# CAPABILITY–LOSS DISSOCIATION IN LANGUAGE MODELS: A THEORY AND EMPIRICAL STUDY OF LOSS SIGNATURES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The standard measure of language-model progress is validation loss, but the ability of validation loss to detect a capability depends on how often the capability is exercised in the evaluation corpus. We formalize this dependence as the *loss signature* of a capability: the product of its token share and its per-token information gain, which is exactly the change in aggregate validation loss from acquiring it, and which gives a criterion for when loss is an appropriate measurement tool. We assess this account on a synthetic dual-route language-modeling task that separates parametric memorization from in-context retrieval. Across 160 controlled training runs whose variants independently vary token share and information content, the predicted signature tracks the observed loss gap, and models with near-identical validation loss differ widely in in-context accuracy, including a loss-matched pair with a $188\times$ difference in ability. A gradient-level intervention on 76M-parameter models reproduces the dissociation causally, yielding a 69.5-point in-context accuracy gap between checkpoints at nearly identical aggregate loss ($\Delta L = 0.0027$ nats). Direct measurement of the leakage term shows the capability's own signature ($f_c \Delta L_c = 0.1601$ nats, 2.5% of operating loss) is almost exactly offset by a compensating gain on the competing memorization route ($\Lambda = -0.1574$ nats), and an ablation over five intervention weights shows this trade is smooth and monotone. On WikiText-103, induction is clearly visible in validation loss because its token share is large, which shows the framework is conditional rather than a blanket criticism of loss. We further derive a sample-size relationship showing that role-resolved evaluation scales more favourably than aggregate loss for rare capabilities. Our results support a practical principle: before using validation loss to measure a capability, estimate its token share and information content; when the resulting loss signature is small, or when a competing route can absorb it, capability-specific evaluation or role-resolved loss is required.

## 1 INTRODUCTION

Held-out cross-entropy is the standard metric for language-model development, but averaging over tokens can hide capabilities that affect only a small part of the sequence. This helps explain why perplexity and capability benchmarks can move differently (Hendrycks et al., 2021; Schaeffer et al., 2023). We ask a more precise question: how much validation loss is a capability worth?

Validation loss gives an exact answer once tokens are partitioned into roles. If a capability affects only role $c$, its loss contribution is the role's token share times its per-token information gain (Shannon, 1948); we call this the *loss signature*. If other roles change as well, their contribution appears as an explicit leakage term.

We test the accounting in a synthetic dual-route setting that contrasts parametric memorization with in-context learning (Brown et al., 2020), while both are scored by the same cross-entropy. Our contributions are:

1. Exact accounting of a capability's loss contribution (Proposition 1), its token-share dependence (Corollary 1), and a leakage term for cases where other roles move (Proposition 2).



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.

2. A dual-route sandbox separating memory from context, plus a conflict probe for route preference.
3. Verification across 160 runs: the capability cost is within $2\,\%$ of the bound, while 336 iso-loss pairs differ by up to $0.948$ in in-context accuracy.
4. A causal replication at 76M parameters, where a 69.5-point capability shift at $\Delta L = 0.0027$ nats is offset by leakage ($\Lambda = -0.1574$).
5. Natural-text, detectability, power, and public-checkpoint analyses (Appendices C, D, G).

The identity is simple, but it makes "loss may hide capability" measurable before training. We use it to measure hiding for rare or substitutable capabilities, including under a designed causal intervention at $760\times$ scale.

## 2 Related Work

**Loss, scale, and capability.** Scaling laws connect loss to compute and model size (Kaplan et al., 2020; Hoffmann et al., 2022), while some capabilities appear abruptly (Wei et al., 2022; Ganguli et al., 2022). Such behavior can coexist with smooth loss because a capability's loss imprint is weighted by its token share; our focus is this measurement problem rather than population-level scaling laws.

**Hidden progress.** Aggregate loss can miss meaningful progress in settings such as parity and grokking (Barak et al., 2022; Power et al., 2022; Nanda et al., 2023). We quantify when this happens: role decomposition gives a capability's maximum possible loss shift before the experiment is run.

**Memorization and in-context learning.** Memorization, capacity, and rare-data effects are well studied (Carlini et al., 2022; Tirumala et al., 2022; Allen-Zhu & Li, 2024; Feldman, 2020). Our sandbox follows work on in-context bigrams and induction heads (Bietti et al., 2023; Olsson et al., 2022), but adds an explicit loss accounting and a causal intervention (Section 6). Capacity trade-offs (Elhage et al., 2022) and position effects (Liu et al., 2024) motivate our controls.

## 3 Loss Signatures of Capabilities

Let a model $\theta$ be scored by mean next-token cross-entropy on a fixed held-out set with scored positions $\mathcal{P}$,

$$L(\theta) \;=\; \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} -\log p_\theta(x_{p+1} \mid x_{\le p})\,. \tag{1}$$

Partition $\mathcal{P}$ into roles $\mathcal{R}$, writing $f_r$ for the share of scored positions in role $r$ and $L_r(\theta)$ for the mean loss over that role. Since the partition is exhaustive and disjoint,

$$L(\theta) \;=\; \sum_{r \in \mathcal{R}} f_r\, L_r(\theta), \qquad \sum_r f_r = 1. \tag{2}$$

Equation (2) is exact. The $f_r$ depend only on the validation set, so they are shared by every evaluated model.

**Proposition 1** (Loss signature)**.** *Let $C$ be the ability to predict on role $c$ and leave all other roles the same. Let $L_c^0$ be the optimal loss on role $c$ without $C$, and $L_c^\star$ the loss with $C$. Then the difference in validation-loss is exactly*

$$\Delta \;=\; f_c \left(L_c^0 - L_c^\star\right). \tag{3}$$

*If role $c$ is unpredictable outside its marginal, which is uniform over an outcome set $O$, when $C$ is absent, and if $C$ determines it exactly, then $L_c^0 = \ln|O|$, $L_c^\star \to 0$ and*

$$\Delta \;=\; f_c \ln|O|. \tag{4}$$

*Proof.* Subtract (2) for the two models. All terms with $r \neq c$ cancel by assumption, leaving $f_c(L_c^0 - L_c^\star)$. Substituting the uniform bound and exact prediction gives (4). □

**Corollary 1** (Relative invisibility)**.** *The contribution of $C$ to the validation loss is $\Delta/L = f_c \ln|O|/L$. It is small when $f_c$ is small, regardless of how much the capability matters for downstream use.*

Equation (4) makes a preregisterable prediction: $f_c$ comes from the evaluation set and $\ln|O|$ from the task, and we vary them independently.

### 3.1 Leakage: when the role partition is not clean

Proposition 1 assumes that the capability changes role $c$ but not the others. This assumption is violated in our data (Appendix B), so we report the counterfactual gap and then separate its role-specific signature from leakage. Let $\theta^-$ and $\theta^+$ be two models that differ only in whether they include $C$, and $\Delta = L(\theta^-) - L(\theta^+)$ the gap. Splitting (2) into the capability role and its complement $R = \sum_{r \neq c} f_r L_r$ gives

$$\Delta \;=\; \underbrace{f_c\,\Delta L_c}_{\text{signature}} \;+\; \underbrace{\textstyle\sum_{r\neq c} f_r \Delta L_r}_{\text{leakage } \Lambda}, \tag{5}$$

so the identity of Proposition 1 becomes the inequality

$$\Delta \;\leq\; f_c \ln|O| \;+\; \Lambda, \tag{6}$$

with equality when the role is driven to the uniform bound. The signature is a lower bound on the observable gap precisely when $\Lambda \geq 0$, and can *over*state it when $\Lambda < 0$. Section 6.3 gives an example of the second case: a capability whose signature alone would be worth $2.5\%$ of validation loss is hidden by a compensating gain elsewhere ($\Lambda = -0.1574$ nats).

Leakage also affects the natural population-level estimator, which regresses total loss on the capability role's loss across a population of runs.

**Proposition 2** (Slope inflation under leakage)**.** *Let $L = f_c L_{\text{ctx}} + R$ over a population of models. The ordinary least-squares slope of $L$ on $L_{\text{ctx}}$ is exactly*

$$\beta \;=\; f_c \;+\; \lambda_c, \qquad \lambda_c \;=\; \frac{\operatorname{Cov}(R, L_{\text{ctx}})}{\operatorname{Var}(L_{\text{ctx}})}, \tag{7}$$

*where $\lambda_c$ is the leakage slope: the rate at which the rest of the loss moves with the capability role across that population.*

*Proof.* $\operatorname{Cov}(L, L_{\text{ctx}}) = \operatorname{Cov}(f_c L_{\text{ctx}} + R, L_{\text{ctx}}) = f_c \operatorname{Var}(L_{\text{ctx}}) + \operatorname{Cov}(R, L_{\text{ctx}})$. Divide by $\operatorname{Var}(L_{\text{ctx}})$. □

**Corollary 2** (When the strict test is valid)**.** *The regression recovers $f_c$ if and only if $\lambda_c = 0$, i.e. only if the capability role is uncorrelated with the remaining loss across the population being regressed. The multiplicative inflation is $\beta/f_c = 1 + \lambda_c/f_c$.*

Equation (7) is exact. Appendix B shows that $\lambda_c$ accounts for the observed slope inflation, while Section 6 measures $\Lambda$ directly under a designed intervention.

