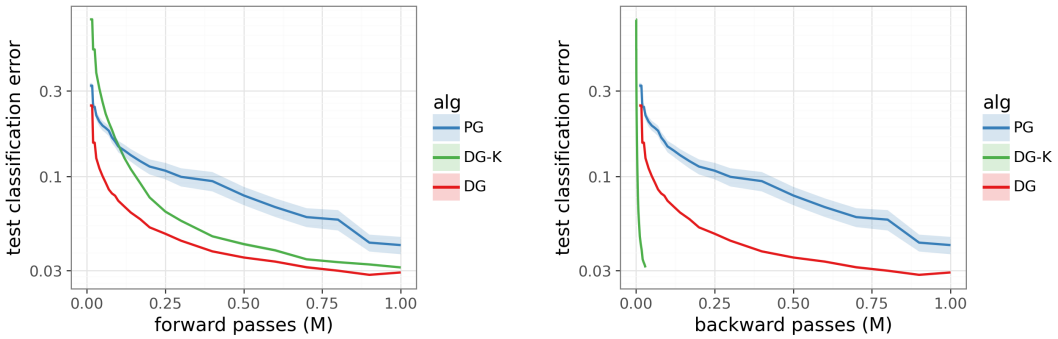


(a) Training error vs. learning rate.

(b) Test error vs. learning rate.

Figure 12: Learning rate sweep on MNIST. All methods are optimal near $\mathrm{lr} = 10^{-3}$; DG dominates across the range. Training and test error track closely, confirming no train/test gap.

Figure 13 replicates the main-body comparison (Figure 1) using test classification error rather than training error. The same pattern holds: DG-K matches DG in forward-pass space and dominates by two orders of magnitude in backward-pass space. This confirms that the Kondo gate's advantage is not specific to the RL reward signal; it transfers to the supervised-learning notion of generalization error that is more standard in MNIST benchmarks.



(a) Forward passes: test error.

(b) Backward passes: test error.

Figure 13: Test classification error on MNIST at $\rho = 0.03$. Same comparison as Figure 1 but with held-out test error. DG-K matches DG in forward-pass space and dominates in backward-pass space; the gate's advantage generalizes beyond training error.

A.3 Baseline Robustness

The main-body results use the expected-confidence baseline $b = \sum_a \pi(a)^2$, which Osband [14] identifies as the natural choice for MNIST. To check that the Kondo gate's advantage is not an artifact of this choice, Figures 14 and 15 repeat the comparison under four baselines: zero ($b = 0$), constant ($b = 0.5$), expected confidence ($b = \sum_a \pi(a)^2$), and oracle ($b = \mathbb{E}[R \mid x]$ using the true label).

The general pattern is the same across all four baselines: DG dominates PG, DG-K matches DG in forward-pass space, and DG-K dominates in backward-pass space. Under the zero baseline, DG-K actually outperforms DG on forward passes; we report the expected baseline in the main body to avoid cherry-picking.

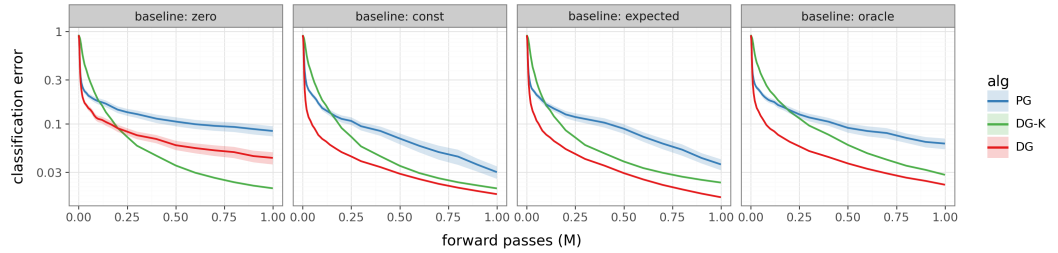


Figure 14: Forward-pass comparison across baselines on MNIST at $\rho = 0.03$. The Kondo gate matches or exceeds DG under all four baselines. Under the zero baseline, DG-K even outperforms DG in forward-pass space.

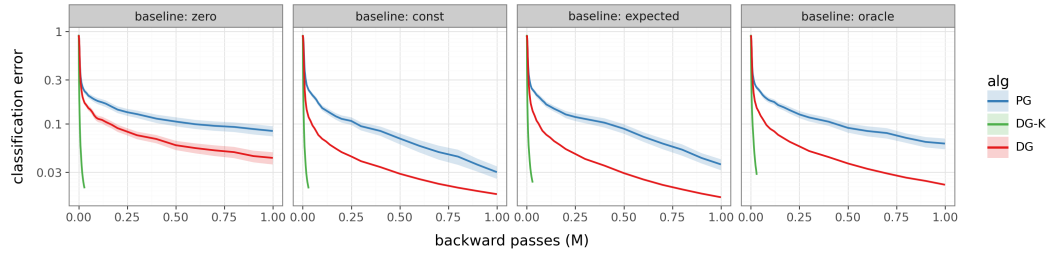


Figure 15: Backward-pass comparison across baselines on MNIST at $\rho = 0.03$. The Kondo gate dominates in backward-pass space under all four baselines: the two-orders-of-magnitude advantage is not baseline-dependent.

