# UNDERSTANDING LLM PARAMETER UPDATE SPARSITY THROUGH THE LENS OF FISHER

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

## ABSTRACT

Recent studies have observed that parameter changes during language-model post-training can be concentrated in a small subset of coordinates. This phenomenon has been reported in reinforcement learning, on-policy distillation, and supervised fine-tuning on near-policy data. Its recurrence across different post-training paradigms suggests shared structure in training dynamics. In this paper, we examine this pattern through the diagonal model Fisher, which measures the sensitivity of the model's output distribution to individual parameters and is independent of any particular reward or teacher signal. Theoretically, we show that small diagonal Fisher leads to small expected gradients across a range of training objectives, providing a common explanation for sparse gradient updates. Empirically, we test this connection in RL and OPD. We find that Fisher identifies where gradients are concentrated, and fixed sparse masks selected from the initial Fisher retain a large proportion of the improvement from full training. Finally, we investigate the mechanisms underlying low Fisher in on-policy training. Our results show that high-probability next tokens tend to have similar parameter sensitivities, contributing to low Fisher. Together, these results establish the diagonal model Fisher as a unifying perspective linking update sparsity to on-policy training dynamics in LLM post-training.

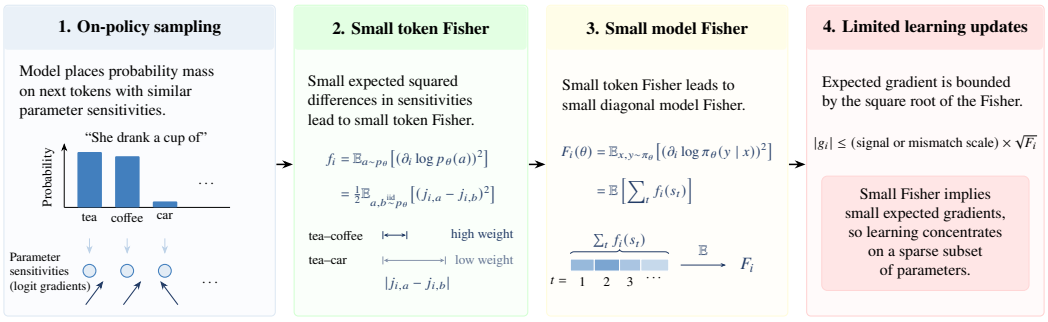


Figure 1: Logical chain of how on-policy training gives rise to low Fisher and concentrated learning. Section 4.1 studies the first two links, showing how on-policy sampling can produce small token Fisher and how these token-level contributions accumulate into small model Fisher. Section 3.1 establishes the third link, showing theoretically how small model Fisher limits expected gradients. Sections 3.2 and 3.3 empirically verify whether Fisher identifies where gradients and effective learning are concentrated.

## 1 INTRODUCTION

Post-training methods such as supervised fine-tuning (SFT), reinforcement learning (RL), and on-policy distillation (OPD) are widely used to adapt pretrained language models to downstream tasks and desired behaviors (Ouyang et al., 2022; Shao et al., 2024; DeepSeek-AI, 2025; Agarwal et al., 2024). Despite different objectives, these methods exhibit a similar pattern in parameter space: parameter changes can be highly concentrated. Recent work has found that only a small fraction of parameters undergo substantial changes during RL, OPD, and SFT on near-policy data, and that sparse subsets identified from training outcomes can retain much of the benefit of full optimization



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

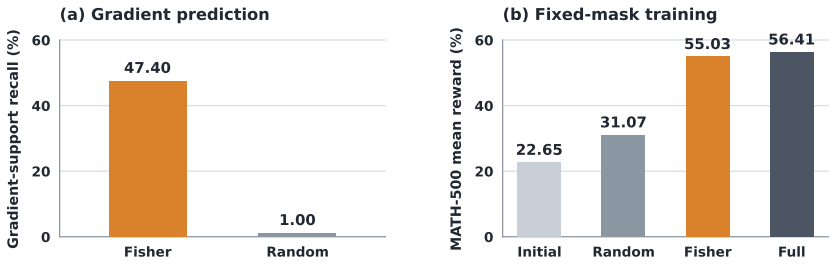


Figure 2: Fisher identifies a small set of parameters that supports learning. (a) Before training, the top 1% Fisher coordinates contain 47.40% of the top 1% gradient coordinates, compared with a 1.00% random baseline. (b) Updating only this Fisher-selected subset achieves much of the performance gain from updating all parameters (Full) and substantially outperforms an equally sized random subset. Results are for Qwen3-1.7B-Base trained with RL for mathematical reasoning.

(Mukherjee et al., 2025; 2026; Yu et al., 2026). This recurrence across post-training paradigms raises a fundamental question:

**Question.** *What common structure underlies the concentration of learning in a small subset of parameters during language-model post-training?*

Prior work has proposed several explanations for update sparsity (Mukherjee et al., 2025; 2026; Zhu et al., 2025; Shenfeld et al., 2026; Miahi & Belilovsky, 2026), as discussed in Section 2. These works explain sparsity from different perspectives, but do not provide a common mechanism linking on-policy sampling to where learning is concentrated across post-training methods. In this paper, we study the question through the diagonal model Fisher (defined in Equation 2), which measures how sensitive the model's own response distribution is to individual parameters. Unlike the gradient induced by a particular training objective, the Fisher depends only on the model and its response distribution, not on a particular reward or teacher signal. One key observation is that the expected gradient for every parameter coordinate can be bounded in the generic form (Section 3.1)

$$|g_i| \leq \text{signal or mismatch scale} \times \sqrt{F_i}, \tag{1}$$

where $F_i$ is the diagonal model Fisher and $g_i$ is the expected gradient for parameter coordinate $i$. When the objective-specific signal or mismatch scale is controlled, a small Fisher therefore implies a small expected gradient. This suggests the following answer:

**Answer.** *Diagonal model Fisher provides an objective-independent indicator for where learning can be concentrated during language-model post-training.*

We then empirically verify whether Fisher remains informative in actual training. In Section 3.2, we find that high-Fisher coordinates capture a much larger fraction of the largest-gradient coordinates than equally sized random subsets. In Section 3.3, updating only parameters with high initial Fisher while freezing the rest retains a substantial portion of the improvement achieved by full training. These results empirically support our Fisher-based explanation of concentrated learning.

We next turn to the low-Fisher pattern itself, asking what gives rise to it in on-policy training. In Section 4.1, we find that, for many parameters, the tokens favored by the model tend to respond similarly to parameter changes. We show that token Fisher is determined by probability-weighted differences in parameter sensitivities across candidate tokens. This leads to our central mechanistic explanation for low Fisher in on-policy training:

**Explanation.** *Fisher is small in on-policy training because the model's probability mass aligns with next tokens that have similar parameter sensitivities.*

These results bring together the two parts of our analysis: on-policy generation can give rise to low Fisher through similarity in the parameter sensitivities of likely next tokens, while low Fisher in

Table 1: Post-training objectives and update conventions. The demonstration distribution $\nu$, teacher $\pi_T$, and reward $r$ are fixed, and $R_T(x, y) = \sum_t \log[\pi_T(y_t \mid s_t)/p_\theta(y_t \mid s_t)]$. For PG-style OPD, $R_T$ is treated as a fixed reward in the policy-gradient estimator. For full-vocabulary OPD, the rollout distribution is held fixed during differentiation, and gradients are taken only through the student's next-token probabilities.

| Method | Response distribution $\mu$ | Learning signal $R_\theta$ |
|---|---|---|
| SFT | $\nu(y \mid x)$ | $\log \pi_\theta(y \mid x)$ |
| On-policy RL | $\pi_\theta(y \mid x)$ | $r(x, y)$ |
| PG-style OPD | $\pi_\theta(y \mid x)$ | $R_T(x, y)$ |
| Full-vocabulary OPD | $\pi_\theta(y \mid x)$ | $-\sum_t D_{\mathrm{KL}}(\pi_T(\cdot \mid s_t) \| p_\theta(\cdot \mid s_t))$ |

turn limits expected gradients and concentrates learning on a sparse set of parameters. The overall logical chain is illustrated by Figure 1, which summarizes the major steps toward our conclusion.