**Interpretation.** Loss is therefore a *frequency-weighted* instrument: it naturally reflects capabilities exercised often, but can barely move for capabilities exercised on few tokens. Multi-step reasoning, rare factual lookup, and safety-relevant refusals can all fall into this regime.

## 4 A Dual-Route Sandbox

To test (4), we need a capability that can be cleanly present or absent, so we build a synthetic corpus with two routes that are separable by construction.

**Corpus.** The $1024$-token vocabulary contains $256$ *global triggers*, $128$ *context triggers*, an output pool $O$, and Zipf fillers in $256$-token sequences. Global triggers have fixed successors under $G$ (parametric memorization). Context triggers choose successors uniformly from $O$ and repeat them within a sequence (in-context retrieval). Both routes use the same cross-entropy, allowing capacity to shift between them at similar total loss. Role $c$ is the successor of a context trigger on its second or later occurrence, with share $f_{\text{ctx}}$.

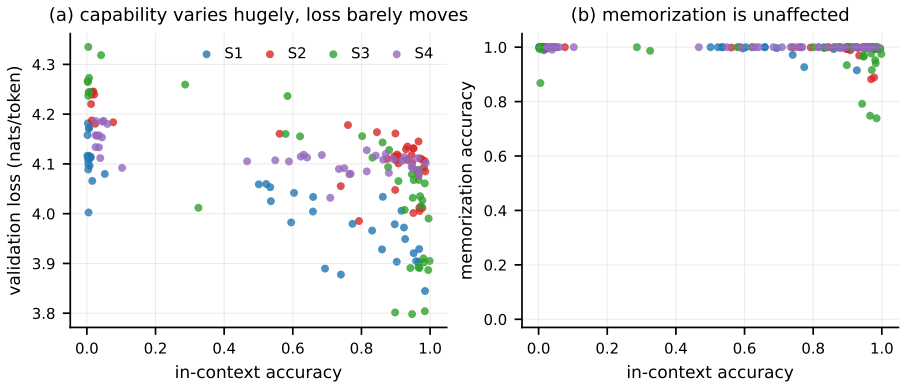



Figure 1: Across 160 runs, in-context accuracy spans nearly the whole unit interval while validation loss moves within a narrow band (a) and memorization is essentially unaffected (b). Colour denotes sandbox variant.

**Probes.** We probe memorization on single-occurrence global triggers and in-context retrieval on repeated context triggers; first occurrences serve as chance-level controls at $1/|O|$. Conflict probes pair a global successor with a contradictory in-context successor, and $\log p(\text{context}) - \log p(\text{memory})$ measures route preference.

**Variants.** Four variants vary the two factors of Eq. (4) independently (Table 1): S2 changes $|O|$, S3 changes token share, and S4 reduces both. The predicted signatures span $6.4\times$.

**Models and training.** We train $\sim\sim 10^5$-parameter decoder-only Transformers (Vaswani et al., 2017) with RMSNorm (Zhang & Sennrich, 2019), RoPE (Su et al., 2024), tied embeddings, and AdamW (Loshchilov & Hutter, 2019) for 3000 steps. We record loss trajectories, probe scores, and the exact role decomposition; iso-loss comparisons use interpolation. The decomposition matches total loss to $2.0 \times 10^{-7}$ (Appendix A).

## 5 Synthetic Results

### 5.1 The two routes dissociate

Across 160 runs, memorization accuracy is $0.9886$ with standard deviation $0.0382$ (149/160 runs above $0.95$), while in-context accuracy has standard deviation $0.3922$ and spans $[0.001, 0.999]$, a spread $10.3\times$ larger (Fig. 1). Architecture family, data distribution, and token budget are fixed; only width, heads, learning rate and seed vary. The outcome is close to binary: only 4 of 160 runs ($2.5\%$) land between accuracy $0.10$ and $0.50$ (Fig. 4, Appendix A). We therefore treat "route present" and "route absent" as two populations. That a mechanism forms at all can depend on seed at fixed scale, consistent with earlier observations on the variability of capability emergence. We use this only as a source of matched pairs that differ in one capability.

### 5.2 The premise holds and the law predicts the loss gap

Proposition 1 assumes that without the capability the role is unpredictable beyond a uniform marginal. Runs without the route pay $5.441$ nats on in-context tokens in S1 against a bound of $\ln|O| = 5.545$; $4.133$ against $4.159$ in S2; $5.420$ against $5.545$ in S3; and $3.460$ against $3.466$ in S4 (Fig. 2b). The largest deviation from the bound across all four variants is 2.3%, and individual failed runs land within $0.005$ nats of it. Runs with the route pay $0.922$–$0.963$ on the same tokens. The span of validation loss attributable to the capability, $f_{\text{ctx}}(\max L_{\text{ctx}} - \min L_{\text{ctx}})$, therefore matches the ceiling $f_{\text{ctx}} \ln|O|$ closely in every variant: $0.1299$ against $0.1294$ (S1), $0.0972$ against $0.0970$ (S2), $0.2604$ against $0.2610$ (S3), and $0.0403$ against $0.0407$ (S4).

Across route-present and route-absent populations, predicted and observed gaps agree within $0.99$–$1.45\times$ and are nearly rank-ordered ($r^2 = 0.9970$; Fig. 2a). This remains true while $f_{\text{ctx}}$ and $\ln|O|$

Table 1: Sandbox variants and the law tested by group comparison. Predicted $\Delta$ is $f_{\text{ctx}} \ln |O|$ from Eq. (4), fixed before any model is trained. Observed is the mean validation-loss difference between route-absent and route-present runs, with bootstrap 95% interval.

| Variant | $\lvert O\rvert$ | $f_{\text{ctx}}$ | $\ln \lvert O\rvert$ | pred. $\Delta$ | obs. $\Delta$ | 95% CI | obs/pred | % of $L$ |
|---|---|---|---|---|---|---|---|---|
| S1 | 256 | 0.02333 | 5.545 | 0.1294 | 0.1436 | [0.1110, 0.1762] | 1.110 | 3.49% |
| S2 | 64 | 0.02333 | 4.159 | 0.0970 | 0.1202 | [0.0953, 0.1458] | 1.238 | 2.85% |
| S3 | 256 | 0.04707 | 5.545 | 0.2610 | 0.2595 | [0.2121, 0.3083] | 0.994 | 6.07% |
| S4 | 32 | 0.01174 | 3.466 | 0.0407 | 0.0588 | [0.0429, 0.0746] | 1.445 | 1.41% |

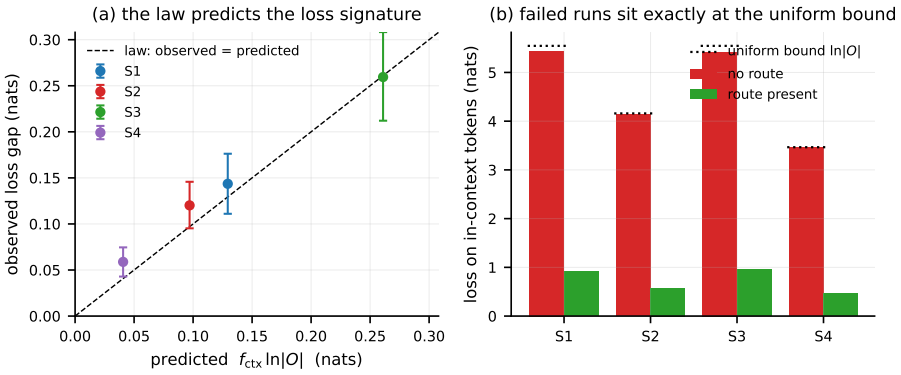



Figure 2: (a) Predicted signature $f_{\text{ctx}} \ln |O|$ against the observed loss gap between route-present and route-absent runs; the dashed line is equality and bars are bootstrap 95% intervals. (b) Runs without the route sit on the uniform bound $\ln |O|$ (dotted).

vary independently across $6.4\times$. The small excess over prediction is explained by correlated deficits in other roles.

