Equal Gain per Unit Statistical Distance: A Principle for Token Reweighting

Anonymous authors
Paper under double-blind review

Abstract

Supervised fine-tuning (SFT) offers a simple way to learn from fixed demonstrations, but improving its generalization remains a challenge. Recent methods assign token-level weights to the cross-entropy loss and report empirical gains, yet benchmark results alone do not explain how these weights value learning progress. In this work, we compare token weights in a common unit: the objective gain each weight assigns per unit of Fisher–Rao (FR) distance toward the demonstrated token. We find that Dynamic Fine-Tuning (DFT) and InfoSFT each assign more uniform gains than SFT over different ranges of target-token probability, offering a possible explanation for their empirical advantages. Building on this insight, we derive Vocab Relative Advantage (VRA), the token weight that gives uniform gain per unit of FR distance across the probability range and is unique up to scale. VRA achieves higher average accuracy than both SFT and DFT on each of six backbones. On Qwen3-8B, it reaches 39.23% across five mathematics benchmarks, compared with 29.30% for SFT, and also surpasses DFT, InfoSFT and ProFit. Matched one-step comparisons further show that, relative to SFT, VRA redistributes progress from low-confidence targets to middle- and high-confidence targets.

1 Introduction

Supervised fine-tuning (SFT) is a standard way to adapt pretrained language models to instructions and downstream tasks (Wei et al., 2022; Sanh et al., 2022; Chung et al., 2024). Learning from demonstrations provides a straightforward training pipeline, and even small, carefully curated datasets can teach useful response behaviors (Ouyang et al., 2022; Zhou et al., 2023). Yet controlled comparisons have identified settings where SFT generalizes less effectively than reinforcement learning (RL) (Chu et al., 2025). RL can improve behavior through human preferences or verifiable outcomes (Christiano et al., 2017; Stiennon et al., 2020; Shao et al., 2024), but commonly requires additional response generation and reward evaluation (Ouyang et al., 2022; Rafailov et al., 2023). This motivates a complementary question: *how can we learn more effectively from fixed demonstrations while retaining the simplicity of SFT?*

One line of work addresses this question by modifying the supervised objective (Pang & He, 2021; Qin & Springenberg, 2025), applying token-level weights to each token's cross-entropy loss in standard SFT. Dynamic Fine-Tuning (DFT) (Wu et al., 2026) weights tokens by their current model probability, which we call *confidence*. InfoSFT (Sabbaghi et al., 2026) emphasizes intermediate-confidence tokens to balance acquisition and retention, while ProFit (Liu et al., 2026) filters out tokens below a probability threshold. Anchored Supervised Fine-Tuning (ASFT) (Zhu et al., 2026a) adds a reference-distribution penalty to stabilize probability-weighted training. These methods report improvements on their respective benchmarks. Beyond benchmark performance, is there a complementary perspective for evaluating their weighting choices?

In this work, we use Fisher–Rao (FR) distance (Ay, 2023) as a common unit for comparing the learning incentives induced by different token weights. We measure the local objective gain per unit of FR distance toward the demonstrated token, treating all other tokens as one alternative. As Figure 1 shows, DFT and InfoSFT each assign more uniform gains than SFT over a portion of the confidence range, although these regions differ. This partial balance offers a possible explanation for their empirical advantages over SFT. It also suggests extending the balance beyond the regions favored by individual methods.

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

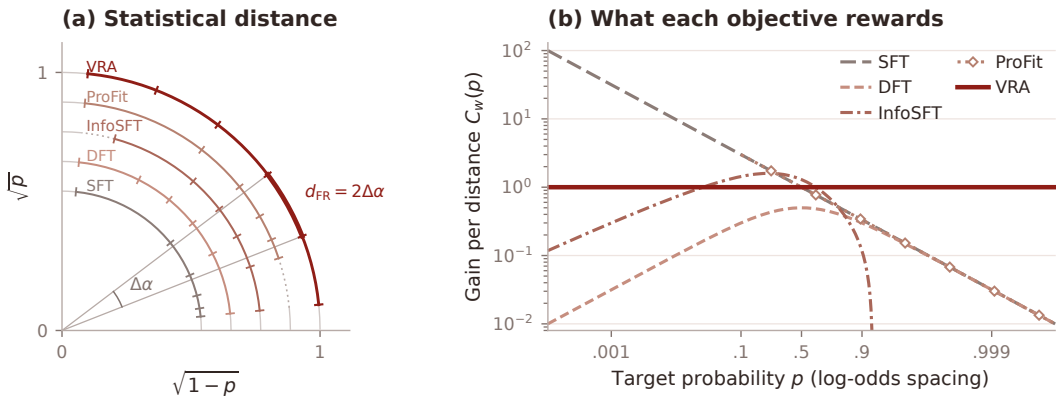


Figure 1: **Existing token weights value equal statistical progress very unequally; VRA is the unique flat line.** (a) The Bernoulli family in square-root coordinates: arc length is Fisher–Rao distance, $d_{\mathrm{FR}} = 2\Delta\alpha$. Ticks split each rule's objective gain over $p \in [0.01, 0.99]$ into five equal parts, so closely spaced ticks mark the confidence regions a rule rewards most per unit distance; only VRA spaces them equally. Dotted segments carry zero gain (ProFit below $\tau = 0.1$, InfoSFT above $\bar{p} = 0.93$); inner arcs are radial offsets for display. (b) Analytic profiles $C_w(p)$. Equal gain per distance is horizontal: SFT diverges at low confidence, DFT and InfoSFT vanish there, and ProFit gates at τ.