## 2 Background

**Diagonal model Fisher.** Let $x$ denote a prompt drawn from a fixed input distribution $\mathcal{D}_x$, $y$ a model response, and $\pi_\theta(y \mid x)$ the response distribution of a model with parameters $\theta$. For parameter $\theta_i$, the diagonal model Fisher is defined as

$$F_i(\theta) = \mathbb{E}_{x\sim\mathcal{D}_x}\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[(\partial_i \log \pi_\theta(y \mid x))^2\right]. \tag{2}$$

It measures the sensitivity of the model's own response distribution to a local perturbation of each parameter. For a fixed model and prompt distribution, it does not depend on the reward, teacher, or demonstration responses used for post-training.

**A common view of post-training objectives.** We write the training objectives and update surrogates in the common form $\max_\theta J(\theta)$, where

$$J(\theta) = \mathbb{E}_{x\sim\mathcal{D}_x}\mathbb{E}_{y\sim\mu(\cdot|x)}[R_\theta(x, y)].$$

Different post-training methods vary in the response distribution $\mu$ and the learning signal $R_\theta$. This common notation allows us to compare their gradient structure directly. As shown in Section 3.1, despite these differences, their expected coordinate gradients admit bounds with the same dependence on the diagonal model Fisher. This common dependence is what allows Fisher to provide a unified view across post-training objectives.

For autoregressive generation, let $s_t = (x, y_{<t})$ denote the prefix at step $t$ and $p_\theta(a \mid s_t)$ the corresponding next-token distribution. We consider SFT, on-policy RL, PG-style OPD, and full-vocabulary OPD with forward KL, as summarized in Table 1. Here, on-policy means that responses are sampled from the current model $\pi_\theta(\cdot \mid x)$.

**Related work.** Mukherjee et al. (2025) show that RL concentrates updates in small subnetworks whose isolated training can recover full-training performance. Yu et al. (2026) report similar observations for OPD despite its dense teacher supervision. Their subnetwork-retraining experiments identify coordinates from observed parameter changes. We instead ask whether Fisher measured before training identifies useful subsets. Rios et al. (2025); Adewuyi et al. (2026) motivate comparisons with random masks.

Existing explanations of update sparsity emphasize several mechanisms. Mukherjee et al. (2025) provide empirical evidence that sparsity is associated with training on near-policy data. Zhu et al. (2025) attribute update concentration to pretrained geometry that favors off-principal updates under a small KL budget, with theoretical guarantees relying on local curvature and spectral assumptions. Shenfeld et al. (2026); Miahi & Belilovsky (2026) show how limited numerical precision can contribute to update sparsity, while Mukherjee et al. (2026) show that optimizer choice strongly affects sparsity. We use the diagonal model Fisher to bound expected coordinate gradients across SFT, RL, and OPD, and investigate why Fisher can be small in on-policy training.

Table 2: Effective signals and squared scales $\mathbb{E}_{x,\, y\sim\pi_\theta}[a^2]$ in Equation (4). Here $w_\nu(x,y) = \nu(y \mid x)/\pi_\theta(y \mid x)$, $\bar{r}(x) = \mathbb{E}_{\pi_\theta}[r \mid x]$, $\bar{R}_T(x) = \mathbb{E}_{\pi_\theta}[R_T \mid x]$, and $A_T(x,y) = \sum_t \big(\pi_T(y_t \mid s_t)/p_\theta(y_t \mid s_t) - 1\big)$.

| Method | Effective signal $a(x,y)$ | Squared scale |
|---|---|---|
| SFT | $w_\nu(x,y) - 1$ | $\mathbb{E}_{x,\, y\sim\pi_\theta}[(w_\nu - 1)^2]$ |
| On-policy RL | $r(x,y) - \bar{r}(x)$ | $\mathbb{E}_x \operatorname{Var}_{\pi_\theta}(r \mid x)$ |
| PG-style OPD | $R_T(x,y) - \bar{R}_T(x)$ | $\mathbb{E}_x \operatorname{Var}_{\pi_\theta}(R_T \mid x)$ |
| Full-vocabulary OPD | $A_T(x,y)$ | $\mathbb{E}_{x,\, y\sim\pi_\theta}[A_T(x,y)^2]$ |

Fisher-based parameter selection has also been studied in SFT. Sung et al. (2021) select high-Fisher parameters for sparse training, while Xu et al. (2021) use an empirical Fisher computed from downstream labels to select trainable parameters. Sharma et al. (2024) show that Fisher information is concentrated in a small fraction of language-model parameters and use this structure to guide regularization. These works use Fisher to guide parameter selection or regularization in SFT. We instead use Fisher to explain why learning becomes concentrated, with fixed-mask training serving as an empirical test of our explanation rather than as a new parameter-selection method.

# 3 Fisher Identifies Where Learning Is Concentrated

## 3.1 Small Diagonal Fisher Implies Small Gradients

The goal of this section is to make the relationship in Equation (1) precise. Under the training objectives and update conventions in Table 1, each expected coordinate gradient can be expressed as the model score weighted by an objective-dependent signal.

Let $h_i(x,y) = \partial_i \log \pi_\theta(y \mid x)$ denote the score of parameter $\theta_i$, so that the diagonal Fisher defined in Equation (2) can be written as

$$F_i = \mathbb{E}_{x\sim\mathcal{D}_x,\, y\sim\pi_\theta(\cdot|x)}[h_i(x,y)^2].$$

For brevity, we suppress the fixed prompt distribution $\mathcal{D}_x$ in subsequent expectations.

**Proposition 1** (A common Fisher bound)**.** *For each method in Table 1, its expected update gradient under the stated differentiation convention can be written as*

$$g_i = \mathbb{E}_{x,\, y\sim\pi_\theta}[a(x,y)h_i(x,y)], \tag{3}$$

*where the effective signal $a(x,y)$ depends on the objective and is given in Table 2. Consequently,*

$$|g_i| \le \sqrt{\mathbb{E}_{x,\, y\sim\pi_\theta}[a(x,y)^2]}\sqrt{F_i}.^1 \tag{4}$$

*Proof sketch.* The model score satisfies

$$\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[h_i(x,y)] = \sum_y \pi_\theta(y \mid x)\partial_i \log \pi_\theta(y \mid x) = 0. \tag{5}$$

Using this identity together with the method-specific gradient expressions gives Equation (3), with derivations provided in Appendix A.1. Applying Cauchy–Schwarz to this representation yields Equation (4). □

**Interpretation.** All four methods admit the same dependence on $\sqrt{F_i}$ in the resulting upper bound. The remaining factor depends on the learning signal or distribution mismatch. When this factor is controlled, small diagonal Fisher is a *sufficient* condition for a small expected gradient for a *single step*.

[1] Similar Fisher-based gradient bounds are derived by Luo et al. (2026).