### 5.3 Leakage in the sweep

Regressing total loss on in-context role loss gives slopes $1.422$–$1.705\times$ the analytic share, as expected when other roles move. The measured leakage slope $\lambda_c$ explains the observed inflation (Tables 7–8). Here $\lambda_c > 0$ because the low learning rates that produce route-absent runs also undertrain other roles. Thus Eq. (4) predicts the capability component, not the full population gap; the measured capability share is $60$–$81\%$ (Fig. 6).

### 5.4 Loss matching: the claim without any modelling

Loss matching avoids the objection that route-absent models are simply worse overall: we compare pairs within $0.01$ nats of validation loss, without regression or grouping assumptions. Across $336$ pairs, in-context accuracy differs by up to $0.948$ (Table 2, Fig. 7).

We also match memorization within $0.005$. In $S1$, one pair has validation losses only $0.0035$ nats apart and matched memorization ($1.000$ vs. $1.000$), yet in-context accuracy differs by $188\times$ ($0.917$ vs. $0.005$). Only one checkpoint expresses the capability.

Applying this to public PolyPythia 410M checkpoints (Appendix D) reproduces the effect at scale, though more weakly: among 33 pairs matched within 0.01 nats, LAMBADA accuracy differs by up to 2.66 points (a $20\times$ amplification). All these stable runs have the capability, so loss matching exposes only a difference in degree, as expected for a capability with a large token share.

### 5.5 Consequences

**Loss curves can hide capability.** Route-present and route-absent loss trajectories are nearly indistinguishable even as their capability trajectories separate (Fig. 5, Appendix A).

Table 2: Loss-matched pairs. For each variant, the pair with the largest capability gap among runs whose validation loss agrees to within 0.01 nats. Validation loss is effectively identical within each row; in-context accuracy is not.

| | matched | validation loss (nats) | | | in-context acc. | | memorization acc. | |
|---|---|---|---|---|---|---|---|---|
| Variant | pairs | run $a$ | run $b$ | $\lvert\Delta L\rvert$ | run $a$ | run $b$ | run $a$ | run $b$ |
| S1 | 46 | 4.0058 | 4.0024 | 0.0035 | 0.917 | 0.005 | 1.000 | 1.000 |
| S2 | 85 | 4.1874 | 4.1780 | 0.0094 | 0.015 | 0.761 | 0.997 | 1.000 |
| S3 | 41 | 4.0151 | 4.0118 | 0.0033 | 0.974 | 0.325 | 1.000 | 0.987 |
| S4 | 164 | 4.1118 | 4.1020 | 0.0097 | 0.039 | 0.987 | 0.991 | 1.000 |

Table 3: Iso-loss comparison at 76M parameters. Both conditions are evaluated at the same validation loss ($\approx 6.49$); the capability gap is read from probe accuracy alone.

| Condition ($w_c$) | Val. loss | ICL acc. | Mem. acc. |
|---|---|---|---|
| Intervened (4.0) | $\approx 6.49$ | 77.3% | 28.9% |
| Control (0.0) | $\approx 6.49$ | 7.8% | 83.8% |
| $\lvert\Delta\rvert$ | 0.0027 | +69.5 pts | −54.9 pts |

**Detectability.** Aggregate loss gives AUC $0.973$–$1.000$, whereas role-resolved loss yields $d = 5.96$–$13.09$ and AUC $= 1.000$ (Table 6). Role partitioning therefore makes capability detection much sharper. Reweighting the capability role by $\alpha^\star = 1/f_c$ further raises AUC from $0.973$–$1.000$ to $1.000$ and improves discrimination of matched pairs by $154$–$519\times$.

## 6 Causal Intervention at Scale: A 76M-Parameter Replication

The sandbox could be criticized as a low-capacity artifact: perhaps a small model simply cannot hold both routes. We therefore repeat the experiment at $76 \times 10^6$ parameters using a designed causal lever, measure leakage directly, and sweep intervention strength.

### 6.1 Architecture, task, and intervention

We use a 6-layer, $76 \times 10^6$-parameter Transformer ($d_{\text{model}} = 1024$, 16 heads) on the dual-route task. Constraining the output space to $|O| = 16$ fixes the capability's maximum information content at $\ln 16 \approx 2.773$ nats (Eq. (4)) and gives well-defined induction-head phase transitions (Olsson et al., 2022; Bietti et al., 2023). The corpus generator fixes the analytic token share at $f_c = 0.0931$. We intervene on the optimization signal by scaling the backward-pass loss on context-trigger tokens (role $c$) by a weight $w_c$, leaving the architecture, data, and all other tokens (unit weight) unchanged. The *intervened* condition ($w_c = 4.0$) quadruples the induction gradient, and the *control* ($w_c = 0.0$) zeroes it. Because $w_c$ modifies only the backward pass, both conditions are scored by the uniformly weighted validation loss of Eq. (2).

### 6.2 Iso-loss comparison

We take checkpoints from each condition where their trajectories reach the same loss, $L \approx 6.49$ (step 650 for the intervened model, step 950 for the control), instead of matching step count, so that the comparison isolates what the intervention changed at fixed overall predictive power (Table 3).

At matched validation loss, the intervened model shows robust in-context retrieval: $77.3\%$ in-context accuracy against $28.9\%$ memorization accuracy. The control has the reverse allocation, $7.8\%$ against $83.8\%$, and relies on rote memorization of the constrained output space instead of routing through context. The two conditions differ by $69.5$ points of in-context accuracy while their validation losses differ by $0.0027$ nats ($0.04\%$ of the operating point). This is the pattern of Table 2, obtained at $760\times$ the parameter count and by a designed intervention instead of selection.

Table 4: Capability divergence across gradient intervention weights $w_c$ at the iso-loss operating point. $\Delta L_c$ and $\Lambda$ are measured against the $w_c = 0.0$ control; the last column is the fraction of the capability term absorbed by leakage.

| $w_c$ | val. loss | ICL acc. | mem. acc. | $\Delta L_c$ | $\Lambda$ | $-\Lambda/(f_c \Delta L_c)$ |
|---|---|---|---|---|---|---|
| 0.0 | 6.4988 | 7.8% | 83.8% | 0.0000 | 0.0000 | – |
| 1.0 | 6.4877 | 16.0% | 77.5% | 0.2660 | -0.0187 | 0.75 |
| 2.0 | 6.4851 | 51.2% | 60.5% | 1.0890 | -0.0927 | 0.91 |
| 3.0 | 6.4862 | 70.7% | 39.5% | 1.5225 | -0.1341 | 0.95 |
| 4.0 | 6.4911 | 77.3% | 28.9% | 1.7194 | -0.1574 | 0.98 |

### 6.3 Direct measurement of capability leakage

A near-zero total gap can arise either because little changed or because two opposing shifts cancel. Eq. (5) separates the two:

$$\Delta L_{\text{total}} \;=\; f_c \, \Delta L_c \;+\; \Lambda. \tag{8}$$

We measure the per-role loss directly at the two iso-loss checkpoints, where $f_c = 0.0931$. The role gap is $\Delta L_c = 1.7194$ nats, giving a capability term $f_c \Delta L_c = 0.1601$ nats (2.5% of operating loss) and leakage $\Lambda = -0.1574$ nats.

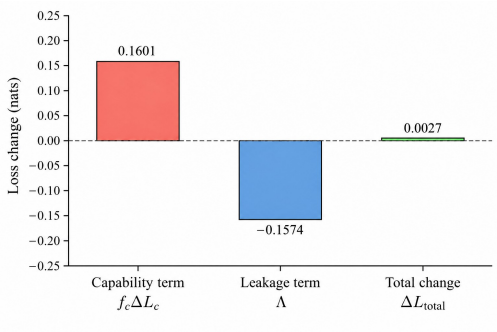



Figure 3: Exact decomposition of the iso-loss gap at 76M parameters, Eq. (8). The capability term $f_c \, \Delta L_c = 0.1601$ nats is offset almost entirely by negative leakage $\Lambda = -0.1574$ nats on the remaining roles, leaving a net $\Delta L_{\text{total}} = 0.0027$ nats.

The leakage is large and negative, and it cancels the capability term to within 0.0027 nats (Fig. 3). A capability term of 2.5% of the operating loss would on its own be clearly visible by the standards of Section 7. It is hidden here because the control, with no gradient pressure to route through context, memorizes the constrained output space *better* (83.8% against 28.9%). That gain, together with the calibration effect on first-occurrence tokens noted in Section 5.3, is worth $\Lambda = -0.1574$ nats. This is the $\Lambda < 0$ case of Eq. (6), where the signature *overstates* the observable gap. The compensating movement sits on the competing memorization route and is not a diffuse loss of capacity, so it is a targeted capability trade and not general degradation. The near-equality of the two terms is also what lets iso-loss selection succeed at all. Direct measurement adds the size of the exchange, which the aggregate gap conceals by two orders of magnitude.