A.4 Gate Selection Profile

Figure 16 shows the empirical CDF of $\pi(y^*)$ for kept vs. skipped samples at three stages of training, aggregated over 100 batches (10,000 samples per stage). At step 100, both distributions nearly overlap: the policy is too uncertain currently for delight to discriminate. By step 1,000, clear separation emerges: kept samples (red) have systematically lower $\pi(y^*)$ than skipped samples (blue), demonstrating first-order stochastic dominance. The gate targets the *learning frontier*: samples the policy predicts correctly but without confidence. By step 10,000, both distributions concentrate near $\pi(y^*) \approx 1$, but the kept distribution remains shifted left, finding the few remaining hard cases.

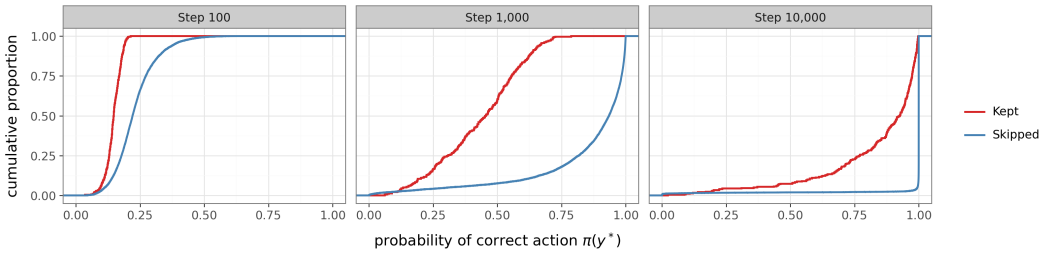


Figure 16: Empirical CDF of $\pi(y^*)$ for kept vs. skipped samples at three training stages ($\rho = 0.03$, 10,000 samples per stage). Kept samples have systematically lower $\pi(y^*)$: the gate selects the learning frontier where the model is correct but uncertain.

A.5 Kept vs. Skipped Images

Figure 17 shows images the Kondo gate keeps versus those it skips at $\rho = 0.03$ across three training stages. Each image is annotated with its true label y, the action selected a, and the probability of the correct action $p = \pi(y^*)$.

At step 100 (a), the model is too uncertain for the gate to make meaningful distinctions: both rows contain diverse digits with low p. By step 1,000 (b), separation emerges: kept images have moderate p (the model is learning but not yet confident), while skipped images are either already solved ($p \approx 1$) or hopelessly misclassified. At step 10,000 (c), the contrast is sharpest: kept images are correctly classified ($a = y$) but with $p < 1$, the learning frontier. Skipped images include both solved cases ($p = 1.00$) and a few confidently wrong predictions ($a \neq y$, very low p), where the negative advantage makes delight negative.

A.6 Delight Noise Robustness

Figure 18 complements Figure 5a (Section 3.2) by showing delight noise robustness on an absolute rather than relative scale. The qualitative pattern is the same: DG tolerates substantial noise before degrading; DG-K is more fragile. The relative-scale figure in the main text is more interpretable for the approximate-delight argument, since the noise level is normalized by the signal magnitude.

B Derivation of the Gate Weight

We derive the closed-form gate weight used in Algorithm 1 (Section 2.1).

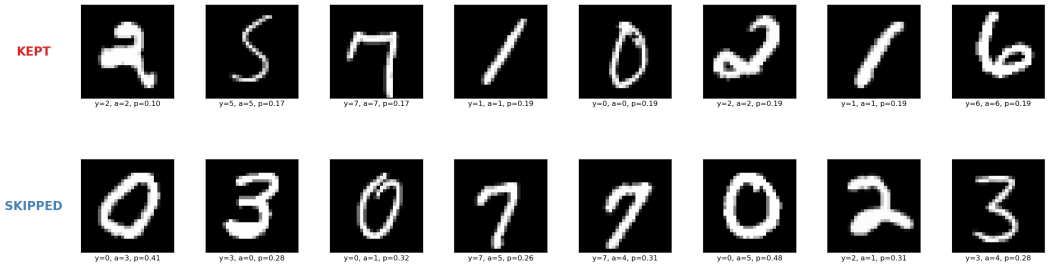
The meta-learner's objective for a single sample is