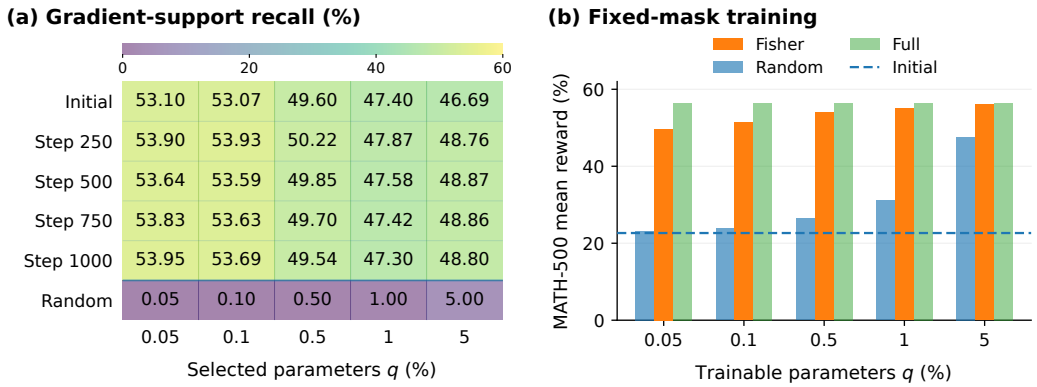


Figure 3: (a) Gradient-support recall (%) across selection percentages $q$. Initial denotes the pre-trained policy. Checkpoint rows average independent full-parameter training runs. Random is the analytic baseline. (b) Final MATH-500 mean reward across fixed-mask budgets. Fisher and Random denote fixed Fisher-selected and uniformly random masks, respectively, while Full updates all parameters. The dashed line marks the initial policy's score (22.65%).

Learning over an entire training trajectory also depends on the optimization dynamics, and the sparse subset that supports learning need not be unique. We next test whether this relationship is reflected in where gradients and effective learning are concentrated in practice. In Section 3.2, we ask whether Fisher identifies the coordinates on which gradients are concentrated at a given policy. In Section 3.3, we move beyond this local relationship and ask whether a sparse subset identified by Fisher can sustain learning over an entire training trajectory.

### 3.2 Fisher Identifies Where Gradients Are Concentrated

The bound in Section 3.1 implies that, when the objective-dependent scale is controlled, large expected gradients require sufficiently large Fisher. We therefore test whether the largest-gradient coordinates are captured by the largest-Fisher coordinates. For a fixed policy, we rank parameter coordinates separately by Fisher and by absolute estimated gradient. At a given selection percentage, we then measure how many of the largest-gradient coordinates are contained in the highest-Fisher set, using an equally sized random subset as a baseline.

We train Qwen3-1.7B-Base (Yang et al., 2025) with Dr. GRPO (Liu et al., 2025) for MATH-500 (Lightman et al., 2023). Experimental details are given in Appendix B.

Using the sequence score $h_i(x, y)$ defined in Section 3.1, we estimate

$$\widehat{F}_i = \frac{1}{N_F}\sum_{n=1}^{N_F} h_i(x_n, y_n)^2, \qquad \widehat{g}_i = \frac{1}{N_g G}\sum_{n=1}^{N_g}\sum_{j=1}^{G}(r_{nj} - \bar{r}_n)h_i(x_n, y_{nj}), \tag{6}$$

where $N_F$ and $N_g$ are the numbers of Fisher and gradient prompts, $G$ is the number of responses per gradient prompt, and $\bar{r}_n = G^{-1}\sum_{j=1}^{G} r_{nj}$.

For $P$ parameter coordinates and a selection percentage $q$, let $K_q = \lfloor qP/100 \rfloor$, and let $S_F(q)$ and $S_g(q)$ denote the globally largest $K_q$ coordinates of $\widehat{F}$ and $|\widehat{g}|$, respectively. We define gradient-support recall by

$$\mathrm{Recall}(q) = 100\frac{|S_F(q) \cap S_g(q)|}{K_q}. \tag{7}$$

A uniformly random subset of the same size has expected recall approximately $q\%$. As shown in Figure 3(a), high-Fisher coordinates are strongly enriched for large-gradient coordinates at initialization and throughout training across all evaluated cutoffs, which is consistent with our theory. Detailed results and corresponding experiments with OPD are reported in Appendix C.

**Takeaway.** High-Fisher coordinates capture where the largest gradients are concentrated.

### 3.3 Initial Fisher Masks Sustain Learning

Gradient-support recall tests the Fisher relationship locally at a fixed policy, but does not establish whether the identified coordinates remain useful for learning as the policy changes. In this section, for each parameter budget $q$, we select the globally highest-Fisher $q\%$ of parameters at initialization and keep this trainable set fixed throughout training. We compare these Fisher-selected masks with two uniformly random masks of the same size and with full-parameter training. We use four training seeds for each condition. Experimental details are given in Appendix B.

Figure 3(b) compares final MATH-500 performance across parameter budgets. The Fisher-selected masks substantially outperform equally sized random masks across all budgets, while recovering $85.21\%$ of the full-training improvement with only $0.1\%$ of parameters trainable and $98.82\%$ at $q = 5\%$. Together with the gradient-support results above, these results provide evidence that Fisher is informative not only about where gradients are concentrated at a given policy, but also about which sparse parameter subsets can support learning over an entire training trajectory. Detailed results and corresponding experiments with OPD are reported in Appendix C.

**Takeaway.** Initial Fisher identifies sparse parameter subsets that can sustain effective learning throughout training.

## 4 Why Does Low-Fisher Structure Arise, and Does It Persist?

The preceding results show that Fisher identifies where learning can be concentrated. Although on-policy RL and OPD use different learning signals, both train on responses generated by the current model. This shared sampling scheme motivates examining the Fisher along the model's own generations. We therefore ask why Fisher is small under on-policy generation and whether the resulting low-Fisher structure persists as the policy evolves.

### 4.1 Why Is the Fisher Small in On-Policy Training?

On-policy training uses responses sampled from the model's own distribution. By definition, a parameter has small Fisher when the log probabilities of these model-generated responses are less sensitive to changes in that parameter. Since a response probability is determined by the next-token probabilities along its generation, we can ask what makes these probabilities less sensitive. Note that shifting all logits by the same amount leaves the next-token distribution unchanged. More generally, a parameter has little effect on the distribution when its perturbation changes the logits of likely tokens similarly. This suggests a potential explanation for small Fisher: *for many parameters, the logits of high-probability next tokens may respond similarly to a small change in that parameter.*

One intuition for this alignment is that likely continuations of a model-generated prefix may share grammatical, topical, or formatting requirements. Many parameters may capture these shared requirements and affect several plausible candidates similarly, while other parameters distinguish between them (Geva et al., 2022; Gurnee et al., 2024). As an illustration, consider the prefix "She drank a cup of", for which "tea" and "coffee" are plausible continuations. A hypothetical parameter direction that increases the model's preference for liquids could affect the logits of both tokens similarly. Such a direction would have little effect on the model's relative preference between these likely continuations. We first make this intuition precise by deriving a token-level expression for Fisher and then empirically test the role of the model's probability assignment.

**Token-level Fisher.** To connect the intuition above to the model Fisher, we relate the Fisher of an entire response to token-level contributions and express these contributions through differences in logit sensitivities. [2]

[2]Related Fisher decompositions for conditional and sequential models are standard in natural-gradient and trajectory-Fisher analyses (Martens, 2020; Peters et al., 2005; Raykov & Veiga, 2026).

For a response $y = (y_1, \ldots, y_T)$, let $s_t = (x, y_{<t})$. Define the token score $\xi_{t,i}$ for parameter $\theta_i$ and the corresponding sequence score $h_i(x, y)$ as

$$\xi_{t,i} = \partial_i \log p_\theta(y_t \mid s_t), \qquad h_i(x, y) = \sum_t \xi_{t,i}.$$

Here, $p_\theta(\cdot \mid s_t)$ denotes the next-token distribution at prefix $s_t$. The *token Fisher* at prefix $s$ is

$$f_i(s) = \mathbb{E}_{a \sim p_\theta(\cdot|s)} \left[ (\partial_i \log p_\theta(a \mid s))^2 \right] = \mathbb{E}[\xi_{t,i}^2 \mid s_t = s].$$

Let $z_a(s)$ denote the logit of token $a$ at prefix $s$, write $p = p_\theta(\cdot \mid s)$ and $p_a = p_\theta(a \mid s) = e^{z_a(s)} / \sum_b e^{z_b(s)}$, and define the logit sensitivity to $\theta_i$ as $j_{i,a}(s) = \partial_i z_a(s)$.