### 6.4 Ablation over intervention weight

To rule out an endpoint effect, we repeat the intervention at $w_c \in \{0, 1, 2, 3, 4\}$, select checkpoints at the common iso-loss point, and recompute Eq. (8) relative to the $w_c = 0.0$ control (Appendix E).

As $w_c$ rises from 1.0 to 4.0, the role gap $\Delta L_c$ increases monotonically from 0.2660 to 1.7194 nats and in-context accuracy from 16.0% to 77.3%, while memorization accuracy falls from 77.5% to 28.9%. There is no discontinuity: the reallocation from memorization to routing varies continuously with how strongly the induction role is weighted in the gradient. Leakage becomes more negative, from -0.0187 to -0.1574, and the fraction of the capability term it absorbs rises from 0.75 to 0.98.

Table 5: Induction's loss signature on WikiText-103 validation ($184,140$ scored tokens, $40,425$ within-type pairs, share $f_{\mathrm{ctx}} = 0.2195$). Gain is position-adjusted.

| Model | params | total $L$ | gain | $\Delta$ | % of $L$ |
|---|---|---|---|---|---|
| GPT-2 | 124 M | 3.395 | 2.105 | 0.462 | 13.6 |
| GPT-2 med. | 355 M | 3.087 | 1.940 | 0.426 | 13.8 |
| GPT-2 large | 774 M | 2.948 | 1.936 | 0.425 | 14.4 |

At every weight the network does the same thing to different degrees: it gives up rote memorization capacity to build in-context routing, and the two movements, scored by the same loss, nearly cancel. This seesaw underlies every loss-matched pair in this paper, shown here explicitly and as a function of dose.

**Interpretation.** The capability gap is therefore not confined to the smallest models. At $76 \times 10^6$ parameters, a designed intervention reproduces it through a smooth trade with memorization. Here $f_c \approx 0.0931$ is not especially small, so the signature itself would be visible; the competing route absorbs it. The $69.5$-point swing is only $0.04\%$ of operating validation loss, yet remains clear under the capability probe.

## 7 Natural Text

The synthetic setting calibrates the law; whether its factors are small in real text is an empirical question.

**Method.** On WikiText-103 (Merity et al., 2017), we compare first and later occurrences of bigrams within $1024$-token windows. Estimating gain within bigram type controls for identity and frequency; we subtract the generic positional advantage from non-repeated tokens. We evaluate GPT-2 base, medium, and large (Radford et al., 2019), which share a tokenizer and thus the same token share.

**Induction is not invisible.** Since $22.0\%$ of WikiText-103 tokens are induction-completable with a position-adjusted gain near $2.105$ nats, the signature reaches $0.425$–$0.462$ nats ($13.6$–$14.4\%$ of total loss; Table 5, Appendix F). Induction is clearly visible here because Corollary 1 depends on a small token share, and this share is large. The absolute signature stays nearly flat across scale ($0.462 \to 0.425$ nats) while total loss falls ($3.395 \to 2.948$), so induction's *relative* share of loss *rises* with scale.

**Separating factors and unsupervised roles.** Stratifying by distance to the matching bigram keeps the per-token gain nearly constant while token share varies $3.1\times$, and the signature follows that share (Table 10). An Otsu threshold on context-ablated loss differences (Otsu, 1979) recovers a similar unsupervised role signature ($18.8$–$22.4\%$ of validation loss) with $1.98$–$2.16\times$ enrichment (Appendix F.1). This is useful for discovery, but not a substitute for a predefined test.

## 8 Implications

**Compute the signature before trusting the loss, and check for a competing route.** Both factors of Eq. (4) can be estimated before training. If their product is small, aggregate loss is a weak instrument. Substitute routes can hide larger capability exchanges, so leakage (Eq. (5)) should be measured rather than assumed away.

**Report role-resolved loss; loss-matched is not capability-matched.** Partitioning validation loss by token role is cheap and greatly improves capability detection (AUC $0.973$ to $\geq 1.000$). Loss-matched checkpoints are not capability-matched: performance gaps can span the capability's entire dynamic range, including under our 76 M-parameter intervention. For bimodal outcomes ($2.5\%$ middle mass), distributions summarize better than means.

**How large must an evaluation set be?** Role-resolved loss needs $\Theta(1/f_c)$ tokens to distinguish models, versus $\Theta(1/f_c^2)$ for aggregate loss (Proposition 3). The estimate is optimistic by $\sim 1.21\times$ but tracks experiments across $579\times$ ($r = 0.9966$). The same logic explains why post-training behaviors such as instruction following may barely move pretraining loss when their $f_c$ is small (Appendix H).

## 9 LIMITATIONS

The clean-partition assumption behind Proposition 1 is violated empirically, so leakage must be modeled explicitly; in our scale intervention it is large and negative. The synthetic task deliberately supplies substitute routes and therefore does not establish leakage magnitudes for natural text or larger models. The public-checkpoint result is correlational ($r = -0.986$) and weaker ($20\times$) because LAMBADA has a large capability token share. More generally, the synthetic models establish the accounting framework, not the prevalence of any particular real-world capability. Natural-text measurements use proxy ablations, the power result is a design estimate, and the observed bimodality is task-specific. See Appendix J.

## 10 CONCLUSION

A capability's effect on validation loss is bounded by its token share times its information content, so whether loss can detect it is computable in advance. In a synthetic dual-route task, the predicted signature separates parametric memorization from in-context retrieval and accounts for 60–81% of the measured loss gap. Models matched in validation loss (differing by only $0.09\%$) can diverge completely in capability. At 76M parameters, a causal intervention induces a $+69.5$-point in-context capability gap at $\Delta L = 0.0027$ nats, and the capability's signature (0.1601 nats) is offset by a compensating gain in the competing route ($\Lambda = -0.1574$ nats). On natural text, induction shows up in aggregate loss because its token share is large. Validation loss measures frequent capabilities without substitutes reliably. Rare capabilities, and those with competing parametric routes, need capability-specific evaluation, and whether a given capability falls in that class can be determined before running experiments.

### ETHICS STATEMENT

This work uses synthetic data generated by our own corpus generator, publicly released model checkpoints (GPT-2, Pythia, PolyPythias), and the public WikiText-103 and LAMBADA datasets. No human subjects, personal data, or proprietary data are involved. The framework is a measurement tool for model evaluation; we do not foresee direct harmful applications, though we note that the same accounting shows that safety-relevant behaviours with small token share are exactly those aggregate loss is least able to audit, which argues for capability-specific evaluation of such behaviours.

### REPRODUCIBILITY STATEMENT

All experiments run on a single NVIDIA RTX 5050 laptop GPU (8 GB). The full synthetic sweep of 160 runs takes under two hours; the natural-text measurement is inference-only; the 76M-parameter causal intervention and its five-weight ablation (Section 6) were trained under the same budget. Every number in this paper is emitted programmatically from the result files into a macro file that the manuscript includes, so no value is transcribed by hand. The corpus construction is specified in Section 4, the intervention in Section 6.1, the natural-text labelling in Section 7, the role-discovery procedure in Appendix F.1, and the real-checkpoint protocol in Appendix D. Proofs of all propositions are given in full in the main text and Appendix G. Source code and the macro-emission script will be released with the camera-ready version.

### AI USE STATEMENT

In this work, we used generative AI tools for grammatical editing, improving the clarity and readability of the manuscript, and assisting with LaTeX formatting of portions of the latest version of the document. We did not use generative AI tools to generate research ideas, design or conduct

experiments, perform experimental runs, generate numerical results, or carry out the research analyses. All research findings, experiments, long-running computations, numerical results, and analyses were conducted independently by the author. All AI-assisted text and formatting were reviewed and verified by the author. We take full responsibility for the final content of this work, including all research claims, analyses, numerical results, and artifacts.

## REFERENCES


Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.3, knowledge capacity scaling laws. *arXiv preprint arXiv:2404.05405*, 2024.

Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: SGD learns parities near the computational limit. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Stella Biderman et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, 2023.

Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Hervé Jegou, and Léon Bottou. Birth of a transformer: A memory viewpoint. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Tom B. Brown et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Nicholas Carlini et al. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

Yann Dubois et al. AlpacaFarm: A simulation framework for methods that learn from human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Nelson Elhage et al. Toy models of superposition. *Transformer Circuits Thread*, 2022.