$$f(w) = \chi\, w - \lambda\, w + \eta\, H(w), \qquad w \in [0, 1],$$

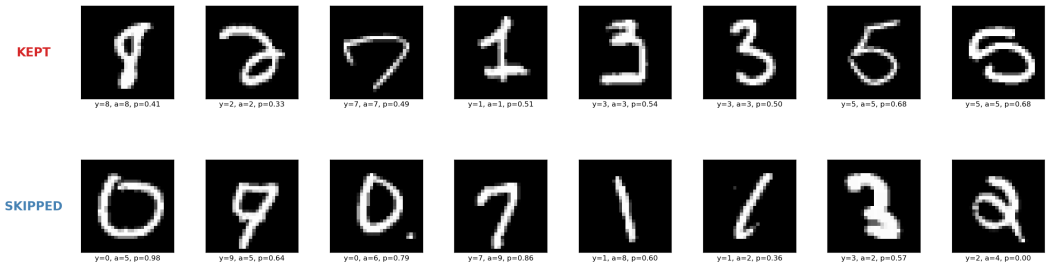
where $H(w) = -w \log w - (1-w)\log(1-w)$ is binary entropy. Differentiating and setting to zero:

$$f'(w) = (\chi - \lambda) + \eta\big(\log(1-w) - \log w\big) = 0 \quad \Longrightarrow \quad \frac{\chi - \lambda}{\eta} = \log \frac{w}{1-w}.$$

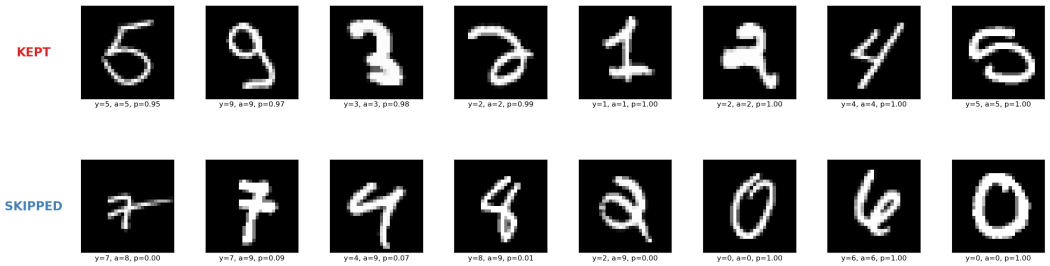
The right-hand side is the logit of w, so inverting gives $w^* = \sigma\big((\chi - \lambda)/\eta\big)$. Since H is strictly concave and the linear terms preserve concavity, f is strictly concave, and the maximizer is unique.



(a) Step 100: the model is too uncertain to meaningfully discriminate.



(b) Step 1,000: separation emerges; kept images have moderate $\pi(y^*)$.



(c) Step 10,000: kept images are correct but uncertain ($a = y$, $p < 1$); skipped are solved or confidently wrong.

Figure 17: Images kept vs. skipped by the Kondo gate at $\rho = 0.03$ across three training stages. Each image shows ground truth y, selected action a, and $p = \pi(y^*)$. The gate progressively concentrates compute on the learning frontier.

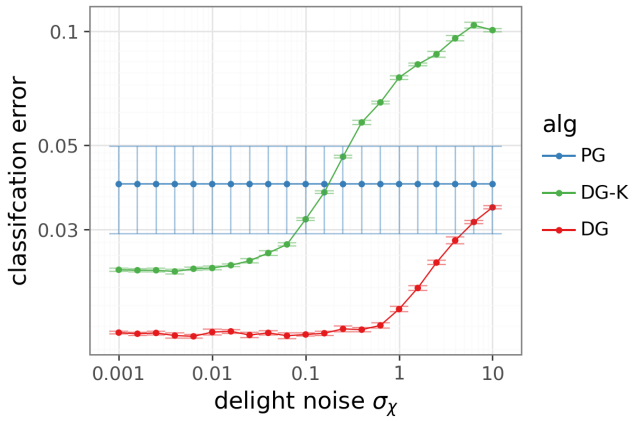


Figure 18: Delight noise robustness (absolute scale): classification error vs. absolute noise σ_χ. Complements Figure 5a; same qualitative pattern on an absolute scale.

The expected gated update for sample t is therefore

$$\mathbb{E}[G_t\, g_t] = w_t^* \cdot U_t \nabla_\theta \log \pi_\theta(A_t \mid \mathcal{H}_t), \qquad w_t^* = \sigma\big((\chi_t - \lambda)/\eta\big).$$

When $\lambda = 0$, this coincides with the DG update: the stochastic Kondo gate is a sparse Monte Carlo estimator of DG.

C Tabular Analysis

This appendix provides the geometric setup and proofs for the three propositions in Section 4: the Kondo gate Pareto improvement (Proposition 1), delight sign-consistency (Proposition 2), and the gambling pathology (Proposition 3). Propositions 1 and 2 share a common bandit setup and geometry lemma, stated first; Proposition 3 states its own two-armed stochastic model.

C.1 Setup and Geometry Lemma

The following assumption underpins Propositions 1 and 2.