**Proposition 2** (Token-level decomposition of model Fisher)**.** *With expectations over the fixed prompt distribution and responses sampled from $\pi_\theta(\cdot \mid x)$, the diagonal model Fisher satisfies*

$$F_i(\theta) = \mathbb{E}\left[ \sum_t f_i(s_t) \right]. \tag{8}$$

*At each prefix $s$, the token Fisher can be expressed as*

$$f_i(s) = \mathrm{Var}_{a \sim p}[j_{i,a}(s)] = \frac{1}{2} \mathbb{E}_{a, b \overset{\text{iid}}{\sim} p} \left[ (j_{i,a}(s) - j_{i,b}(s))^2 \right]. \tag{9}$$

*Proof sketch.* Under on-policy generation, each next token is sampled from $p_\theta(\cdot \mid s_t)$, so

$$\mathbb{E}[\xi_{t,i} \mid s_t] = \sum_a p_\theta(a \mid s_t) \partial_i \log p_\theta(a \mid s_t) = 0. \tag{10}$$

Thus, the token scores form a martingale difference sequence with respect to the generation history. For $u < t$, the earlier score $\xi_{u,i}$ is determined by $s_t$, and the tower property gives

$$\mathbb{E}[\xi_{u,i} \xi_{t,i}] = \mathbb{E}[\xi_{u,i} \mathbb{E}[\xi_{t,i} \mid s_t]] = 0.$$

The cross-token terms therefore vanish in expectation, yielding

$$F_i(\theta) = \mathbb{E}\left[ \left( \sum_t \xi_{t,i} \right)^2 \right] = \mathbb{E}\left[ \sum_t \xi_{t,i}^2 + 2 \sum_{u<t} \xi_{u,i} \xi_{t,i} \right] = \mathbb{E}\left[ \sum_t f_i(s_t) \right].$$

Derivation of Equation (9) is provided in Appendix A.2. □

**Interpretation.** Equation (8) shows that the diagonal model Fisher is determined by the token-level contributions accumulated along the model's own responses. Equation (9) formalizes the intuition above: token Fisher depends on how differently a parameter changes the logits of possible next tokens. It can therefore be small when the model assigns most probability to tokens with similar sensitivities.

**Testing the alignment between probability mass and sensitivities.** To empirically examine our explanation, we compare the same pairwise quantity under the model's probabilities and a uniform distribution over next tokens, while keeping the model and prefix fixed. Formally, let $u$ be the uniform distribution over the vocabulary of size $V$, with $u_a = 1/V$. We define the uniform reference as

$$f_i^{\text{unif}}(s) = \frac{1}{2} \mathbb{E}_{a, b \overset{\text{iid}}{\sim} u} \left[ (j_{i,a}(s) - j_{i,b}(s))^2 \right]. \tag{11}$$

Observing $f_i(s) < f_i^{\text{unif}}(s)$ would be consistent with the hypothesis that high-probability tokens tend to have more similar logit sensitivities to $\theta_i$ than tokens drawn uniformly from the vocabulary.

We perform this comparison at initialization and throughout training, both over all parameters and within current low-Fisher groups. For each group, we report its average token Fisher under the model's probabilities and the corresponding uniform reference, together with their ratio. Experimental details are provided in Appendix B.

Table 3: Average token Fisher under the model's probabilities (Real), the uniform reference (Uniform), and their ratio. Fisher values are in units of $10^{-6}$. Low-Fisher groups are selected from each checkpoint. Checkpoint values average four training runs. The initial policy is shared.

| | | | Current low-Fisher group (%) | | | | |
|---|---|---|---|---|---|---|---|
| Checkpoint | Metric | All | 99.95 | 99.9 | 99.5 | 99 | 95 |
| Initial | Real | 22.19 | 4.48 | 3.98 | 3.06 | 2.72 | 1.87 |
| | Uniform | 141.21 | 41.81 | 39.36 | 34.43 | 32.48 | 27.87 |
| | Ratio | **0.16** | **0.11** | **0.10** | **0.09** | **0.08** | **0.07** |
| Step 250 | Real | 11.57 | 3.00 | 2.71 | 2.13 | 1.90 | 1.32 |
| | Uniform | 184.21 | 46.77 | 43.07 | 35.72 | 32.89 | 26.01 |
| | Ratio | **0.06** | **0.06** | **0.06** | **0.06** | **0.06** | **0.05** |
| Step 500 | Real | 10.99 | 2.91 | 2.63 | 2.05 | 1.82 | 1.22 |
| | Uniform | 185.50 | 45.87 | 42.36 | 35.35 | 32.59 | 25.86 |
| | Ratio | **0.06** | **0.06** | **0.06** | **0.06** | **0.06** | **0.05** |
| Step 750 | Real | 10.09 | 2.69 | 2.43 | 1.89 | 1.68 | 1.11 |
| | Uniform | 200.53 | 48.69 | 44.93 | 37.41 | 34.43 | 27.26 |
| | Ratio | **0.05** | **0.06** | **0.05** | **0.05** | **0.05** | **0.04** |
| Step 1000 | Real | 10.71 | 2.86 | 2.57 | 2.00 | 1.77 | 1.17 |
| | Uniform | 242.76 | 63.16 | 58.05 | 47.88 | 43.89 | 34.20 |
| | Ratio | **0.04** | **0.05** | **0.04** | **0.04** | **0.04** | **0.03** |

As shown in Table 3, the Real-to-Uniform ratios are consistently below one at initialization and throughout training, both across all parameters and within the low-Fisher groups. This indicates that, along the model's own prefixes, high-probability next tokens tend to have more similar parameter sensitivities, reducing the token-level Fisher and hence the sequence Fisher accumulated along its generations.

**Takeaway.** In on-policy training, high-probability tokens tend to have more similar sensitivities, which leads to lower Fisher.

### 4.2 Stationarity of the Diagonal Fisher

In this section, we examine whether the initial Fisher pattern persists as the policy changes during training. This is relevant beyond a single step because the bounds in Section 3.1 depend on the Fisher of the current policy. It is also important for interpreting the fixed-mask experiments in Section 3.3, where trainable coordinates are selected only once at initialization.

We examine whether coordinates remain in their initial low- or high-Fisher sets and whether initially low-Fisher coordinates remain low in magnitude. Using the Fisher estimates from Section 3.2, we compare the initial policy with later checkpoints of full-parameter training. For each selection percentage $q$, let $K_q = \lfloor qP/100 \rfloor$, where $P$ is the number of parameters. At step $k$, let $H_k(q)$ contain the $K_q$ coordinates with the highest estimated Fisher, and let $L_k(q)$ contain the remaining coordinates.

We measure how much of each initial set remains in the corresponding set at step $k$:

$$\mathrm{Retention}_k^{\mathrm{low}}(q) = 100\,\frac{|L_0(q) \cap L_k(q)|}{P - K_q}, \qquad \mathrm{Retention}_k^{\mathrm{high}}(q) = 100\,\frac{|H_0(q) \cap H_k(q)|}{K_q}. \quad (12)$$

To measure whether initially low-Fisher coordinates remain low in magnitude, we keep the initial low-Fisher set $L_0(q)$ fixed and evaluate its average Fisher at each checkpoint:

$$M_k(q) = \frac{1}{|L_0(q)|} \sum_{i \in L_0(q)} \widehat{F}_{k,i}. \quad (13)$$

We report $M_k(q)/M_0(q)$, so that a value near one indicates that the average Fisher of the initially low-Fisher coordinates remains close to its initial scale. For comparison, we also report the corresponding ratio over all parameters.