Building on this insight, we propose *Vocab Relative Advantage* (VRA), a token-weighting rule that extends this balance across the confidence range: it provides uniform objective gain per unit of FR distance, as illustrated by VRA's flat gain profile in Figure 1.

Empirically, VRA delivers consistent improvements in mathematical reasoning. On all six language-model backbones, it achieves higher average accuracy than SFT and DFT over five mathematics benchmarks, evaluating each problem with eight sampled solutions. On Qwen3-8B, VRA reaches 39.23% average accuracy versus 29.30% for SFT, and also exceeds DFT, InfoSFT, and ProFit in the aggregate. Matched one-step comparisons on fixed batches also show that, relative to SFT, VRA redistributes progress from low- to middle- and high-confidence targets.

Our contributions are threefold: (1) We compare token weights by objective gain per unit FR distance and show that likelihood and probability weighting assign diverging and vanishing incentives, respectively, to low-confidence progress. (2) We derive VRA as the unique weight satisfying a normalized equal-gain principle, establish its bounded target-logit gradient, and provide a lightweight stabilized implementation. (3) We show that VRA achieves higher average mean@8 than SFT and DFT on all six backbones, also surpassing InfoSFT and ProFit in the main comparison, and measure its signal redistribution through gold-data censuses and matched one-step updates.

2 A Statistical Unit for Token-Level Progress

2.1 Token-Weighted Cross-Entropy

Let x be a prompt and $y^\star$ a target response. At position t, write $s = (x, y^\star_{<t})$, $a^\star = y^\star_t$, and $p = \pi_\theta(a^\star \mid s) \in (0, 1)$. Token weighting multiplies the standard cross-entropy loss by a token-level weight $w(p)$:

$$\mathcal{L}_t = \mathrm{sg}(w(p))\,(-\log p).$$

Here $-\log p$ is the cross-entropy for the demonstrated token, and sg holds the weight fixed during backpropagation. Standard SFT uses $w(p) = 1$; token-weighting methods differ only in $w(p)$.

Different weights assign different gradient magnitudes to the same token. Comparing these choices requires a unit of learning progress that does not depend on the weight itself; otherwise each method is measured by its own ruler. Because the weight is detached, it enters each token's gradient only as a scalar function of p multiplying $\nabla_\theta \log p$. For comparing the local objective gains induced by these weights, it is therefore sufficient to work with the target probability and to define a distance on its one-dimensional axis.

2.2 A Canonical Distance on Target Probability

Since the weighted objective depends on the vocabulary distribution only through the target probability p, the distribution over V tokens can be reduced to Bernoulli(p) via the projection $a \mapsto \mathbf{1}[a = a^\star]$. This coarse graining retains precisely the quantity on which the objective depends—the probability of the demonstrated token—and discards reallocations among non-target alternatives, which do not enter the detached weighted log-likelihood.

On this Bernoulli family, the Fisher–Rao metric provides a distance that is invariant under reparameterization and contracts under coarse graining, making it a canonical unit for measuring changes in p (Čencov, 1982; Ay et al., 2017). With the conventional Fisher normalization, the metric and its arc-length coordinate are

$$ds_F^2 = \frac{dp^2}{p(1-p)}, \qquad \ell(p) = 2\arcsin\sqrt{p}, \qquad d\ell = \frac{dp}{\sqrt{p(1-p)}}.$$

The distance between two Bernoulli states is $|\ell(p_2) - \ell(p_1)|$. Re-expressing p in another smooth coordinate leaves this distance unchanged.

2.3 Marginal Gain per Unit Distance

With a weight-independent distance in hand, we can ask: when the model moves one Fisher–Rao unit toward the target token, how much objective gain does each weighting rule assign?

A detached weight defines a local ascent direction with potential

$$U_w(p) = \int_{1/2}^{p} \frac{w(u)}{u}\, du, \qquad dU_w = w(p)\, d\log p.$$

Thus the weighted likelihood gradient is $\nabla_\theta U_w(p)$.

Definition 1 (Statistical-progress gain)**.** The marginal gain profile of a token weight is

$$\boxed{C_w(p) = \frac{dU_w}{d\ell} = w(p)\sqrt{\frac{1-p}{p}}.}$$

It measures the objective gain assigned to a unit of statistical progress toward the target event.

Three equivalent readings make this profile concrete. By the chain rule it is the ratio of the utility differential to Fisher–Rao arc length. In the square-root embedding $(\sqrt{1-p}, \sqrt{p})$, the distribution lies on a quarter circle and ℓ is twice its angle. Finally, writing $s_w = w(p)(1-p)$ for the target-logit ascent gradient and $\sigma = \sqrt{p(1-p)}$ for the target-match standard deviation gives $C_w = s_w/\sigma$. Appendix B gives the derivations. The profile measures how an objective values progress; model structure and optimization also shape realized progress, a gap we measure directly with matched one-step updates in Section 6.2.

3 What Existing Weights Reward

3.1 From Uniform Weights to Unequal Gain

Unit weights attach the same multiplier to every log-probability, but assign very different gains to equal statistical progress. For SFT, $C_1(p) = \sqrt{(1-p)/p}$, which diverges as $p \to 0$ and vanishes as $p \to 1$. For illustration, the gain at $p = 0.01$ is 99 times the gain at $p = 0.99$, although the raw weights are identical. The steep rise at low confidence is visible in the SFT curve of Figure 1(b).