Assumption 1 (Symmetric K-armed bandit)**.** *K arms ($K \geq 3$), a single correct arm y^*. Deterministic reward $R = \mathbb{I}\{A = y^*\}$ (Section C.4 extends to stochastic). Softmax policy with success probability $p := \pi(y^*) \in (0,1)$ and uniform incorrect: $\pi(a) = (1-p)/(K-1)$ for $a \neq y^*$. Baseline $b \in (0,1)$. Score $\phi_\pi(a) := e_a - \pi$ (logit-space gradient of $\log \pi(a)$). Batch size B; normalized step $z^+ = z + \alpha\, \bar{g}/\|\bar{g}\|$.*

Each sample yields a per-sample gradient $g(a) = U(a)\, \phi_\pi(a)$, where the advantage takes two values:

$$U(y^*) = 1 - b > 0, \qquad U(a \neq y^*) = -b < 0. \tag{2}$$

Correct-action gradients carry pure signal; incorrect-action gradients carry mostly noise.

Lemma 1 (Softmax gradient geometry)**.** *Under Assumption 1, the true gradient is $\nabla_z J = p\, \phi_\pi(y^*)$. Write $\Pi_\perp$ for orthogonal projection away from $\nabla_z J$, and $\mathrm{Var}_\perp(g) := \mathbb{E}\|\Pi_\perp(g - \mathbb{E}g)\|^2$.*

1. ***Correct action:*** *$\phi_\pi(y^*)$ is a positive scalar multiple of $\nabla_z J$ with $\|\phi_\pi(y^*)\| = \Theta(1-p)$. $\Pi_\perp(\phi_\pi(y^*)) = 0$.*
2. ***Incorrect action:*** *$\|\phi_\pi(a)\| = \Theta(1)$ and the cosine with $\nabla_z J$ is $\Theta(p)$:*

$$\frac{|\langle \phi_\pi(a),\, \nabla_z J\rangle|}{\|\phi_\pi(a)\| \cdot \|\nabla_z J\|} = \Theta(p).$$

Each incorrect-action gradient term has $\Theta(1)$ perpendicular noise and only an $O(p)$ fraction aligned with the true gradient.

Proof. **Correct action.** $\phi_\pi(y^*) = e_{y^*} - \pi$. Writing $p_a := (1-p)/(K-1)$ for the uniform incorrect probability: $\|\phi_\pi(y^*)\|^2 = (1-p)^2 + (K-1)\cdot p_a^2 = (1-p)^2 \cdot K/(K-1) = \Theta((1-p)^2)$. Since $\nabla_z J = p\, \phi_\pi(y^*)$, $\phi_\pi(y^*)$ is a positive scalar multiple of $\nabla_z J$, so $\Pi_\perp(\phi_\pi(y^*)) = 0$.

Incorrect action. For $a \neq y^*$: $\phi_\pi(a) = e_a - \pi$ has $\|\phi_\pi(a)\|^2 = 1 - 2p_a + \|\pi\|^2 = \Theta(1)$. The inner product with $\phi_\pi(y^*)$ is:

$$\langle \phi_\pi(a), \phi_\pi(y^*)\rangle = \langle e_a - \pi,\, e_{y^*} - \pi\rangle = -p - \pi(a) + \|\pi\|^2 = -\frac{p(1-p)K}{K-1}.$$

So $\langle \phi_\pi(a), \nabla_z J\rangle = -p^2(1-p)\, K/(K-1) = -\Theta(p^2)$. The cosine is $|-\Theta(p^2)|/(\Theta(1)\cdot\Theta(p)) = \Theta(p)$. Since $\cos(\phi_\pi(a), \nabla_z J) = \Theta(p) \ll 1$, the perpendicular component is $\Theta(1)$. □

Remark 1 (The arithmetic of noise)**.** *A batch of B samples contains $\sim pB$ correct draws and $\sim (1-p)B$ incorrect draws. For a baseline b bounded away from 0 and 1, the signal (parallel to $\nabla_z J$) scales as $\Theta(pB)$ and the noise (perpendicular) as $\Theta(\sqrt{B})$; for $b \to 0$ the incorrect-action noise shrinks with b. When $p^2 B \ll 1$: $\cos(\bar{g}, \nabla_z J) \approx p\sqrt{B}$; the gradient is nearly random unless $B \gg 1/p^2$. When $p^2 B \gg 1$: signal dominates noise.*

C.2 Proof of Proposition 1

The gate at $\lambda = 0$ skips whenever $\chi < 0$. Since $\ell > 0$ always, $\chi < 0$ iff $U < 0$ iff the action was incorrect. The gate keeps only correct-action samples.

Proof. **Part 1: Direction.** The gate fires iff $A = y^*$ (probability p). Every kept term is $(1-b)\,\phi_\pi(y^*)$. $\mathbb{E}[g_{\mathrm{KG}}] = p(1-b)\,\phi_\pi(y^*) = (1-b)\,\nabla_z J$. Under normalized steps, the scale $(1-b)$ does not affect direction.