Table 4: Persistence of the initial Fisher pattern during full-parameter training. Retention reports low/high-Fisher set overlap (%), and $M_k/M_0$ reports the Fisher magnitude of the fixed initial low-Fisher set relative to initialization. Values are averaged over four runs. Random gives the analytic retention baseline.

| | | | Selection $q$ (%) | | | | |
|---|---|---|---|---|---|---|---|
| Checkpoint | Metric | All | 0.05 | 0.1 | 0.5 | 1 | 5 |
| Step 250 | Retention | – | 99.99 / 75.76 | 99.98 / 76.96 | 99.89 / 77.67 | 99.76 / 76.70 | 98.78 / 76.89 |
| | $M_k/M_0$ | 0.70 | 0.92 | 0.92 | 0.93 | 0.93 | 0.92 |
| Step 500 | Retention | – | 99.99 / 73.02 | 99.97 / 74.36 | 99.88 / 75.30 | 99.74 / 74.37 | 98.66 / 74.62 |
| | $M_k/M_0$ | 0.73 | 0.99 | 1.00 | 1.00 | 1.01 | 0.99 |
| Step 750 | Retention | – | 99.99 / 72.02 | 99.97 / 73.48 | 99.87 / 74.61 | 99.73 / 73.68 | 98.63 / 73.96 |
| | $M_k/M_0$ | 0.76 | 1.04 | 1.06 | 1.07 | 1.07 | 1.06 |
| Step 1000 | Retention | – | 99.99 / 71.21 | 99.97 / 72.85 | 99.87 / 74.09 | 99.73 / 73.22 | 98.61 / 73.61 |
| | $M_k/M_0$ | 0.83 | 1.15 | 1.16 | 1.17 | 1.17 | 1.16 |
| Random | Retention | – | 99.95 / 0.05 | 99.90 / 0.10 | 99.50 / 0.50 | 99.00 / 1.00 | 95.00 / 5.00 |

As shown in Table 4, the initial high-Fisher sets retain roughly $70\%$–$78\%$ of their coordinates at later checkpoints, far above the corresponding random baselines. The average Fisher of the fixed initial low-Fisher sets remains close to its initial value through the middle of training and increases by only about $15\%$–$17\%$ by the final checkpoint.

**Takeaway.** The initial diagonal Fisher pattern remains largely preserved throughout training in our setting.

## 5 Limitations and Future Work

Our bounds concern expected gradients at a fixed policy rather than accumulated parameter changes. Extending the analysis to the latter requires accounting for gradient noise, optimizer dynamics, and learning-rate schedules. Although our token-level analysis explains how the model's probability weighting can reduce Fisher, it does not fully explain how architecture and pretraining shape the locations of low-Fisher coordinates.

Our masked-training experiments use dense computation and do not establish memory or runtime savings. Realizing such benefits requires implementations that exploit the selected masks. Beyond parameter selection, future work could examine Fisher as a training diagnostic. Extending the study to multiple tasks and longer training trajectories would also help determine whether Fisher-selected subsets can support new learning while preserving previously acquired capabilities.

## 6 Conclusion

We studied the concentration of learning during language-model post-training through the diagonal model Fisher. Across SFT, RL, and OPD, our analysis shows that small Fisher constrains expected coordinate gradients when the objective-specific signal or distribution-mismatch scale is controlled. Empirically, Fisher-selected coordinates are enriched for large gradients, and fixed masks selected at initialization retain a large proportion of the improvement from full training in RL and OPD.

Our token-level analysis expresses the model Fisher as the expected sum of probability-weighted variations in logit sensitivities. This explains how the model's own probability assignment can contribute to low Fisher when likely next tokens respond similarly to parameter changes. The observed persistence of Fisher-selected sets further indicates that the initial selection remains informative as training progresses. Together, these findings trace the path illustrated in Figure 1, from on-policy generation to low Fisher and small expected gradients for many parameters, helping explain why effective learning can be concentrated in a sparse parameter subset.

### AI USE STATEMENT

We used generative AI tools to aid or polish writing, for retrieval and discovery, for research ideation or execution, and to draft sections of the paper. We have reviewed all AI-assisted work. We take responsibility for the final content of this work with the aid of generative AI.

## REFERENCES

Israel Adewuyi, Solomon Okibe, and Vladmir Ivanov. The multiple ticket hypothesis: Random sparse subnetworks suffice for RLVR. *arXiv preprint arXiv:2602.01599*, 2026. URL https://arxiv.org/abs/2602.01599.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 30–45. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.3.

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. Universal neurons in GPT2 language models. *arXiv preprint arXiv:2401.12181*, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025. URL https://arxiv.org/abs/2503.20783.

Mao-Lin Luo, Zhe-Xu Wang, Zi-Hao Zhou, Bo Ye, Jian Zhao, Min-Ling Zhang, and Tong Wei. RL forgets! Towards continual policy optimization. *arXiv preprint arXiv:2607.04364*, 2026.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020. URL https://www.jmlr.org/papers/v21/17-678.html.

Erfan Miahi and Eugene Belilovsky. Understanding and exploiting weight update sparsity for communication-efficient distributed RL. arXiv preprint arXiv:2602.03839, 2026.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4399. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/bf235a1d6780afd979f2f81676f43413-Abstract-Conference.html.

Sagnik Mukherjee, Lifan Yuan, Pavan Jayasinha, Dilek Hakkani-Tür, and Hao Peng. Do we need Adam? Surprisingly strong and sparse reinforcement learning with SGD in LLMs. *arXiv preprint arXiv:2602.07729*, 2026. doi: 10.48550/arXiv.2602.07729. URL https://arxiv.org/abs/2602.07729.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-2011. URL https://proceedings.neurips.cc/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Jan Peters, Sethu Vijayakumar, and Stefan Schaal. Natural actor-critic. In *Machine Learning: ECML 2005*, pp. 280–291. Springer, 2005.

Yordan Raykov and Rodrigo Veiga. Information-geometric forward policy training in GFlowNets. *arXiv preprint arXiv:2608.03967*, 2026.

Jesus Rios, Pierre Dognin, Ronny Luss, and Karthikeyan Natesan Ramamurthy. Sparsity may be all you need: Sparse random parameter adaptation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 18650–18666, 2025. doi: 10.18653/v1/2025.findings-emnlp.1013. URL https://aclanthology.org/2025.findings-emnlp.1013/.

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

Mandar Sharma, Nikhil Muralidhar, Shengzhe Xu, Raquib Bin Yousuf, and Naren Ramakrishnan. Information guided regularization for fine-tuning language models. In *Conference on Language Modeling*, 2024.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7HNRYT4V44.

Yi-Lin Sung, Varun Nair, and Colin A. Raffel. Training neural networks with fixed sparse masks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24193–24205, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/cb2653f548f8709598e8b5156738cc51-Abstract.html.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. Raise a child in large language model: Towards effective and generalizable fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9514–9528, 2021. doi: 10.18653/v1/2021.emnlp-main.749. URL https://aclanthology.org/2021.emnlp-main.749/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Guo Yu, Wenlin Liu, Yulan Hu, Hao-Xuan Ma, Jun-Peng Jiang, and Han-Jia Ye. Dense supervision, sparse updates: On the sparsity and geometry of on-policy distillation. *arXiv preprint arXiv:2606.13657*, 2026. URL https://arxiv.org/abs/2606.13657.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, David Z. Pan, Zhangyang Wang, Yuandong Tian, and Kai Sheng Tai. The path not taken: RLVR provably learns off the principals. *arXiv preprint arXiv:2511.08567*, 2025. URL https://arxiv.org/abs/2511.08567.

# A Mathematical Derivations

## A.1 A Common Fisher Bound

We prove Proposition 1 under the differentiation conventions in Table 1. Throughout, the prompt distribution is fixed independently of $\theta$, and $\mathbb{E}_\pi$ denotes expectation over prompts and responses $y \sim \pi_\theta(\cdot \mid x)$. We assume differentiable probabilities, that differentiation and expectation can be interchanged, and finite second moments. For variable-length responses, generation stops at EOS or a fixed maximum horizon; token scores after termination are zero.