The task-level issue is the value assigned to unfamiliar targets. A target does not acquire additional semantic value merely because its current probability is extremely small. Yet unit-weight likelihood automatically assigns unbounded marginal gain to its statistical progress.

Table 1: **A taxonomy in statistical-progress units.** Constants that rescale the entire update do not change the endpoint classes. The last row anticipates the equal-gain construction in Section 4.

Rule	Weight $w(p)$	Marginal gain $C_w(p)$
SFT / unit	1	$\sqrt{(1-p)/p}$: low-end divergence
DFT	p	$\sqrt{p(1-p)}$: both ends vanish
InfoSFT	$p[a - \text{logit}\, p]_+$	$\sqrt{p(1-p)}[a - \text{logit}\, p]_+$
ProFit	$\mathbf{1}[p > \tau]$	Unit profile above τ, zero below
VRA	$\sqrt{p/(1-p)}$	1: constant over confidence

For InfoSFT, $a = \text{logit}\, \bar{p}$, where $\bar{p}$ is its fixed calibration probability.

3.2 Suppressing Unfamiliar Targets Can Suppress Necessary Learning

Probability weighting removes the low-confidence divergence by reversing its endpoint behavior. DFT uses $w(p) = p$, giving $C_{\text{DFT}}(p) = \sqrt{p(1-p)}$; unit progress receives vanishing gain at both endpoints (Wu et al., 2026). InfoSFT uses the calibrated rule $w_{\text{Info}}(p) = p[\text{logit}\, \bar{p} - \text{logit}\, p]_+$, whose derivation targets population fitting under a KL budget (Sabbaghi et al., 2026). Its low-confidence gain behaves as $\sqrt{p}\log(1/p)$: the logarithmic correction softens the suppression relative to DFT, but the limit remains zero. At $p \geq \bar{p}$, positive clipping sets the direct signal to zero.

ProFit makes selection explicit with $w(p) = \mathbf{1}[p > \tau]$ (Liu et al., 2026). Below the threshold there is no direct fitting signal; above it, the profile is the unit-weight profile. Table 1 summarizes all three alternatives to likelihood; Figure 1 visualizes the contrasts, with panel (a) showing non-uniform tick spacing on the Bernoulli arc and panel (b) plotting the analytic profiles.

The quantity being equalized matters. Flattening the target-logit gradient would instead require $w(p) \propto 1/(1-p)$, yielding $C_w(p) \propto 1/\sqrt{p(1-p)}$. This analytic counterpoint shows that constant logit response does not give constant statistical-progress gain.

These choices leave a common design problem. Low probability can mark either useful novelty or incidental wording, so confidence alone does not justify an extreme judgment of learning value. We seek to remove the unbounded gain attached to unfamiliarity without making the gain for unfamiliar progress vanish. Equal gain provides one explicit default principle for meeting both requirements.

4 Equal Gain and Vocab Relative Advantage

4.1 An Explicit Default Principle

We propose to value equal amounts of target-event statistical progress equally, independent of current confidence. Because confidence alone supplies no additional estimate of learning value, this default assigns the same marginal value to a fixed unit of statistical progress throughout the confidence range.

Proposition 1 (Equal gain determines the normalized weight)**.** *Let $w : (0,1) \to (0,\infty)$ satisfy $C_w(p) = c$ for a constant c, and normalize $w(1/2) = 1$. Then the unique weight is*

$$\boxed{w_{\text{VRA}}(p) = \sqrt{\frac{p}{1-p}},} \qquad C_{\text{VRA}}(p) = 1.$$

Proof. Solving $w(p)\sqrt{(1-p)/p} = c$ gives $w(p) = c\sqrt{p/(1-p)}$; the midpoint normalization fixes $c = 1$. □

The resulting Vocab Relative Advantage (VRA)[1] removes both the diverging gain of likelihood and the vanishing low-confidence gain of probability weighting. Uniqueness is conditional on equal gain:

[1]The name reflects an equivalent policy-gradient reading: standardizing the target-match reward in DFT's policy-gradient view of SFT over the vocabulary distribution, in the manner of GRPO's group-relative advantages, yields the same weight (Appendix C).

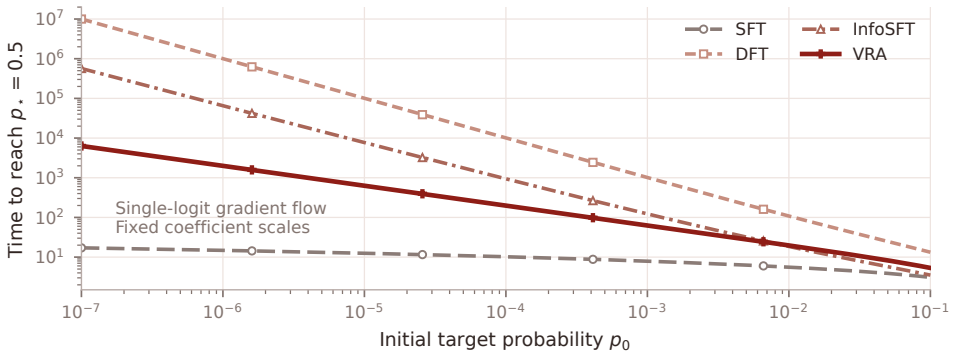


Figure 2: **Acquisition in the scalar reference model.** Time to reach $p_\star = 0.5$ from p_0, under the displayed single-logit gradient flow. SFT, DFT, and ideal VRA use exact closed forms; InfoSFT uses numerical quadrature with $\bar{p} = 0.93$. Canonical coefficient scales are held fixed. VRA is slower than likelihood but has a lower low-probability time exponent than DFT, avoiding the latter's stronger acquisition barrier. These are analytic reference dynamics, not language-model training curves.