Part 2: Variance. Every kept term is the same vector $(1-b)\,\phi_\pi(y^*)$. Every kept term lies along $\nabla_z J$, so $\mathrm{Var}_\perp = 0$. For PG, each incorrect sample contributes $\|\Pi_\perp(-b\,\phi_\pi(a))\|^2 = b^2 \cdot \Theta(1)$, arriving with probability $1 - p$. Per-sample: $\mathrm{Var}_\perp(g_{\mathrm{PG}}) = (1-p) \cdot b^2 \cdot \Theta(1)$.

Part 3: Cost. $\Pr(\text{gate fires}) = p$. Expected backward passes: pB.

Part 4: Alignment. The KG batch gradient (given at least one correct draw) is $\bar{g}_{\mathrm{KG}} = (1-b)\,\phi_\pi(y^*)$, deterministic in direction. $\cos(\bar{g}_{\mathrm{KG}}, \nabla_z J) = 1$. For b bounded away from zero, PG's batch cosine is $\Theta(p\sqrt{B})$ (Remark 1), requiring $B = \Omega(1/p^2)$ to approach 1. KG needs at least one correct draw, which occurs with probability $1 - (1-p)^B \geq 1 - \delta$ once $B \geq \log(1/\delta)/p$. □

C.3 Proof of Proposition 2

We separate sign consistency (Part 1) from the additive failure mode (Part 2).

Proof. **Part 1: Sign consistency.** $\chi(a) = U(a) \cdot \ell(a)$. $\ell(a) = -\log \pi(a) > 0$ for any $\pi(a) < 1$. So $\mathrm{sgn}(\chi) = \mathrm{sgn}(U)$. $U(y^*) = 1 - b > 0$; $U(a \neq y^*) = -b < 0$.

Part 2: Additive failure. For any baseline b: $U(y^*) = 1 - b$, $U(a) = -b$, $\ell(y^*) = -\log p$, $\ell(a) = \log((K-1)/(1-p))$. The additive scores $f_\alpha = (1-\alpha)U + \alpha\ell$ are:

$$f_\alpha(y^*) = (1-\alpha)(1-b) - \alpha \log p,$$
$$f_\alpha(a) = -(1-\alpha)b + \alpha \log \frac{K-1}{1-p}.$$

Taking the difference and simplifying:

$$f_\alpha(y^*) - f_\alpha(a) = (1-\alpha) - \alpha \log \frac{p(K-1)}{1-p} = (1-\alpha) - \alpha\, L.$$

When $p \leq 1/K$: $L \leq 0$ and the difference is positive for every $\alpha \in [0, 1)$. When $p > 1/K$: $L > 0$ and the difference is positive iff $\alpha < 1/(1+L) = \alpha^*$.

α^* shrinks with both action-space size and policy quality:

(K, p)	$L = \log(p(K-1)/(1-p))$	α^*
$(10,\ 0.5)$	$\log 9 \approx 2.2$	0.31
$(100,\ 0.5)$	$\log 99 \approx 4.6$	0.18
$(100,\ 0.9)$	$\log 891 \approx 6.8$	0.13
$(50000,\ 0.5)$	$\log 49999 \approx 10.8$	0.08

As K or p grows, α must approach 0 (pure advantage), losing the information axis of delight. □

C.4 Proof of Proposition 3

We extend the bandit to stochastic rewards: arm 1 pays μ^* deterministically, arm 2 pays $R_2 \sim \mathcal{N}(\mu^* - \Delta, \sigma^2)$. Policy $\pi(1) = 1 - \varepsilon$; baseline $b = V^\pi = \mu^* - \varepsilon\Delta$.

Proof. **Part 1: Reliable regime.** $U_2 \mid A{=}2$ is Gaussian with mean $-(1{-}\varepsilon)\Delta$ and variance σ^2. By the Gaussian tail bound: $\Pr(U_2 > 0 \mid A{=}2) = \Pr(R_2 > b) \leq \exp(-(1{-}\varepsilon)^2\Delta^2/(2\sigma^2))$.

Part 2: Pathological regime. $\Pr(U_2 > 0 \mid A{=}2) = 1 - \Phi\big((1{-}\varepsilon)\Delta/\sigma\big)$. When $\sigma \gg \Delta$, the argument is $o(1)$, so this probability is bounded away from 0: $\Pr(U_2 > 0 \mid A{=}2) = \Theta(1)$.

Part 3: Amplification. $\{U_2 > 0\} = \{R_2 > b\}$ depends only on the reward distribution, not the priority signal. Given $U_2 > 0$, advantage-priority assigns weight $|U_2|$, while delight assigns $|U_2| \cdot \ell_2$. Since $\ell_2 = \log(1/\varepsilon) \to \infty$ as $\varepsilon \to 0$, delight inflates the false positive by a factor that grows as the policy improves. □

Remark 2 (Limits of per-sample screening)**.** *No per-sample statistic computed from (R, π) can distinguish a genuine breakthrough from a lucky draw. The distinction requires either multi-trial statistics or a learned reward model. The gambling pathology is a property of the reward structure ($\sigma/\Delta \gg 1$), and every advantage-based gate inherits it.*

D Token Reversal

We provide experimental details and supplementary scaling metrics for the token reversal experiments in Section 5.