Vitaly Feldman. Does learning require memorization? A short tale about a long tail. In *ACM Symposium on Theory of Computing (STOC)*, 2020.

Deep Ganguli et al. Predictability and surprise in large generative models. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2022.

Dan Hendrycks et al. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

Jordan Hoffmann et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Jared Kaplan et al. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Bill Yuchen Lin et al. The unlocking spell on base LLMs: Rethinking alignment via in-context learning. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2312.01552.

Nelson F. Liu et al. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations (ICLR)*, 2023.

Catherine Olsson et al. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

Long Ouyang et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Denis Paperno et al. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 (3):379–423, 1948.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Oskar van der Wal et al. PolyPythias: Stability and outliers across fifty language model pre-training runs. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2503.09543.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jason Wei et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Chunting Zhou et al. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

# Appendix

## A Additional Sandbox Details and Figures

**Implementation check.** The decomposition is numerically exact: Eq. (2) matches the reported total to $2.0 \times 10^{-7}$, and the role shares sum to $1.0000000000$.

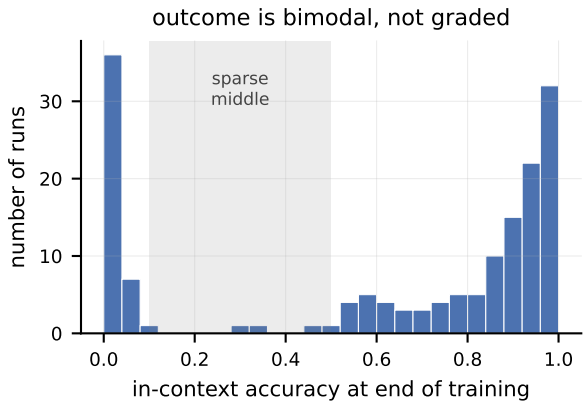



Figure 4: Final in-context accuracy is bimodal; the shaded band between $0.10$ and $0.50$ holds only $2.5\%$ of runs.

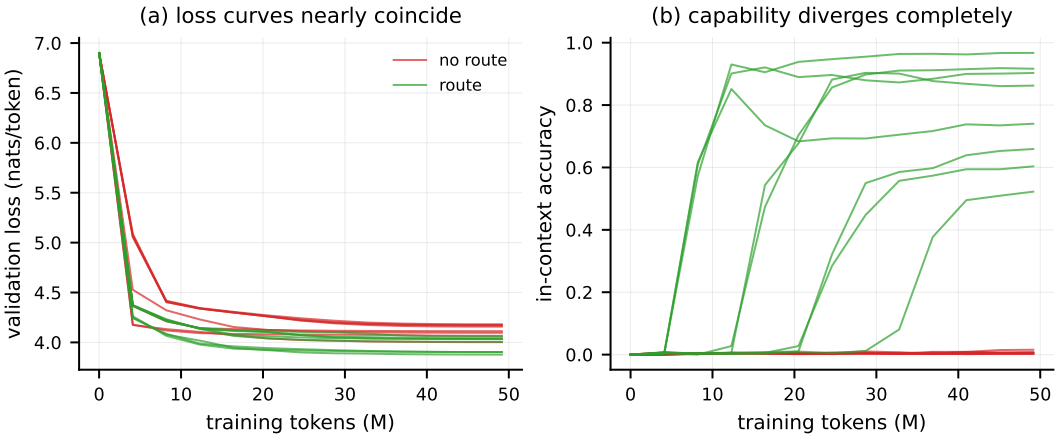



Figure 5: S1 runs. (a) Validation-loss trajectories of route-present (green) and route-absent (red) runs nearly coincide. (b) The same runs' in-context accuracy diverges completely.

**Where the gap comes from.** Because the shares are common to all runs, the observed gap decomposes exactly into per-role contributions with no residual (Fig. 6). The in-context role supplies 60–81% of the gap depending on variant. The remainder is dominated by the trigger role, which is where the correlated-deficit leakage of Appendix B appears. One contribution is consistently *negative*: on the unpredictable first-occurrence tokens, route-absent runs are measurably *better* (e.g. $-0.0145$ nats in S1). Those tokens are irreducibly uniform, so the only way to do better is to be better calibrated on them, which a model that has not learned to expect a retrievable answer will be.

**The conflict probe.** The conflict probe is out of distribution: global triggers never contradict $G$ during training, so the result in Section 5.5 reflects learned gating rather than a defect. Among route-present runs at in-context accuracy $0.860$, $0.6\%$ of conflict cases are answered from context and $97.5\%$ from memory (allocation index $-10.8$), indicating trigger-specific gating rather than general copying.

## B THE STRICT REGRESSION TEST AND ITS RESOLUTION

Because $L = f_{\text{ctx}} L_{\text{ctx}} + \sum_{r \neq c} f_r L_r$, regressing total validation loss on the in-context role loss across *all* runs of a variant should yield a slope equal to $f_{\text{ctx}}$ if the capability truly leaves other roles untouched. This is a sharp prediction with no free parameter. It does not hold: the fitted slopes are

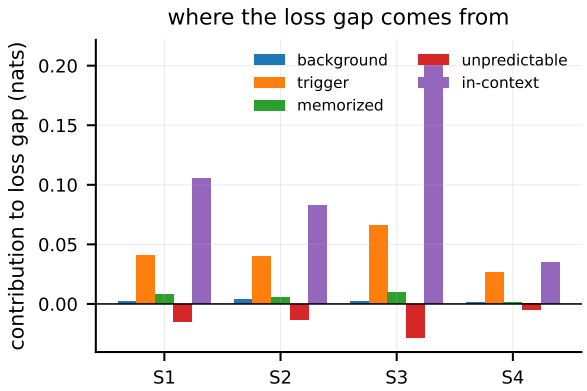



Figure 6: Exact per-role contributions to the observed loss gap. The in-context role dominates; negative bars on other roles show route-absent runs compensating elsewhere.

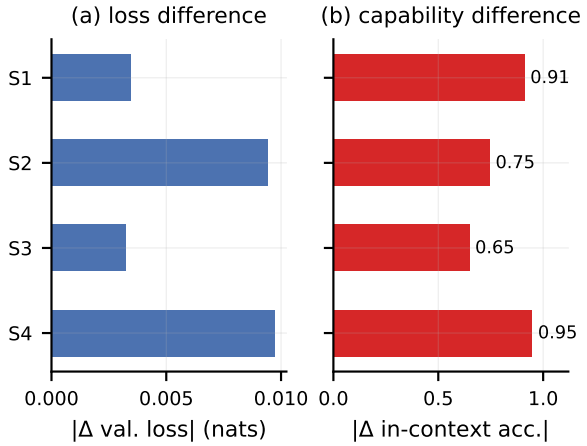



Figure 7: For each variant's most extreme loss-matched pair, the validation-loss difference (left, nats) against the in-context accuracy difference (right). The instrument barely moves while the capability changes completely.

1.422–1.705× the analytic share and the 95% intervals cover it in only 0 of 4 variants (Table 7), with the residual $L - f_{\text{ctx}} L_{\text{ctx}}$ retaining a positive slope in $L_{\text{ctx}}$.

Proposition 2 resolves this directly. Measuring $\lambda_c = \text{Cov}(R, L_{\text{ctx}}) / \text{Var}(L_{\text{ctx}})$ and adding it to $f_c$ reproduces the fitted slope in every variant (Table 8). Because Eq. (7) is an identity, the residual $|\beta - (f_c + \lambda_c)| = 5.2e-17$ is only a consistency check. The substantive result is the size of $\lambda_c$: $1 + \lambda_c / f_c$ matches the observed $\beta / f_c$, explaining the full 1.422–1.705× inflation. We induced this leakage by using low learning rates to enrich route-absent runs; those runs are also mildly undertrained elsewhere, so $\lambda_c > 0$ by construction.

## C Importance-Reweighted Loss

If a capability is diluted because its tokens are rare, reweight its role instead of averaging all tokens equally:

$$\widetilde{L}(\alpha) = \frac{\sum_r w_r f_r L_r}{\sum_r w_r f_r}, \qquad w_c = \alpha,\ w_{r \neq c} = 1. \tag{9}$$

Setting $\alpha = 1$ recovers ordinary loss. We use $\alpha^\star = 1/f_c$, which gives the capability role weight comparable to the rest of the distribution and depends only on $f_c$. In our variants, $\alpha^\star$ ranges from 21 to 85. This is a reweighted loss rather than a new metric: the importance is made explicit instead of inherited from corpus frequency. Section 6 applies the same idea to the training gradient.