Write $h_i(x, y) = \partial_i \log \pi_\theta(y \mid x)$. Normalization gives

$$\mathbb{E}_\pi[h_i \mid x] = \sum_y \pi_\theta(y \mid x)\partial_i \log \pi_\theta(y \mid x) = \partial_i \sum_y \pi_\theta(y \mid x) = 0. \tag{14}$$

Thus any response-independent baseline $b(x)$ satisfies $\mathbb{E}_\pi[b(x)h_i] = 0$. Once $g_i = \mathbb{E}_\pi[ah_i]$ is established, Cauchy–Schwarz yields

$$|g_i| \le \sqrt{\mathbb{E}_\pi[a^2]}\sqrt{\mathbb{E}_\pi[h_i^2]} = \sqrt{\mathbb{E}_\pi[a^2]}\sqrt{F_i}. \tag{15}$$

It remains to identify $a$ for the four objectives.

**Supervised fine-tuning.** Let the fixed demonstration distribution $\nu(\cdot \mid x)$ be absolutely continuous with respect to $\pi_\theta(\cdot \mid x)$. With $w_\nu(x, y) = \nu(y \mid x)/\pi_\theta(y \mid x)$,

$$\begin{aligned} g_i &= \partial_i \mathbb{E}_{x,y\sim\nu}[\log \pi_\theta(y \mid x)] = \mathbb{E}_{x,y\sim\nu}[h_i] \\ &= \mathbb{E}_\pi[w_\nu h_i] = \mathbb{E}_\pi[(w_\nu - 1)h_i]. \end{aligned}$$

The last equality follows from Equation (14). Hence $a = w_\nu - 1$, and its squared scale is

$$\mathbb{E}_\pi[a^2] = \mathbb{E}_x\left[\chi^2(\nu(\cdot \mid x) \,\|\, \pi_\theta(\cdot \mid x))\right].$$

**On-policy reinforcement learning.** For a reward $r(x, y)$ with no direct parameter dependence, the score-function identity gives $g_i = \partial_i \mathbb{E}_\pi[r] = \mathbb{E}_\pi[rh_i]$. Subtracting $\bar{r}(x) = \mathbb{E}_\pi[r \mid x]$ leaves this expectation unchanged:

$$g_i = \mathbb{E}_\pi[(r - \bar{r}(x))h_i], \qquad \mathbb{E}_\pi[(r - \bar{r}(x))^2] = \mathbb{E}_x \operatorname{Var}_\pi(r \mid x).$$

This establishes the RL row of Table 2.

**PG-style on-policy distillation.** The sequence signal is

$$R_T(x, y) = \sum_t \log \frac{\pi_T(y_t \mid s_t)}{p_\theta(y_t \mid s_t)}.$$

The teacher is fixed and assigns positive probability to student-sampled tokens. Under the stated convention of holding this signal fixed during the local score-function update, $g_i = \mathbb{E}_\pi[R_T h_i]$. Letting $\bar{R}_T(x) = \mathbb{E}_\pi[R_T \mid x]$, we obtain

$$g_i = \mathbb{E}_\pi[(R_T - \bar{R}_T(x))h_i], \qquad \mathbb{E}_\pi[(R_T - \bar{R}_T(x))^2] = \mathbb{E}_x \operatorname{Var}_\pi(R_T \mid x).$$

**Full-vocabulary on-policy distillation.** At a sampled prefix $s_t$, write $p_t(a) = p_\theta(a \mid s_t)$ and $q_t(a) = \pi_T(a \mid s_t)$, with $q_t \ll p_t$. Differentiating negative teacher-to-student KL while holding the sampled prefixes and teacher fixed gives

$$g_i = \mathbb{E}_\pi\left[\sum_t \sum_a q_t(a)\partial_i \log p_t(a)\right]. \tag{16}$$

Define

$$b_t = \frac{q_t(y_t)}{p_t(y_t)} - 1, \qquad A_T = \sum_t b_t, \qquad \xi_{t,i} = \partial_i \log p_t(y_t), \qquad h_i = \sum_t \xi_{t,i}.$$

Both $b_t$ and $\xi_{t,i}$ have zero conditional mean given the history before token $t$:

$$\mathbb{E}[b_t \mid s_t] = \sum_a q_t(a) - 1 = 0, \qquad \mathbb{E}[\xi_{t,i} \mid s_t] = 0.$$

For $t \neq u$, conditioning on the history of the later token therefore gives $\mathbb{E}[b_t \xi_{u,i}] = 0$. For the same-token terms,

$$\mathbb{E}[b_t \xi_{t,i} \mid s_t] = \sum_a (q_t(a) - p_t(a)) \partial_i \log p_t(a) = \sum_a q_t(a) \partial_i \log p_t(a).$$

Consequently,

$$\mathbb{E}_\pi[A_T h_i] = \mathbb{E}_\pi \left[ \sum_t b_t \xi_{t,i} \right] = g_i.$$

This proves the required representation with $a = A_T$. The same conditional-mean argument eliminates cross terms in $A_T^2$:

$$\mathbb{E}_\pi[A_T^2] = \mathbb{E}_\pi \left[ \sum_t \chi^2(q_t \,\|\, p_t) \right].$$

Applying Equation (15) to each representation completes the proof of Proposition 1.

### A.2 Token-Level Decomposition of Model Fisher

We prove both identities in Proposition 2. Let $s_t = (x, y_{<t})$, $\xi_{t,i} = \partial_i \log p_\theta(y_t \mid s_t)$, and $h_i = \sum_t \xi_{t,i}$.

**From sequence Fisher to token Fisher.** At an active position, on-policy sampling and normalization imply

$$\mathbb{E}[\xi_{t,i} \mid s_t] = \sum_a p_\theta(a \mid s_t) \partial_i \log p_\theta(a \mid s_t) = 0.$$

For $u < t$, the earlier score $\xi_{u,i}$ is determined by the history at $t$. The tower property thus gives

$$\mathbb{E}[\xi_{u,i} \xi_{t,i}] = \mathbb{E}[\xi_{u,i} \mathbb{E}[\xi_{t,i} \mid s_t]] = 0.$$

The identity remains valid when scores are zero after termination. Expanding the square of the sequence score yields

$$\begin{aligned} F_i &= \mathbb{E}_\pi \left[ \left( \sum_t \xi_{t,i} \right)^2 \right] \\ &= \mathbb{E}_\pi \left[ \sum_t \xi_{t,i}^2 \right] + 2 \sum_{u<t} \mathbb{E}_\pi[\xi_{u,i} \xi_{t,i}] \\ &= \mathbb{E}_\pi \left[ \sum_t \mathbb{E}[\xi_{t,i}^2 \mid s_t] \right] = \mathbb{E}_\pi \left[ \sum_t f_i(s_t) \right]. \end{aligned}$$

The cancellation holds in expectation, not for each sampled response.

**Token Fisher as pairwise sensitivity variation.** Fix a prefix $s$, write $p_a = p_\theta(a \mid s)$, and let $j_{i,a} = \partial_i z_a(s)$. Differentiating the log softmax gives

$$\partial_i \log p_a = j_{i,a} - \sum_b p_b j_{i,b}.$$

With $\bar{j}_i = \sum_a p_a j_{i,a}$, this implies

$$f_i(s) = \sum_a p_a (j_{i,a} - \bar{j}_i)^2 = \mathrm{Var}_{a \sim p}[j_{i,a}].$$

For independent $a, b \sim p$, $\mathbb{E}[j_{i,a} j_{i,b}] = \bar{j}_i^2$. Therefore,

$$\begin{aligned} \frac{1}{2} \mathbb{E}_{a,b \overset{\text{iid}}{\sim} p}[(j_{i,a} - j_{i,b})^2] &= \frac{1}{2} \left( 2\mathbb{E}_{a \sim p}[j_{i,a}^2] - 2\bar{j}_i^2 \right) \\ &= \mathrm{Var}_{a \sim p}[j_{i,a}] = f_i(s). \end{aligned}$$

This proves the second identity and completes Proposition 2.