D.1 Experimental Details

Architecture. The agent is a decoder-only Transformer with causal attention, model dimension $d_{\text{model}} = 64$, 3 layers, 4 attention heads, feed-forward dimension 128, and learned position embeddings (about 100K parameters), as in Osband [14].

Environment. The token reversal task presents a prompt of H tokens drawn uniformly from a vocabulary of size M; the agent must output the tokens in reverse order. The per-episode reward is $R = N/H \in [0, 1]$, where N is the number of consecutive correct tokens before the first error.

Baseline. All methods use a grouped empirical baseline: each batch consists of $P = 10$ prompts with $S = 10$ sampled responses each, and the baseline for each prompt is the mean reward across its responses. This is the group-mean baseline of GRPO [20] without its standard-deviation normalization; any other value function could be substituted.

Optimization. All methods use Adam with learning rate 10^{-4}, except PG, whose learning rate is tuned. Training runs for K gradient steps with batch size $P \times S = 100$ episodes per step. For the learning curve experiments (Figure 9), $K = 3{,}000$; for the scaling sweeps (Figures 10–11), $K = 10{,}000$. All figures average over 10 seeds.

Kondo gate. We test the Kondo gate at fixed gate rates $\rho \in \{0.03, 0.05, 0.1, 0.2, 0.5, 1.0\}$ and in adaptive mode ($\lambda = 0$, variable ρ). The priority signal for screening is delight.

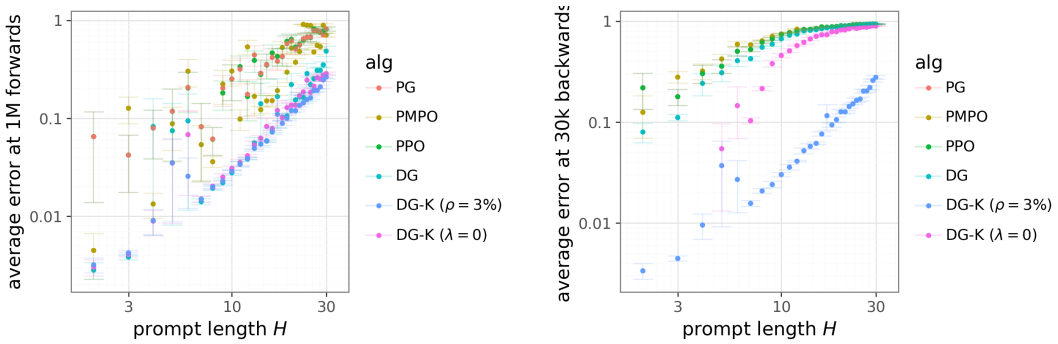
Baselines. We compare against three standard reinforcement learning methods within our codebase: PG (importance-weighted REINFORCE with no clipping), PPO [19] with $\epsilon = 0.2$ and $\beta_{\text{KL}} = 0$, and PMPO [1] with $\alpha = 1$ and $\beta_{\text{KL}} = 0$. All baselines use identical architecture, optimizer, and grouped baseline.

Scaling protocol. For the vocab scaling sweep (Figure 10), we fix $H = 10$ and sweep $M \in \{2, 3, \ldots, 30\}$. For the length scaling sweep (Figure 11), we fix $M = 2$ and sweep $H \in \{2, 3, \ldots, 30\}$. A problem is considered solved once the average reward over training exceeds 0.75 (0.7 for the vocabulary sweep), and H^* and M^* never decrease. The main-body figures report M^* (largest M solved) and H^* (largest H solved) as functions of forward or backward compute.

D.2 Average Error Scaling

The main-body scaling figures (Figures 10–11) report the largest problem solved as a function of compute. A complementary view is the average error across all problem sizes at a fixed compute budget, which captures how gracefully each method degrades as problems exceed its capacity.