Table 6: Detecting whether the in-context route is present, from aggregate validation loss versus role-resolved loss.

| | AUC | | Cohen's $d$ | |
|---|---|---|---|---|
| Variant | aggregate | role-res. | aggregate | role-res. |
| S1 | 0.977 | 1.000 | 2.53 | 6.56 |
| S2 | 1.000 | 1.000 | 2.56 | 9.96 |
| S3 | 1.000 | 1.000 | 2.40 | 5.96 |
| S4 | 0.973 | 1.000 | 2.77 | 13.09 |

Table 7: Stricter regression test. The fitted slope should equal the analytic token share $f_{\mathrm{ctx}}$ if the capability left all other roles untouched.

| Variant | runs $n$ | analytic $f_{\mathrm{ctx}}$ | fitted slope | slope 95% CI | slope / $f_{\mathrm{ctx}}$ | $R^2$ |
|---|---|---|---|---|---|---|
| S1 | 40 | 0.02333 | 0.03406 | [0.02823, 0.03990] | 1.460 | 0.775 |
| S2 | 40 | 0.02333 | 0.03694 | [0.02858, 0.04530] | 1.584 | 0.664 |
| S3 | 40 | 0.04707 | 0.06693 | [0.05525, 0.07861] | 1.422 | 0.768 |
| S4 | 40 | 0.01174 | 0.02002 | [0.01565, 0.02439] | 1.705 | 0.680 |

Applied to the same 160 runs, it repairs the ranking failures of aggregate loss (Table 9). Detection AUC rises from $0.973$–$1.000$ to $1.000$ in every variant, misranked pairs fall from as high as $2.7\%$ to $0.0\%$, and the effect size grows from at most $2.77$ to $5.61$–$12.78$. Fig. 8 shows the transition is not knife-edged: AUC saturates well before $\alpha^\star$, so the correction does not require $f_c$ to be known precisely. Restricting to loss-matched pairs that nonetheless differ by more than $0.5$ in in-context accuracy, the reweighted loss separates them by $154$–$519\times$ more than the raw loss does.

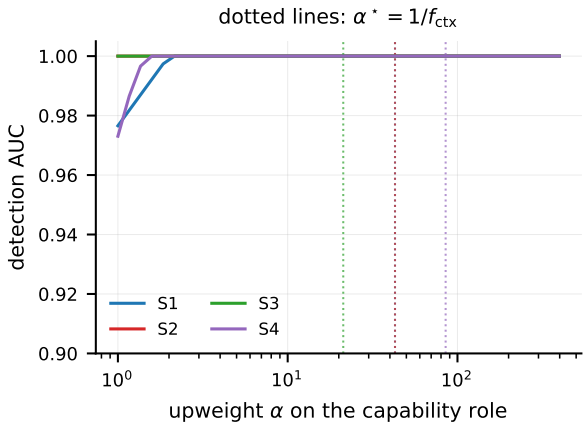



Figure 8: Detection AUC as the capability role is upweighted by $\alpha$ in Eq. (9). Dotted lines mark $\alpha^\star = 1/f_{\mathrm{ctx}}$. The correction saturates well before $\alpha^\star$, so it is insensitive to error in $f_c$.

## D Loss Matching on Real Pretrained Checkpoints

The main text relies on models trained for this study, so we repeat loss matching on public checkpoints that were not designed for this experiment.

**Why PolyPythias.** The test needs models that differ in as little as possible. PolyPythias (van der Wal et al., 2025) extends the Pythia suite (Biderman et al., 2023) with additional pretraining runs that differ from one another *only by random seed*: identical architecture, identical data, identical

Table 8: Proposition 2 on the sweep. The measured leakage slope $\lambda_c$ added to the analytic share reproduces the fitted regression slope and quantifies the inflation.

| Var. | $f_c$ | $\lambda_c$ | $f_c + \lambda_c$ | $\beta$ (fitted) | residual |
|---|---|---|---|---|---|
| S1 | 0.02333 | 0.01073 | 0.03406 | 0.03406 | 1.4e-17 |
| S2 | 0.02333 | 0.01361 | 0.03694 | 0.03694 | 2.8e-17 |
| S3 | 0.04707 | 0.01986 | 0.06693 | 0.06693 | 4.2e-17 |
| S4 | 0.01174 | 0.00828 | 0.02002 | 0.02002 | 5.2e-17 |
| *inflation* $\beta/f_c$*: observed vs. predicted* $1 + \lambda_c/f_c$ | | | | | |
| S1 | 1.460 vs. 1.460 | | S3 | 1.422 vs. 1.422 | |
| S2 | 1.584 vs. 1.584 | | S4 | 1.705 vs. 1.705 | |

Table 9: Ordinary versus importance-reweighted validation loss (Eq. (9), $\alpha = \alpha^\star = 1/f_c$) as a detector of the in-context route, on the same runs as Table 6.

| | | AUC | | Cohen's $d$ | | misrank |
|---|---|---|---|---|---|---|
| Var. | $\alpha^\star$ | raw | rewt. | raw | rewt. | raw→rewt. |
| S1 | 42.9 | 0.977 | 1.000 | 2.53 | 6.38 | 2.3%→0.0% |
| S2 | 42.9 | 1.000 | 1.000 | 2.56 | 9.41 | 0.0%→0.0% |
| S3 | 21.2 | 1.000 | 1.000 | 2.40 | 5.61 | 0.0%→0.0% |
| S4 | 85.2 | 0.973 | 1.000 | 2.77 | 12.78 | 2.7%→0.0% |

data order, dense checkpointing. We use all 9 public 410M seed runs at the two late checkpoints (steps 133000, 143000), giving 18 real models. Everything is inference-only and runs in 8 GB.

**Instrument and capabilities.** The instrument is mean next-token loss on a fixed held-out corpus (130,944 scored tokens). The auditable capability is LAMBADA accuracy (Paperno et al., 2016) on 5,153 examples, predicting the final word of a passage, which requires integrating long-range context and is scored exactly. We also recompute the induction-role accounting of Section 7 on the same text, where the role's share is $f_{\mathrm{ctx}} = 0.2222$.

**Result.** Loss and capability are strongly correlated across these checkpoints ($r = -0.986$), so at this scale loss is a genuinely informative instrument. 2 of 18 runs are outliers with markedly worse loss and capability (seed4—step133000, seed4—step143000), consistent with the instability PolyPythias reports; Fig. 9(a) shows them. Of the 33 pairs whose held-out loss agrees to within 0.01 nats, LAMBADA accuracy differs by up to $0.0266$ ($2.66$ points), with median $0.0064$ and 90th percentile $0.0210$. The extreme pair is two fully trained checkpoints differing only in seed:

| | seed8, step143000 | seed9, step143000 |
|---|---|---|
| held-out loss | 2.8923 | 2.9004 |
| LAMBADA accuracy | 0.5028 | 0.4762 |

Their loss differs by 0.0081 nats, or 0.28%; their capability differs by 0.0266 absolute, or 5.6% in relative terms. The capability moves roughly $20\times$ further than the instrument.

**How to read this honestly.** This is the same phenomenon as Section 5.4, at a real scale, but much weaker. The two settings differ in exactly the way the theory predicts: our synthetic capability is bimodal, so matched pairs can straddle the whole dynamic range, whereas these Pythia runs are stable and converged, all possess the capability, and seed variation only perturbs how well. Even among real checkpoints a practitioner would treat as interchangeable on loss, an auditable capability varies by several percent, and the variation is not predictable from the loss difference. We do not claim, on this evidence, that real pretraining produces loss-matched checkpoints differing in kind; establishing that would require capabilities with a much smaller token share than LAMBADA. We also could not use the Pile validation split as intended, so the held-out corpus is WikiText-103

validation, in-domain for Pythia but not verifiably unseen; only differences across checkpoints on one fixed corpus are used.