endpoint boundedness alone would admit other weights. The associated potential and target-logit ascent gradient have simple forms,

$$U_{\mathrm{VRA}}(p) = 2\arcsin\sqrt{p} - \frac{\pi}{2}, \qquad \frac{\partial U_{\mathrm{VRA}}}{\partial z^\star} = w_{\mathrm{VRA}}(p)(1-p) = \sqrt{p(1-p)} \le \frac{1}{2}.$$

Thus a large coefficient near $p = 1$ does not imply an unbounded target-logit gradient. The potential increases linearly in Fisher–Rao arc length, which is exactly the chosen equal-gain property.

4.2 Does Low-Probability Acquisition Remain Possible?

The weight itself still approaches zero at low confidence. To characterize what this means for learning, consider an isolated Bernoulli logit $p = \mathrm{sigmoid}(z)$, trained by continuous-time gradient ascent on U_w with unit learning-rate scale. Then

$$\dot{z} = w(p)(1-p), \qquad \dot{p} = w(p)p(1-p)^2, \qquad T_w(p_0 \to p_\star) = \int_{p_0}^{p_\star} \frac{dp}{w(p)p(1-p)^2}.$$

For a fixed target $p_\star \in (0,1)$, the low-initial-probability asymptotics are

$$T_{\mathrm{SFT}} = \Theta(\log(1/p_0)), \quad T_{\mathrm{VRA}} = \Theta(p_0^{-1/2}),$$
$$T_{\mathrm{DFT}} = \Theta(p_0^{-1}), \qquad T_{\mathrm{Info}} = \Theta\big([p_0\log(1/p_0)]^{-1}\big).$$

The InfoSFT statement holds for fixed $\bar{p}$ and $p_\star < \bar{p}$; its clipped threshold cannot be crossed in finite time by this direct scalar signal. ProFit's strict gate $w(p) = \mathbf{1}[p > \tau]$ gives $T_{\mathrm{ProFit}} = \infty$ for $p_0 \le \tau < p_\star$, and $T_{\mathrm{ProFit}} = T_{\mathrm{SFT}}$ for $\tau < p_0 < p_\star$. Exact SFT, DFT, and VRA primitives and the InfoSFT asymptotic proof appear in Proposition 5. Figure 2 plots these learning times as a function of p_0: the $p_0^{-1/2}$ growth of VRA lies between the logarithmic growth of SFT and the inverse-probability growth of DFT.

The comparison captures the intended compromise. VRA does not promise faster isolated-target acquisition than SFT, which assigns increasingly large gain to low-confidence progress. It reduces this preference while retaining a lower asymptotic acquisition cost than linear probability weighting. Whether the rule improves model-level task performance is the empirical question we turn to next.

5 Experiments: Learning from Gold Demonstrations

We use Qwen3-8B and Qwen3-4B as the main experimental settings, reporting the complete benchmark breakdown for SFT, DFT, InfoSFT, ProFit, and VRA. A separate, compact comparison examines average mean@8 on four additional models. This organization first establishes the detailed performance of the weighting rule and then examines whether its aggregate benefit extends beyond the main backbones.

Table 2: **Main experiment: gold-data fine-tuning on Qwen3-8B and Qwen3-4B.** Mean@8 on five mathematics benchmarks and their macro-average. Base denotes the pretrained model. Bold marks the highest AVG for each model.

Model	Method	AIME24	MATH-500	Minerva	Olympiad	AMC23	AVG ↑
Qwen3-8B	Base	8.34	58.10	23.48	26.62	38.44	31.00
	SFT	4.58	60.23	22.18	22.36	37.19	29.30
	DFT	5.84	69.00	27.98	33.41	48.44	36.93
	InfoSFT	6.25	68.05	26.10	31.08	48.44	35.98
	ProFit	7.51	68.58	27.71	32.88	45.63	36.46
	VRA	8.75	68.53	31.60	33.54	53.75	**39.23**
Qwen3-4B	Base	4.99	37.65	12.84	17.24	25.31	19.61
	SFT	2.49	48.68	14.85	16.20	24.38	21.32
	DFT	2.48	55.80	29.48	22.01	30.31	28.02
	InfoSFT	3.33	57.33	25.74	25.33	28.44	28.03
	ProFit	1.24	57.45	26.98	24.19	30.00	27.97
	VRA	5.84	58.17	24.65	25.10	33.44	**29.44**

5.1 Experimental Setup

Main setting. We fine-tune Qwen3-8B and Qwen3-4B (Yang et al., 2025) on a fixed 30K-example subset of NuminaMath-CoT (Li et al., 2024), using gold demonstrations throughout training. The core comparison uses SFT, DFT (Wu et al., 2026), and VRA without a reference-policy KL term. InfoSFT (Sabbaghi et al., 2026) uses calibration $\bar{p} = 0.93$; ProFit (Liu et al., 2026) supplies the hard-selection comparison. The training configuration is given in Appendix F. We implement VRA by applying the detached weight $w_\epsilon(p) = \sqrt{p/(1-p+\epsilon)}$, with $\epsilon = 10^{-3}$, to the target-token log-likelihood. Implementation details and the effect of numerical stabilization are given in Appendices F.1 and B.4. Additional-model results are summarized in Section 5.3.