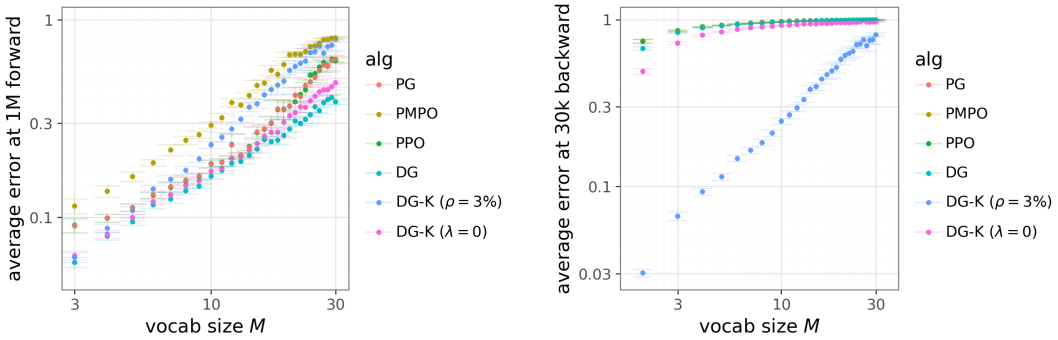
Figure 19 plots average error against sequence length H on a log–log scale. All methods follow roughly linear trends on log–log axes, and the DG family (DG, DG-K) occupies a uniformly lower curve than PG, PPO, and PMPO. In backward-pass space (b), DG-K ($\rho = 3\%$) separates dramatically from the pack, achieving the same average error as full DG at a small fraction of the backward cost.



(a) Forward passes. (b) Backward passes.

Figure 19: Average error vs. sequence length H (log–log, $M = 2$). DG-K compresses the backward axis.

Figure 20 shows the same comparison over vocabulary size M. As with sequence length, DG and DG-K ($\lambda = 0$) trace the lowest curves in forward-pass space, while DG-K ($\rho = 3\%$) is best in backward-pass space. As M grows and informative tokens become rarer, the gate's backward-pass advantage widens.



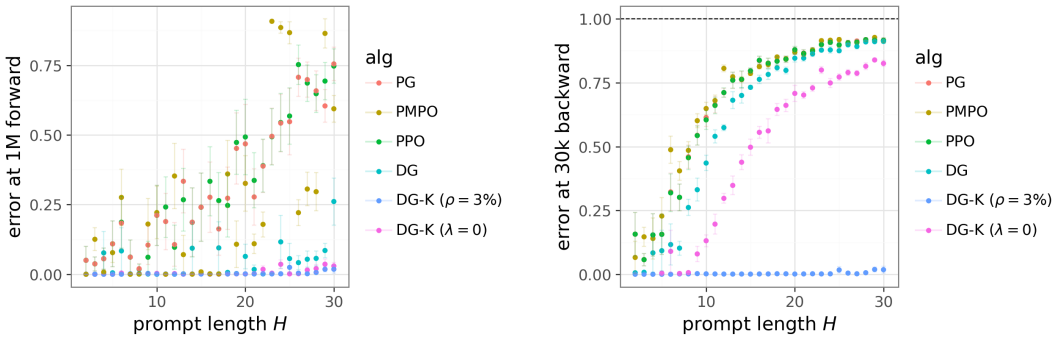
(a) Forward passes. (b) Backward passes.

Figure 20: Average error vs. vocabulary size M (log–log, $H = 10$). The gate's backward-pass advantage widens with M.

D.3 Final Error Scaling

The average error view aggregates performance across the full training trajectory. An alternative is to examine the final error at a fixed compute budget, which isolates where each method converges given a finite allocation.

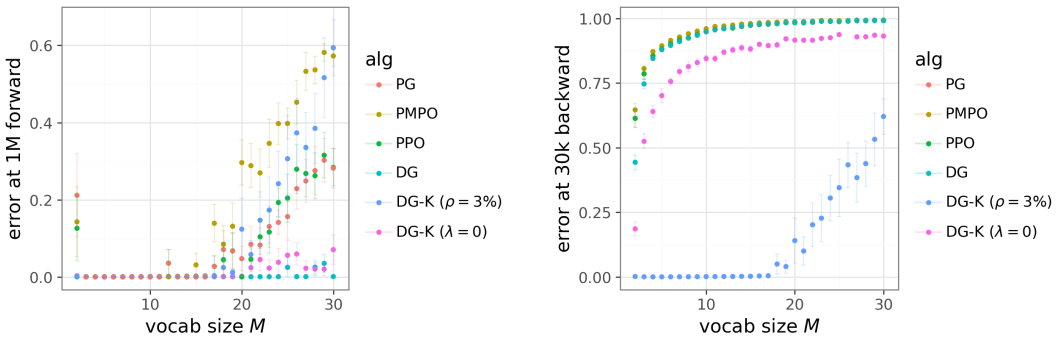
Figure 21 plots final error against sequence length H. DG and both DG-K variants maintain near-zero final error across a wide range of H, while PG, PPO, and PMPO degrade steadily. In backward-pass space (b), the advantage is stark: DG-K ($\rho = 3\%$) stays near zero across almost the entire range, while baselines climb steeply.



(a) Forward passes. (b) Backward passes.

Figure 21: Final error vs. sequence length H ($M = 2$). DG-K stays near zero where baselines degrade.

Figure 22 shows the same over vocabulary size M. The adaptive gate ($\lambda = 0$) tracks full DG faithfully, while the fixed gate ($\rho = 3\%$) degrades at large M where a fixed 3% budget becomes too aggressive. In backward-pass space, both Kondo variants still outperform all baselines across the full range.