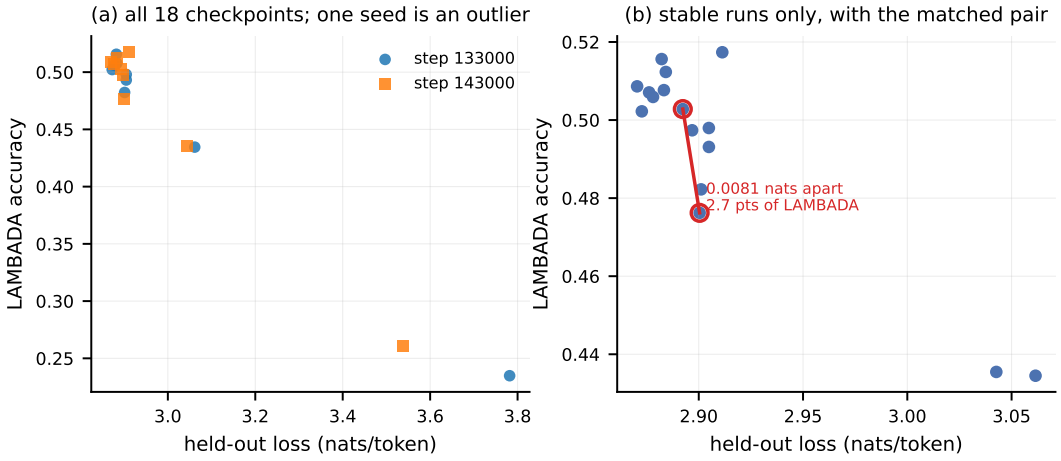



Figure 9: Real PolyPythias 410M checkpoints, 9 seeds at two late steps. (a) Held-out loss against LAMBADA accuracy; the outlier seed is visible at upper left. (b) The stable runs only, with the extreme loss-matched pair joined: the two checkpoints sit 0.0081 nats apart in loss and 2.66 points apart in capability.

## E Additional Results for the 76M-Parameter Intervention

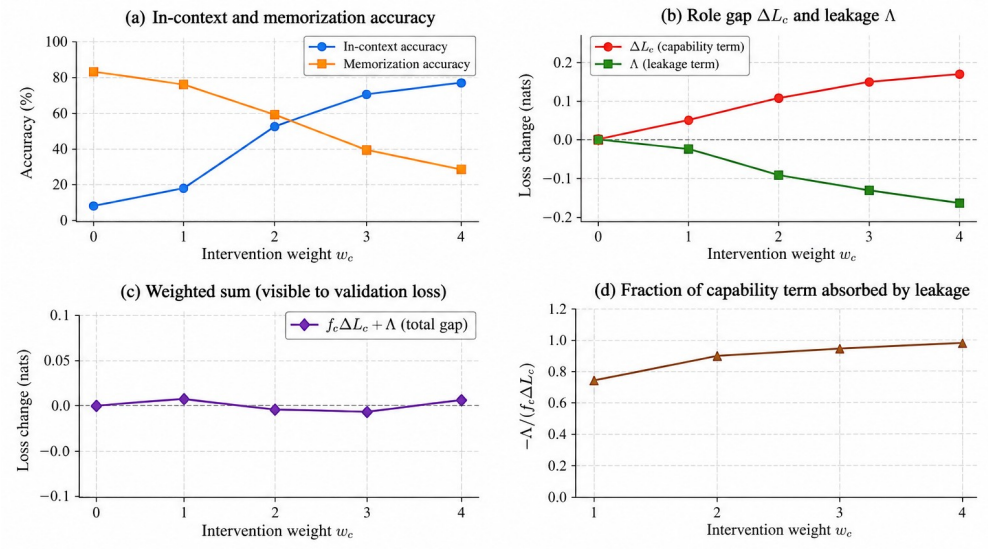



Figure 10: Capability divergence across gradient intervention weights $w_c$ at the iso-loss operating point. In-context accuracy rises monotonically from 7.8% to 77.3% while memorization accuracy falls from 83.8% to 28.9%; the role gap $\Delta L_c$ and the leakage $\Lambda$ grow in lockstep with opposite sign, so their weighted sum, the gap visible to validation loss, stays near zero at every weight.

# F NATURAL TEXT: ADDITIONAL TABLES AND FIGURES

Table 10: Induction on WikiText-103 stratified by distance to the matching bigram (GPT-2). Gain is nearly constant; the share and hence the signature are not.

| Distance | share $f$ | gain | $\Delta$ | % of $L$ |
|---|---|---|---|---|
| $< 64$ | 0.0379 | 2.30 | 0.0873 | 2.57 |
| 64–256 | 0.0632 | 1.99 | 0.1258 | 3.71 |
| $> 256$ | 0.1185 | 2.10 | 0.2491 | 7.34 |

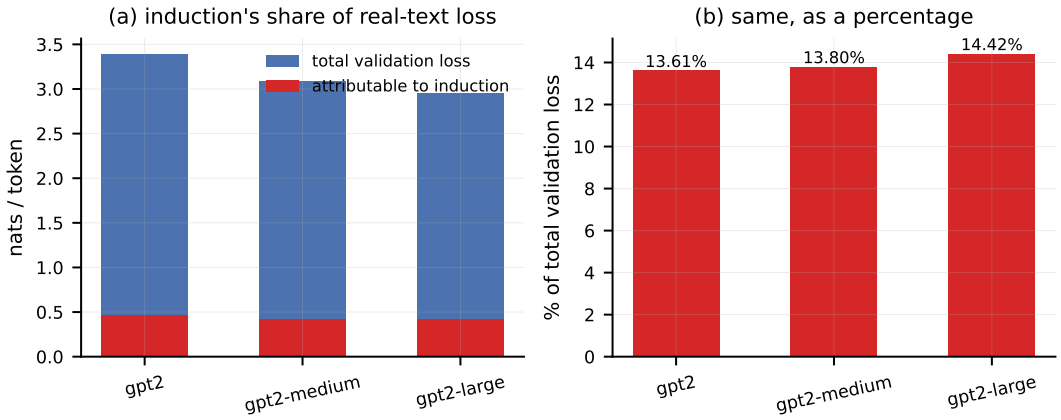



Figure 11: Induction's share of WikiText-103 validation loss for three GPT-2 sizes, in nats (a) and as a percentage (b).

## F.1 DISCOVERING ROLES WITHOUT LABELS

**Method.** Score every token twice: once with the full 1024-token window, and once with the window cut into independent segments of length 32, which destroys all long-range information while leaving local syntax intact. The per-token context dependency is

$$\Delta_t \;=\; \mathrm{nll}_{\mathrm{local}}(t) - \mathrm{nll}_{\mathrm{full}}(t). \tag{10}$$

Tokens are split into high- and low-dependency groups by a threshold chosen from the distribution of $\Delta$ alone using Otsu's criterion (Otsu, 1979); no label is consulted. The high group is the *discovered role*, and we rerun the accounting of Section 7 on it. Only tokens at within-segment offset $\geq 8$ are scored, so $\Delta$ measures long-range dependence rather than start-of-sequence effects.

**Result.** The procedure finds a role of share 0.1156–0.1189 against the hand-labelled 0.2195, with a per-token context gain of 5.38–5.71 nats, far larger than the hand-labelled role's 2.105, because thresholding selects the most context-dependent tokens rather than all induction-completable ones. The resulting signature is 18.8–22.4% of validation loss (Table 11, Fig. 12). It does not recover the same *set* of tokens: precision against the hand label is 0.44–0.48 and recall 0.30–0.34, an enrichment of 1.98–2.16× over the base rate 0.220. The discovered role is best understood as a broader "high long-range dependency" set that contains induction along with other long-range phenomena. Segment length and threshold rule both influence which role emerges.

# G DETECTION SAMPLE SIZE

**Proposition 3** (Detection sample size, normal approximation)**.** *Let per-token loss on role $c$ have variance $\sigma_c^2$, and let the capability shift that role's mean loss by $\delta$. Under a normal approximation*

Table 11: Unsupervised role discovery on WikiText-103 validation. Share, signature and overlap with the hand-labelled induction role.

| Model | thr. | $f_{\text{disc}}$ | gain | % of $L$ | prec. | rec. |
|---|---|---|---|---|---|---|
| GPT-2 | 2.877 | 0.1189 | 5.377 | 18.83 | 0.476 | 0.342 |
| GPT-2 med. | 2.921 | 0.1176 | 5.483 | 20.89 | 0.442 | 0.315 |
| GPT-2 large | 3.056 | 0.1156 | 5.711 | 22.40 | 0.436 | 0.305 |

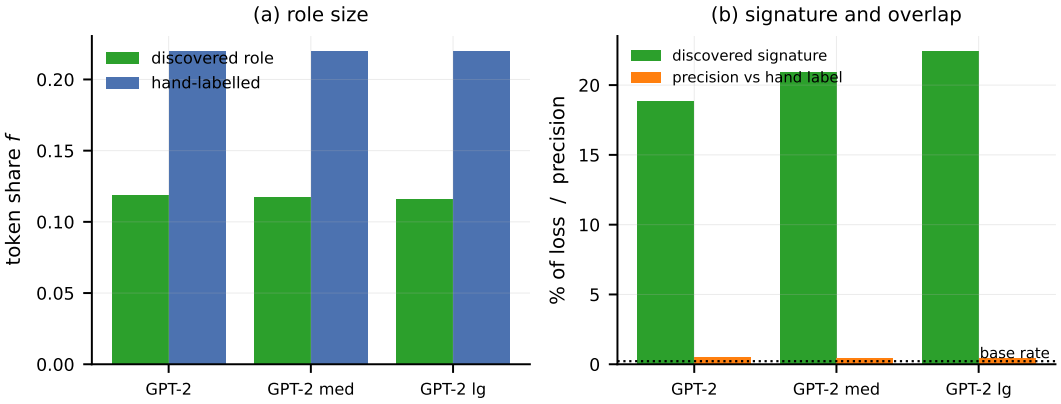



Figure 12: Unsupervised role discovery. (a) The discovered role is smaller than the hand-labelled induction role. (b) Its loss signature is comparable, and its precision against the hand label sits well above the base rate (dotted).