Evaluation and reporting. The main settings evaluate AIME 2024 (Mathematical Association of America, 2024), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), and AMC 2023 (Mathematical Association of America, 2023). Mean@8 averages correctness over eight generated answers per problem; AVG is the unweighted mean across the five benchmarks.

5.2 Main Results

Improvement over likelihood and probability weighting. On Qwen3-8B, VRA reaches 39.23 AVG mean@8, compared with 31.00 for the base model and 29.30 for SFT (Table 2). Thus ordinary likelihood fine-tuning reduces the AVG by 1.70 points in this setting, whereas VRA improves it by 8.23 points. Relative to the trained baselines, VRA exceeds SFT by 9.93 points, DFT by 2.30 points, InfoSFT by 3.25 points, and ProFit by 2.77 points. On Qwen3-4B, VRA likewise has the highest AVG, 29.44, compared with 28.03 for InfoSFT, 28.02 for DFT, 27.97 for ProFit, and 21.32 for SFT. These comparisons support the practical value of equal gain against both smooth probability weighting and hard token selection.

Where the benchmark gains appear. On Qwen3-8B, VRA has higher mean@8 than SFT and InfoSFT on all five benchmarks. Against InfoSFT, the largest differences are on Minerva Math and AMC23, at 5.50 and 5.31 points, followed by AIME24 at 2.50 points and OlympiadBench at 2.46 points. The MATH-500 difference is smaller, at 0.48 points. Compared with DFT, VRA leads on four of five benchmarks; DFT retains the higher MATH-500 score, 69.00 versus 68.53. Against ProFit, VRA leads on four of five benchmarks; ProFit scores 0.05 points higher on MATH-500. VRA's largest margin over ProFit is 8.12 points, on AMC23.

5.3 Additional Model Comparisons

Table 3 reports average mean@8 for Qwen2.5-7B (Yang et al., 2024), MiMo-7B (Xiaomi LLM-Core Team, 2025), OLMo3-7B (Team Olmo et al., 2025), and Ling 3.0 Tiny 8B (Ling Team, 2026). All runs use the shared gold-data configuration of Appendix F. The table summarizes the same SFT/DFT/VRA comparison at the aggregate level, leaving the detailed benchmark analysis to the main Qwen3 settings.

Table 3: **Additional-model results.** Average mean@8 percentages; higher is better. Bold marks the highest fine-tuned score in each row. Base denotes the pretrained model.

Model	Base	SFT	DFT	VRA ↑
Qwen2.5-7B	19.94	24.52	31.84	**31.99**
MiMo-7B	13.30	21.37	21.59	**24.56**
OLMo3-7B	4.82	13.56	19.34	**22.58**
Ling 3.0 Tiny 8B	36.16	28.36	38.45	**40.36**

VRA has the highest mean@8 among the fine-tuned methods on all four additional models. Its margins over DFT are 0.15 points on Qwen2.5-7B, 2.97 on MiMo-7B, 3.24 on OLMo3-7B, and 1.91 on Ling 3.0 Tiny 8B. On Ling, VRA also exceeds SFT by 12.00 points. These aggregate results extend the benefit beyond the main Qwen3 settings, with gains that vary across backbones.

Appendix D extends the evaluation to on-policy training, with coefficient and rollout ablations and code transfer, and to KL regularization.

6 Understanding the Gains on Gold Data

The benchmark results support the value of the weighting rule at the task level. We now examine where its different treatment of statistical progress acts on gold demonstrations. The evidence combines an initial target-probability census, matched one-step updates that measure the realized redistribution, and a case study comparing the models' responses to the same benchmark problem. Complementary full-run acquisition and held-out response-structure measurements are reported in Appendices E.2 and E.3.

6.1 Where Gold Targets Lie on the Probability Axis

We score a fixed 4,000-example subset of the NuminaMath-CoT training data with the Qwen3-8B base model in the training format, obtaining 1.72M target tokens. The census measures the distribution of demonstrated targets encountered in training; it is not an independent generalization test. Scoring and aggregation details appear in Appendix E.1.

Low-probability targets are a minority with a disproportionate nominal SFT signal. Targets with $p < 0.1$ account for 6.11% of the census, but 32.89% of the summed unit-weight target-logit signal $1 - p$; at $p < 0.5$, the corresponding shares are 16.58% and 73.97%. Figure 3 makes the alternative allocations explicit. The $p < 0.1$ region receives 3.52% of DFT's normalized signal, 8.71% of InfoSFT's, none of ProFit's, and 6.01% of VRA's. Conversely, the $p \geq 0.9$ region receives 4.04% under SFT and 25.58% under VRA. On the same census, the empirical 95th-to-5th percentile ratio of C_{SFT} is approximately 967, showing that the preference identified in Section 3 spans a broad range in actual gold data.

The low-probability region includes targets on which the rules strongly disagree, but probability alone does not identify their learning value. The high-confidence tail is also substantial: 45.57% of tokens have $p > 0.996$. VRA's 25.58% share of the $p \geq 0.9$ region, well below the region's token mass, follows mainly from its target-logit signal $\sigma(p) = \sqrt{p(1-p)}$, which vanishes as $p \to 1$ even under equal gain; stabilization further attenuates only the $p > 0.999$ tail.