# B EXPERIMENTAL SETTINGS

## B.1 FISHER IDENTIFIES WHERE GRADIENTS ARE CONCENTRATED

For Section 3.2, we measure Qwen3-1.7B-Base at initialization and at checkpoints from four full-parameter RL training runs. Training and prompt formatting follow Appendix B.2. The Fisher and gradient prompt sets are sampled without replacement from the training data, are disjoint, and remain fixed across policies. Each policy generates its own on-policy responses.

Table 5: Test 1 measurement settings.

| Setting | Value |
|---|---|
| Model | Qwen3-1.7B-Base |
| Policies | Shared Initial; Full training seeds 0–3 at steps 250, 500, 750, and 1000 |
| Prompt source | The 8,513-problem RL training set |
| Fisher sample size | 800 prompts, one response per prompt |
| Gradient sample size | 300 prompts, eight responses per prompt |
| Generation | Temperature 1, top-$p = 1$, no top-$k$ truncation |
| Maximum prompt / response length | 1,024 / 4,096 tokens |
| Sampling seeds | Prompt selection: 800; Fisher generation: 100; gradient generation: 200 |
| Selection percentages $q$ | $0.05,\ 0.1,\ 0.5,\ 1,\ 5\%$ |
| Precision | FP32 parameters and score accumulation; BF16 computation |

**Estimators.** For each response, let $h_i(x, y) = \partial_i \sum_t \log p_\theta(y_t \mid x, y_{<t})$, summing over valid response tokens, including observed stopping tokens. With $\bar{r}_n = 8^{-1} \sum_{j=1}^{8} r_{nj}$, we compute

$$\widehat{F}_i = \frac{1}{800} \sum_{n=1}^{800} h_i(x_n, y_n)^2, \qquad \widehat{g}_i = \frac{1}{300 \cdot 8} \sum_{n=1}^{300} \sum_{j=1}^{8} (r_{nj} - \bar{r}_n) h_i(x_n, y_{nj}). \tag{17}$$

Fisher squares each trajectory's score before averaging. The gradient is averaged with its sign intact, and only the resulting vector is ranked by absolute value. Neither estimator divides by sampled response length. The gradient omits the training objective's common fixed-horizon factor, which does not affect coordinate ranking.

For each $q$, the highest $K_q = \lfloor qP/100 \rfloor$ coordinates are selected globally from $\widehat{F}$ and $|\widehat{g}|$, where $P$ counts unique parameter coordinates. Ties use a fixed coordinate order. Recall is 100 times the support intersection divided by $K_q$; an equal-size uniform random set has expected recall $100K_q/P\%$, approximately $q\%$.

## B.2 INITIAL FISHER MASKS SUSTAIN LEARNING

For Section 3.3, we compare Full, Fisher, and two random masks (R0/R1). At each budget, Fisher trains the globally highest $K_q$ initial-Fisher coordinates; R0/R1 train exact-size uniform global subsets with mask seeds 0 and 1. Each mask is fixed throughout training and shared across training seeds 0–3. Different budgets for a given random-mask seed use prefixes of the same random permutation. Full and Initial are shared across budgets within each suite.

RL and OPD use separate initial models and Fisher estimates. Both initial Fisher estimates use 800 training prompts with one on-policy response per prompt and the trajectory-score estimator above. Within a suite and training seed, all conditions share initialization, data-ordering and sampling rules, optimizer, training budget, and evaluation settings. Frozen gradients are zeroed after accumulation and before clipping and the optimizer step.

Both suites use FP32 parameters and gradient accumulation, BF16 computation, and no adapters. Each fresh batch receives one optimizer update. Sampling has no top-$k$ or min-$p$ restriction. Evaluation uses the same prompt format and length limits as training. Observed stopping tokens are retained; length-truncated responses are not discarded or given artificial EOS tokens.

Table 6: Training and evaluation settings for the reported Test 2 comparisons. All conditions within each column share these settings.

| Setting | RL | OPD |
|---|---|---|
| Initial model | Qwen3-1.7B-Base | Qwen3-1.7B (post-trained) |
| Teacher | None | Qwen3-4B-Instruct-2507, frozen |
| Training dataset | SimpleRL-Zoo, Abel levels 3–5 | DAPO-Math-17k |
| Training data size | 8,513 problems | 17,183 problems after deduplication and removal of exact MATH-500 matches |
| Objective | Dr. GRPO | Token-local PG-style distillation |
| Optimizer updates | 1,000 | 20 |
| Prompts per update | 4 | 512 |
| Responses per prompt | 8 | 1 |
| Microbatch size | 4 responses | 2 responses |
| Gradient accumulation | 8 microbatches | 256 microbatches |
| Learning rate | $5 \times 10^{-7}$ | $5 \times 10^{-6}$ |
| Optimizer | AdamW | AdamW |
| Adam parameters | $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$ | $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$ |
| Schedule / weight decay | Constant, no warmup / 0 | Constant, no warmup / 0 |
| Gradient clipping | Global norm 1.0 | Global norm 1.0 |
| Training temperature / top-$p$ | $1/1$ | $1/1$ |
| Maximum prompt / response tokens | $1{,}024/4{,}096$ | $2{,}048/8{,}192$ |
| Prompt format | Plain text | Chat, thinking disabled |
| Training seeds | 0, 1, 2, 3 | 0, 1, 2, 3 |
| Reported mask budgets $q$ | $0.05,\ 0.1,\ 0.5,\ 1,\ 5\%$ | $0.5,\ 1,\ 5\%$ |
| Evaluation dataset | MATH-500, 500 problems | AIME24, 30 problems |
| Responses per evaluation problem | 4 | 8 |
| Evaluation temperature / top-$p$ | $1/1$ | $1/0.7$ |
| Evaluation generation seeds | 300–303 | 300–307 |
| Evaluation metrics | Mean response accuracy | Avg@8 and pass@8 |

**RL objective and prompts.** Rewards are binary mathematical-correctness scores from the shared answer extraction and `math-verify` scorer. Dr. GRPO subtracts the prompt group's mean reward without standard-deviation normalization. The token-summed objective is divided by the response count and the fixed horizon of 4,096. The clipping parameter is 0.2; there is no reference-policy KL penalty or entropy bonus. Prompts use

```
Question:
{question}
Answer:
Let's think step by step.
```

**OPD objective and prompts.** The frozen teacher scores tokens sampled from the student. For a batch with $N_{\text{tok}}$ valid response tokens, the implemented loss is

$$L_{\text{K2}} = \frac{1}{2N_{\text{tok}}} \sum_{n,t \text{ valid}} \left(\log p_\theta(y_{nt} \mid s_{nt}) - \text{sg}[\log \pi_T(y_{nt} \mid s_{nt})]\right)^2. \tag{18}$$

Its negative gradient weights each token score by $\log \pi_T - \log p_\theta$, using the same whole-batch denominator for all microbatches. This is a token-local update on fixed sampled trajectories, rather than the sequence-signal estimator $R_T h_i$ in the theoretical PG convention. Answer rewards, group centering, and PPO clipping are not used in the OPD loss.

Student and teacher receive the same prompt token IDs. The Qwen chat template is applied with `enable_thinking=False` to

```
{question}
Please reason step by step, and put your final answer within \boxed{}.
```

Generation stops at `<|endoftext|>` or `<|im_end|>`. The OPD evaluation uses AIME I and II from `HuggingFaceH4/aime_2024`.