*to the sampling distribution of the mean per-token loss, distinguishing two models with a two-sample test at level $\alpha$ and power $1-\beta$ requires approximately*

$$N_c \gtrsim \frac{2\,(z_{1-\alpha/2}+z_{1-\beta})^2\,\sigma_c^2}{\delta^2} \tag{11}$$

*role-$c$ tokens, hence a corpus of $N_c/f_c$ tokens. If only* aggregate *loss is available, the capability shifts the aggregate mean by $f_c\delta$ against per-token noise $\sigma_{\text{tot}}$, so*

$$N \gtrsim \frac{2\,(z_{1-\alpha/2}+z_{1-\beta})^2\,\sigma_{\text{tot}}^2}{(f_c\,\delta)^2}. \tag{12}$$

*Proof.* Both are the standard two-sample normal-approximation requirements for detecting a mean difference, applied to the role-$c$ loss and to the total loss respectively; in the latter the decomposition (2) contributes the capability only through $f_c\delta$. □

**Corollary 3** (The cost of averaging)**.** *Under the same approximation, role-resolved evaluation needs $\Theta(1/f_c)$ tokens and aggregate loss needs $\Theta(1/f_c^2)$. Their ratio is $(\sigma_{\text{tot}}^2/\sigma_c^2)/f_c$, which diverges as the capability becomes rarer.*

**What the approximation does and does not promise.** Per-token cross-entropy is skewed and heavy-tailed, so the central-limit approximation behind Eqs. (11)–(12) is only asymptotic and is expected to be optimistic at the sample sizes where it matters most. The formulas are design estimates for planning an evaluation, not exact guarantees; the scaling statement of Corollary 3 is the robust content, since the $1/f_c$ versus $1/f_c^2$ dependence does not rely on the distributional form.

**Validation.** A single extreme pair would not test (11): at $\delta \approx \ln|O|$ the requirement falls below one token and any formula "passes." We therefore trained a ladder of 8 models spanning the capability range, collected per-token losses, and measured by Monte-Carlo resampling the sample size

at which a two-sample test first reaches 80% power. Across the 12 pairs whose requirement is measurable, predictions span $579\times$ in magnitude and the ratio of empirical to predicted $N_c$ has median 1.21 and geometric mean 1.21 (range 0.90–1.50), with log–log correlation $r = 0.9966$ (Fig. 13). A practitioner using Eq. (11) should inflate the result by a safety factor of this order. For the extreme pair (in-context accuracy 0.003 vs 0.975, $\delta = 5.004$ nats), Eq. (11) asks for 0.04 role tokens, i.e. a corpus of 2, whereas Eq. (12) asks for 6,307 tokens of aggregate loss, $3270\times$ more, matching the $(\sigma^2_{\text{tot}}/\sigma^2_c)/f_c = 3270\times$ of Corollary 3.

**How to use it.** Before evaluation, estimate $f_c$, the capability shift $\delta$ (at most $\ln|O|$), and $\sigma^2_c$. Equation (11), with the empirical safety factor, gives the required number of role tokens. With fewer tokens, the experiment may fail to resolve the capability and produce a null result for the instrument rather than the model.

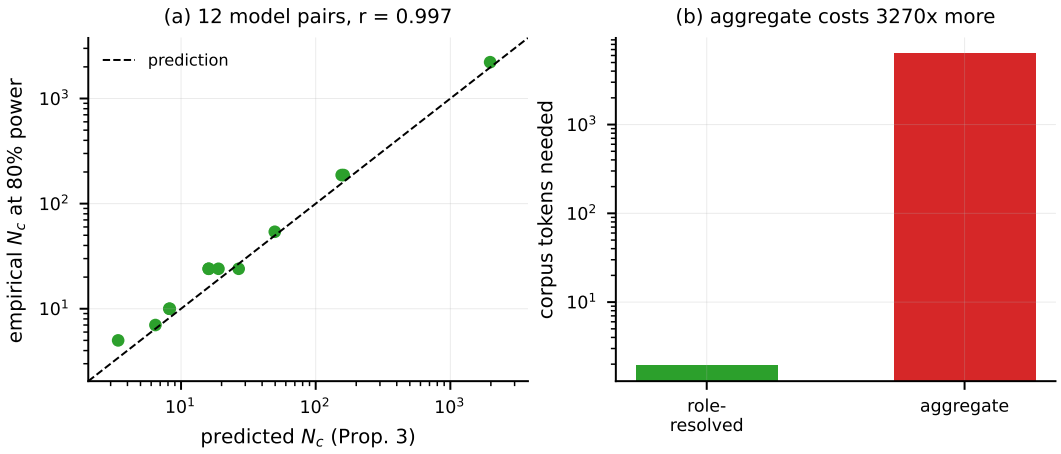



Figure 13: Validating Proposition 3. (a) Predicted against empirically measured $N_c$ at 80% power over 12 model pairs spanning $579\times$; dashed line is the prediction, which is systematically optimistic by a factor $\approx 1.21$. (b) Power as a function of tokens per group, role-resolved versus aggregate loss.

## H Why Post-Training Barely Moves Validation Loss

Instruction tuning and preference optimization (Ouyang et al., 2022; Rafailov et al., 2023) can change behaviour substantially while barely moving pretraining-style validation loss, motivating checkpoint selection by win rate or human preference rather than held-out loss (Ouyang et al., 2022; Dubois et al., 2023). Under Corollary 1, this is expected: instruction-following and refusal affect only a small fraction of positions in generic text, so with tiny $f_c$, Eq. (4) bounds their aggregate-loss effect even when behaviour changes substantially.

Published measurements support the premise quantitatively. Lin et al. (2024) compare base and alignment-tuned models token by token and find they agree on the top-ranked token at the majority of positions, with the shifts concentrated on stylistic tokens such as discourse markers and safety disclaimers; this is the "superficial alignment" picture argued from the data side by LIMA (Zhou et al., 2023). In our terms, post-training is a large intervention on a role with a very small $f_c$, so Corollary 1 predicts its loss signature is negligible, and Proposition 3 predicts that no feasible amount of generic held-out text would let aggregate loss detect it. This reframes checkpoint selection by win rate from a pragmatic workaround into the correct response to a measurement problem, and predicts the fix: score post-training on a corpus where the relevant role is not rare, or reweight as in Eq. (9). We did not run a new experiment here; the token-share evidence is from the published measurements cited above.

# I Decision Aid

Table 12: Practical implications for capability evaluation based on loss signature properties.

| **Evaluation situation** | **Expected loss signature** | **Recommended measurement** |
|---|---|---|
| Frequent capability, large token share | Potentially large | Aggregate validation loss may be informative |
| Rare capability, small token share | Potentially small | Capability-specific benchmark |
| Known token role | Computable | Role-resolved loss (Eq. (2)) or reweighted loss (Eq. (9)) |
| Unknown capability role | Must be estimated | Context ablation or unsupervised discovery (Appendix F.1) |
| Competing route can substitute for the capability | May cancel ($\Lambda < 0$, large negative) | Measure the leakage term directly (Section 6.3) |
| Comparing loss-matched checkpoints | Loss alone is insufficient | Report capability accuracy separately |
| Evaluating instruction-following or refusal | Generic loss may miss behaviour | Task-specific evaluation |

# J Extended Limitations

Unsupervised role discovery (Appendix F.1) recovers a signature of the right magnitude but not the same token set; it is a screening tool, not a replacement for a stated hypothesis, and its context ablation is one particular choice. The natural-text position control is estimated on non-repeated tokens and assumes the generic positional advantage transfers; we study one corpus and one model family beyond the scale experiment. The real-checkpoint held-out corpus is WikiText-103 validation rather than the Pile validation split, so absolute loss values there should not be interpreted. Appendix H is analysis of published measurements rather than a new experiment. The numerical values for the 76M-parameter experiments are reported directly from the completed experiment logs and are defined as macros in the preamble for consistency throughout the paper.