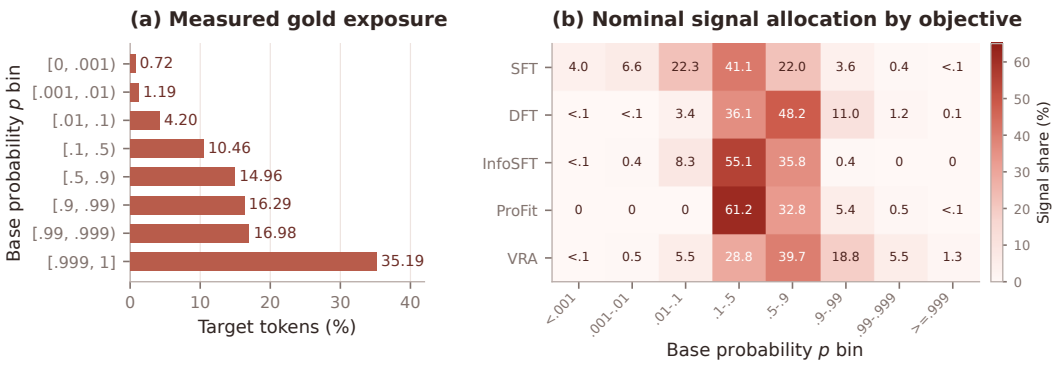


Figure 3: **Gold targets and the nominal signals acting on them.** Left: measured token shares in eight probability bins for the Qwen3-8B base gold census. Bars show mass per bin, not probability density. Right: within-method shares of the nominal direct target-logit signal $s_w(p) = w(p)(1-p)$ in the same bins; every row sums to 100%. VRA uses $\epsilon = 10^{-3}$; InfoSFT is plotted with $\bar{p} = 0.93$ and ProFit with $\tau = 0.1$. Shares describe objective-level allocation on the teacher-forced census and do not compare absolute scales across methods.

6.2 From Nominal Allocation to Realized Updates

The shares above are properties of the objectives. Between them and the realized update stand AdamW's per-parameter normalization and global gradient clipping, so the nominal allocation need not survive training. We test this with matched one-step updates. From two checkpoints of the SFT run (steps 100 and 300), with the optimizer state restored, we take one step with unit weights and one with VRA weights on the same batch, repeat this for three batches, and rescore the census after each step. We compare the two updates by the signed Fisher–Rao change of each target, grouped by its probability before the step and averaged over prompts.

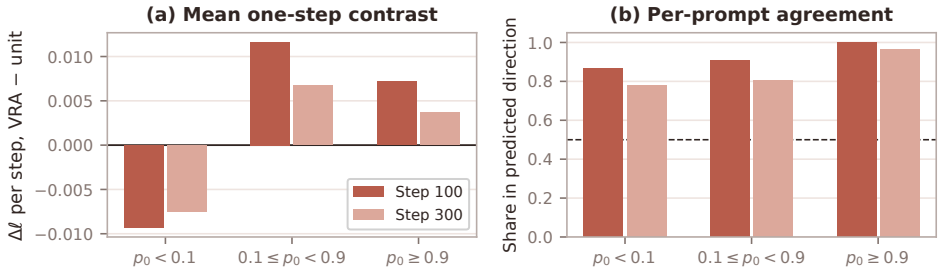


Figure 4: **Matched one-step updates on Qwen3-8B.** (a) Signed Fisher–Rao change of census targets after one VRA step minus one unit-weight step, from two SFT checkpoints, averaged over three batches. (b) Fraction of prompts moving in the predicted direction: negative in the low region, positive in the middle and high regions; the dashed line marks one half. All six comparisons show the $(-,+,+)$ redistribution; per-batch values are in Table 10.

All six comparisons show the predicted redistribution (Figure 4): relative to unit weighting, one VRA step moves low-probability targets less and middle- and high-probability targets more. In every comparison both gradients exceed the global clipping norm of 1.0, and the VRA gradient is four to five times larger before clipping. Clipping thus removes the difference in raw gradient size, so the contrast reflects gradient direction, and the redistribution persists through AdamW. Nor is the pattern driven by a few prompts: in every comparison, 77–100% of prompts with targets in a region move in the predicted direction (Figure 4b). All six comparisons come from one training run and are measured on training data, so we read them as evidence about direction. Appendix E.1 gives the protocol.

Problem.

Let set $S = \{1, 2, 3, 4, 5, 6\}$, and let set T be the set of all subsets of S (including the empty set and S itself). Let t_1, t_2, t_3 be elements of T, not necessarily distinct. The ordered triple (t_1, t_2, t_3) is called satisfactory if either
(a) both $t_1 \subseteq t_3$ and $t_2 \subseteq t_3$, or
(b) $t_3 \subseteq t_1$ and $t_3 \subseteq t_2$.
Compute the number of satisfactory ordered triples (t_1, t_2, t_3). *Reference answer:* 31186

Base ✘

2. For condition (b), t_3 must be a subset of both t_1 and t_2.
• Choosing t_3: There are 64 choices.
• For each choice of t_3, there are $2^{6-|t_3|}$ possible ways to choose t_1 and t_2, because t_1 and t_2 must both include t_3 and can contain any subset of the remaining elements in S:

$$\sum_{i=0}^{6} 2^{6-i} = 1 + 2 + 4 + 8 + 16 + 32 + 64 = 127$$

So, there are $64 \times 127 = 8128$ triples satisfying condition (b).