(a) Forward passes. (b) Backward passes.

Figure 22: Final error vs. vocabulary size M ($H = 10$). Adaptive gate tracks DG; fixed gate degrades at large M but dominates in backward-pass space.

E Limitations and Scope

The main-body conclusion states the main limitations: savings counted rather than timed, deliberate bias with open convergence, tasks without preference-style reward ambiguity, the gambling pathology, and scale. Here we collect additional caveats.

Backward-pass counts vs. wall-clock time. Our experiments measure backward-pass counts, not accelerator wall-clock time. This isolates whether learning quality survives sparse backward computation, but it does not imply that a naive loss mask yields proportional wall-clock savings. Real speedups require implementations that avoid differentiating skipped samples, for example through example-level gating, sequence packing, or a two-stage screening pipeline.

Credit assignment at the token level. In sequence-level reward settings, token-level delight inherits the usual credit-assignment ambiguity: a high-surprisal token in a successful response may be incidental rather than causal. Span-level or trajectory-level gates may be more appropriate in some language-model applications.

Advantage estimation. The gate depends on the sign and scale of the advantage U_t. Errors in the baseline can change which samples receive backward passes. Our MNIST experiments show baseline

robustness (Appendix A.3), but in RLHF or long-horizon RL, advantage estimation noise is a central concern.

Negative evidence. Positive-delight gating systematically skips negative-advantage samples. While Proposition 1 shows this is beneficial in the deterministic-reward setting, rare catastrophic failures or newly discovered traps may carry useful negative evidence that the gate discards. A magnitude-based gate ($|\chi|$) or a two-sided gate could address this, though at the cost of reintroducing some perpendicular noise.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction claim that the Kondo gate skips most backward passes while retaining nearly all of DG's learning quality, with gains that grow as problems get harder. These claims are supported by experiments on MNIST (Section 3), tabular bandits (Section 4), and transformer token reversal (Section 5). Limitations are stated explicitly in the Conclusion (Section 7) and Appendix E.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Conclusion (Section 7) states the main limitations: savings are counted in selected gradient terms rather than wall-clock time, the gated estimator is deliberately biased with general convergence open, the tasks lack the reward ambiguity of preference-based learning, and the gambling pathology (Proposition 3) marks where delight misleads. Appendix E provides extended discussion of additional caveats including credit assignment at the token level, advantage estimation sensitivity, and the systematic exclusion of negative evidence. The scale question is identified as the central open problem.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Propositions 1 and 2 use Assumption 1, stated in full in Appendix C; Proposition 3 states its own two-armed stochastic model. Complete proofs are provided in the appendix with labeled paragraphs for each part. The gate derivation appears in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Full experimental details are provided: MNIST setup in Appendix A.1 (architecture, optimizer, learning rate sweeps, baseline, gate rate sweeps) and token reversal in Appendix D.1 (architecture, environment, baseline, optimizer, gate configurations, scaling protocol). Algorithm 1 specifies the method precisely. All hyperparameter values and sweep ranges are listed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is available at `https://github.com/google-deepmind/egg`. The experiments use MNIST (publicly available) and a synthetic token reversal task fully specified in Appendix D.1. No proprietary data is required.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix A.1 specifies the MNIST architecture, optimizer (Adam), learning rate sweep, batch size, number of gradient steps, validation schedule, baseline, and gate rate sweep. Appendix D.1 specifies the transformer architecture, environment definition, baseline (grouped empirical, analogous to GRPO), optimizer, training length, gate configurations, and scaling protocol. All seed counts are reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All MNIST experiments average over 30 seeds with ± 1 standard error shading (Section 3.1). All token reversal experiments average over 10 seeds with ± 1 standard error shading (Section 5). Error bars are clearly identified as standard error of the mean in figure captions.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Specific hardware types and wall-clock times are not reported. All experiments are small-scale (two-layer MLP on MNIST, 3-layer transformer on token reversal) and run on a single CPU or GPU in minutes to hours.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: The research uses only publicly available data (MNIST) and synthetic environments (token reversal). No human subjects, private data, or dual-use concerns are involved.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: This is foundational research on compute-efficient gradient estimation for reinforcement learning. The method reduces training compute, which has a positive environmental impact. There is no direct path to negative applications beyond what applies to any generic improvement in neural network training efficiency.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release pre-trained models, scraped datasets, or any artifacts with high misuse risk. The released code implements a gradient-gating algorithm for small-scale RL experiments.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: MNIST is cited via LeCun et al. [9]. All baseline methods (PG, PPO, PMPO, DG) are cited to their original papers. The codebase is released under the license specified in the repository.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code release includes the Kondo gate implementation, all experimental configurations, and instructions for reproducing the results. The token reversal environment is fully specified in Appendix D.1.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as a component of the core method. The Kondo gate is a gradient-gating mechanism applied to standard policy gradient updates; no LLM is involved in the algorithm or experiments.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.