**Evaluation and averaging.** For $N$ problems and $G$ responses per problem, mean response accuracy is $100(NG)^{-1}\sum_{n,j} r_{nj}$. Thus RL averages 2,000 binary rewards, and OPD Avg@8 averages 240. OPD pass@8 instead reports the percentage of questions with at least one correct response among the eight. All conditions use the same binary scorer.

For each training seed $s$ and evaluation metric, we compute

$$M_{R,s} = \frac{M_{R0,s} + M_{R1,s}}{2}, \qquad \Delta_s = M_{F,s} - M_{R,s}, \qquad \text{Recovery}_s = 100\,\frac{M_{F,s} - M_0}{M_{\text{Full},s} - M_0}.$$

Here $M_0$ is the corresponding suite's shared initial score. Tables report equal-weight means over the four training seeds. Recovery is calculated within each seed before averaging, and $\Delta$ is in percentage points.

### B.3 Testing the Alignment Between Probability Mass and Sensitivities

For the comparison in Section 4.1, Real and Uniform use the same model and exactly the same prefixes. Only the distribution used to sample candidate next-token pairs changes.

**Measurement and averaging.** The selected prompts and positions are fixed before measurement and are not filtered by correctness. For a parameter group $\mathcal{G}$ and a sampled pair $(a, b)$, we compute

$$D_{\mathcal{G}}(s, a, b) = \frac{1}{2|\mathcal{G}|} \sum_{i \in \mathcal{G}} \big(j_{i,a}(s) - j_{i,b}(s)\big)^2.$$

This is obtained by backpropagating $(z_a - z_b)/\sqrt{2}$, squaring each coordinate gradient, and averaging within the group. Model parameters are not updated. All six groups share each condition's pairs and backward calculations. Equal-token pairs contribute zero and remain in the sample count.

For each condition, we average the eight pairs at each position, then the eight positions within a response, and finally the 200 prompt means with equal weight. Each policy therefore uses 1,600

Table 7: Settings for the Real–Uniform token-Fisher comparison.

| Setting | Value |
|---|---|
| Model and policies | Qwen3-1.7B-Base; shared Initial and Full seeds 0–3 at steps 250/500/750/1000 |
| Prompts | 200 prompts sampled without replacement from Test 1's 300 gradient prompts, shared across policies |
| Responses | Each policy's first saved on-policy response per prompt |
| Response generation | Temperature 1, top-$p = 1$, no top-$k$ truncation; prompt/response limits 1,024/4,096 |
| Positions | Eight uniformly selected positions per response; identical for Real and Uniform |
| Token pairs | Eight independent pairs per position per condition, sampled with replacement |
| Real distribution | The model's full softmax probabilities |
| Uniform distribution | Uniform over the full vocabulary, including special tokens |
| Parameter groups | All; current-policy low-Fisher 99.95%, 99.9%, 99.5%, 99%, and 95% |
| Group selection | Each policy's own 800-response sequence-Fisher estimate |

prefixes and 12,800 pairs per condition. This estimates mean token Fisher over sampled positions, not the unnormalized sequence Fisher used in Test 1. Across the four training seeds, Real and Uniform are averaged separately before taking their ratio; Initial is measured once. The comparison keeps the original logit sensitivities fixed while changing both probability concentration and assignment.

# C Additional Experimental Results

The following results supplement Sections 3.2 and 3.3. In the fixed-mask learning tables, entries are equal-weight means over training seeds 0–3. R0 and R1 denote the two random masks; Random averages their scores. Recovery is computed within each training seed before averaging, and $\Delta$ denotes Fisher minus Random in percentage points. Training and evaluation settings are given in Appendix B.2.

## C.1 RL: MATH-500

Table 8 reports the numerical values for the MATH-500 comparison in the main text. The score is the mean binary reward over four responses to each of 500 problems. Initial scores $22.65\%$; its evaluation and the Full results are shared across mask budgets.

Table 8: RL on MATH-500: four-training-seed mean response accuracy (%). The initial model is Qwen3-1.7B-Base. All models are evaluated after 1,000 updates. Recovery is a percentage and $\Delta$ is in percentage points.

| $q$ (%) | Full | Fisher | R0 | R1 | Random | Recovery | $\Delta$ |
|---|---|---|---|---|---|---|---|
| 0.05 | 56.41 | 49.65 | 22.75 | 23.66 | 23.21 | 79.98 | 26.44 |
| 0.1 | 56.41 | 51.41 | 24.36 | 23.44 | 23.90 | 85.21 | 27.51 |
| 0.5 | 56.41 | 54.00 | 26.39 | 26.45 | 26.42 | 92.87 | 27.58 |
| 1 | 56.41 | 55.03 | 30.93 | 31.21 | 31.07 | 95.90 | 23.96 |
| 5 | 56.41 | 56.01 | 47.58 | 47.60 | 47.59 | 98.82 | 8.43 |

## C.2 OPD: Fisher–Gradient Support Recall

We extend the same-policy Fisher–gradient support comparison in Section 3.2 to OPD. Table 9 compares the highest-Fisher coordinates with those having the largest gradient magnitudes at the shared initial policy and checkpoints from full-parameter OPD training. Checkpoint results average training seeds 0 and 1; the initial result is shared.

## C.3 OPD: AIME24

We report AIME24 using the OPD protocol's chat non-thinking evaluation, which matches the training prompt format. Table 10 contains the completed four-seed comparisons at $q = 0.5\%$, $1\%$, and

Table 9: OPD Fisher–gradient support recall (%), supplementing Section 3.2. At each policy, recall is the fraction of the top-$q\%$ gradient-magnitude coordinates captured by the top-$q\%$ Fisher coordinates. Checkpoint rows are equal-weight means over full-training seeds 0 and 1; the shared initial policy is measured once. Random gives the analytic expectation for a uniform subset of the same size.

| Checkpoint | $q = 0.5\%$ | $q = 1\%$ | $q = 5\%$ |
|---|---|---|---|
| Initial (step 0) | 37.12 | 35.66 | 37.84 |
| Step 5 | 43.26 | 41.05 | 41.90 |
| Step 10 | 47.64 | 45.63 | 47.64 |
| Step 15 | 47.75 | 45.80 | 48.25 |
| Step 20 | 49.98 | 47.86 | 50.11 |
| Random | 0.50 | 1.00 | 5.00 |

$5\%$. Avg@8 averages all $30 \times 8$ binary rewards; pass@8 is the percentage of problems answered correctly at least once among the eight responses. The shared initial Qwen3-1.7B student scores $16.25\%$ Avg@8 and $36.67\%$ pass@8. These OPD scores use a different initial model and evaluation protocol from the RL results above.

Table 10: OPD on AIME24: means over four training seeds after 20 updates. Scores and Recovery are percentages; $\Delta$ is in percentage points. Both metrics use the same eight responses per problem.

| $q$ (%) | Metric | Full | Fisher | R0 | R1 | Random | Recovery | $\Delta$ |
|---|---|---|---|---|---|---|---|---|
| 0.5 | Avg@8 | 26.67 | 25.10 | 14.69 | 15.42 | 15.05 | 89.49 | 10.05 |
| 0.5 | pass@8 | 47.50 | 52.50 | 33.33 | 30.83 | 32.08 | 156.25 | 20.42 |
| 1 | Avg@8 | 26.67 | 26.15 | 15.00 | 16.46 | 15.73 | 99.47 | 10.42 |
| 1 | pass@8 | 47.50 | 55.00 | 29.17 | 38.33 | 33.75 | 185.42 | 21.25 |
| 5 | Avg@8 | 26.67 | 26.25 | 24.27 | 23.33 | 23.80 | 106.59 | 2.45 |
| 5 | pass@8 | 47.50 | 50.83 | 49.17 | 45.83 | 47.50 | 147.92 | 3.33 |