… *Answer:* 15360

InfoSFT ✘

Step 3: Analyzing condition (b)
• If condition (b) holds, then t_3 must be a subset of both t_1 and t_2. For any subset t_3, there are $2^{|t_3|}$ subsets that contain t_3.
• Therefore, for each t_3, there are $(2^{|t_3|})^2$ ways to choose t_1 and t_2 such that condition (b) is satisfied.
• Summing over all possible t_3, we get the total number of ordered triples satisfying condition (b) as $\sum_{i=0}^{6} \binom{6}{i}(2^i)^2$.

… *Answer:* 34357

SFT ✘

Calculations for $|t_3| = k$:
• Consider the $\binom{64}{k}$ subsets t_3.
• Case (a): $t_1, t_2 \subseteq t_3$ with $\binom{2^k}{2}$ pairs or when $t_1 = t_2 = t_3$, there are 2^k options: $\binom{2^k}{2} + 2^k$.

… *Answer:* 19338891

ProFit ✘

5. Combining both cases:
• The total number of satisfactory ordered triples is the sum of the counts from both cases:

$$15625 + 15625 = 31250$$

… *Answer:* 31250

DFT ✘

4. Combining the Results:
Since the two cases are mutually exclusive, the total number of satisfactory triples is the sum of the number of triples in each case:

$$15625 + 15625 = 31250$$

… *Answer:* 31250

VRA ✔

3. Overlap Consideration: The overlap occurs when $t_1 = t_2 = t_3$. There are 64 such triples.
4. Total Count: Adding the counts from both cases and subtracting the overlap:

$$15625 + 15625 - 64 = 31186.$$

… *Answer:* 31186

Figure 5: **One problem, six model responses.** Responses to OlympiadBench problem 553. Brick red marks errors; green marks the overlap correction. Symbols indicate final-answer correctness; ellipses mark omitted continuations.

6.3 Case Study: A Missing Intersection

Figure 5 compares the base model and five fine-tuned Qwen3-8B models on OlympiadBench problem 553. DFT and ProFit obtain both individual counts but omit their intersection; VRA subtracts the 64 overlapping triples to obtain $5^6 + 5^6 - 2^6 = 31186$.

7 Conclusion

We propose equal gain per unit Bernoulli Fisher–Rao distance as an explicit principle for token weighting and show that it uniquely determines the normalized Vocab Relative Advantage weight. The stabilized rule improves average mean@8 over SFT and DFT on all six backbones, and over InfoSFT and ProFit in the main Qwen3-8B and Qwen3-4B comparisons. Gold-data measurements characterize the objectives' nominal signal allocation across confidence regions, and matched one-step comparisons on training data show that the direction of VRA's reallocation relative to unit weighting persists through AdamW and gradient clipping. These findings support the practical value of the rule, while its contribution to specific output behaviors remains an open empirical question.

AI Use Statement

We used generative AI tools to aid and polish the writing of this paper, including assistance with writing up the derivation in Appendix C from the authors' own derivation, which the authors verified step by step. We did not use generative AI tools to develop the equal-gain principle or the theoretical framework, formulate mathematical claims, provide critical ingredients for proofs, propose or refine hypotheses, design experiments, implement methods, generate or process data, assist with translation, or support qualitative or thematic data analysis. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text produced with the aid of generative AI.

Reproducibility Statement

The ideal token weight and scalar reference dynamics are specified in Section 4. The implemented loss, masking, and reduction are given in Appendix F.1, with stabilization properties in Appendix B.4. Complete proofs appear in Appendix B; experimental settings appear in Appendix F. Figures 1 and 2 are generated from the displayed formulas. The gold census and matched one-step updates in Section 6 follow the procedures in Appendix E.1. The main results use the mathematics benchmarks described in Section 5; supplementary training regimes, ablations, and code evaluation are specified in Appendix D.

References

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2): 251–276, 1998. doi: 10.1162/089976698300017746. URL https://doi.org/10.1162/089976698300017746.

Nihat Ay. On the locality of the natural gradient for learning in deep Bayesian networks. *Information Geometry*, 6:1–49, 2023. doi: 10.1007/s41884-020-00038-y. URL https://link.springer.com/article/10.1007/s41884-020-00038-y. First published online in 2020.

Nihat Ay, Jürgen Jost, Hông Vân Lê, and Lorenz Schwachhöfer. *Information Geometry*, volume 64 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge*. Springer, 2017.

N. N. Čencov. *Statistical Decision Rules and Optimal Inference*, volume 53 of *Translations of Mathematical Monographs*. American Mathematical Society, 1982.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/7017-deep-reinforcement-learning-from-human-preferences.pdf.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10818–10838, 2025. URL https://proceedings.mlr.press/v267/chu25c.html.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. URL https://www.jmlr.org/papers/v25/23-0870.html.

Yaoyou Fan, Chao Zhang, Xiaoyu Tan, Chenxing Sun, Yu Yuan, Haoyu Feng, Lu Pan, Ke Zeng, and Xunliang Cai. Navigating the pareto frontier of alignment: Spectrum-adaptive fine-tuning for LLMs. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://github.com/sjtu-scx/SAFT/blob/main/SAFT.pdf.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. DeepSeek-Coder: When the large language model meets programming—the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024. URL https://arxiv.org/abs/2401.14196.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, 2025.

Xue Jiang, Yihong Dong, Mengyang Liu, Hongyi Deng, Tian Wang, Yongding Tao, Zhi Jin, Wenpin Jiao, and Ge Li. CODERL+: Improving code generation via reinforcement with execution semantics alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3608–3622. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.164.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath. Hugging Face repository, 2024. URL https://huggingface.co/datasets/AI-MO/NuminaMath-CoT.

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

Ling Team. Ling-3.0-tiny-base. Hugging Face repository, 2026. URL https://huggingface.co/inclusionAI/Ling-3.0-tiny-base.

Tao Liu, Taiqiang Wu, Runming Yang, Shaoning Sun, Junjie Wang, and Yujiu Yang. ProFit: Leveraging high-value signals in SFT via probability-guided token selection. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15383–15401, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.755. URL https://aclanthology.org/2026.findings-acl.755/.

Mathematical Association of America. 2023 AMC 12A and AMC 12B. American Mathematics Competitions, 2023.

Mathematical Association of America. 2024 American Invitational Mathematics Examination (AIME) I and II. American Mathematics Competitions, 2024.

Rui Ming, Haoyuan Wu, Shoubo Hu, Zhuolun He, and Bei Yu. One-token rollout: Guiding supervised fine-tuning of LLMs with policy gradient. *arXiv preprint arXiv:2509.26313*, 2025.

Long Ouyang, Jeff 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 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. URL https://proceedings.neurips.cc/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Richard Yuanzhe Pang and He He. Text generation by learning from demonstrations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=RovX-uQ1Hua.

Chongli Qin and Jost Tobias Springenberg. Supervised fine tuning on curated data is reinforcement learning (and can be improved). *arXiv preprint arXiv:2507.12856*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Mahdi Sabbaghi, George Pappas, Adel Javanmard, and Hamed Hassani. InfoSFT: Learn more and forget less with information-aware token weighting. *arXiv preprint arXiv:2605.14967*, 2026. doi: 10.48550/arXiv.2605.14967. URL https://arxiv.org/abs/2605.14967.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2110.08207.

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.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Ke Wang, Shuangqi Li, Mathieu Salzmann, and Pascal Frossard. PriFT: Prior-support guided supervised fine-tuning. *arXiv preprint arXiv:2606.09396*, 2026.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2109.01652.

Yongliang Wu, Yizhou Zhou, Ziheng Zhou, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the generalization of SFT: A reinforcement learning perspective with reward rectification. In *International Conference on Learning Representations*, 2026.

Xiaomi LLM-Core Team. MiMo: Unlocking the reasoning potential of language model – from pretraining to posttraining. *arXiv preprint arXiv:2505.07608*, 2025.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. A minimalist approach to LLM reasoning: from rejection sampling to Reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ac662d74829e4407ce1d126477f4a03a-Abstract-Conference.html.

He Zhu, Junyou Su, Peng Lai, Ren Ma, Wenjia Zhang, Linyi Yang, and Guanhua Chen. Anchored supervised fine-tuning. In *International Conference on Learning Representations*, 2026a.

Wenhong Zhu, Ruobing Xie, Rui Wang, Xingwu Sun, Di Wang, and Pengfei Liu. Proximal supervised fine-tuning. In *International Conference on Learning Representations*, 2026b.

A Related Work

Learning from fixed demonstrations. Several methods improve learning from fixed demonstrations by connecting supervised objectives to reinforcement learning. GOLD combines approximate probability-based importance weights with returns on demonstrated trajectories in an offline policy-gradient formulation (Pang & He, 2021). Importance-weighted SFT relates learning from curated data to a lower bound on a sparse-reward RL objective and introduces trajectory-level importance weights to tighten that bound (Qin & Springenberg, 2025). DFT examines the implicit reward obtained by expressing the SFT gradient as a policy gradient and rectifies it through detached target-probability weighting (Wu et al., 2026). We build on this objective-design perspective by comparing local incentives in a common unit: objective gain per unit of Fisher–Rao distance toward the demonstrated token.

Confidence weighting and reference regularization. Recent methods make different choices about which target probabilities should receive the strongest learning signal. InfoSFT derives information-aware token weights from population fitting under a KL budget to balance acquisition and retention (Sabbaghi et al., 2026). ProFit selects tokens above a probability threshold, removing the direct supervised signal from lower-probability targets (Liu et al., 2026). Spectrum-Adaptive Fine-Tuning (SAFT) studies a family of probability-dependent weights spanning SFT and DFT, with geometric and harmonic variants that trade learning efficiency against noise robustness (Fan et al., 2026). The square-root member of SAFT's geometric family shares VRA's low-confidence scaling, but does not maintain constant FR gain across the full probability range. These methods prescribe different confidence preferences. Our framework compares those preferences in common statistical units and derives a weight with confidence-independent marginal gain.

Reference-based methods address the additional issue of policy drift. ASFT combines DFT-style weighting with a KL penalty anchored to the base model (Zhu et al., 2026a), while Proximal SFT uses a reference-relative clipped objective inspired by trust-region policy optimization (Zhu et al., 2026b). PriFT instead computes token weights from a frozen pretrained reference rather than from the model being fine-tuned (Wang et al., 2026). These methods use a reference distribution to constrain departure or to set token weights, whereas the equal-gain principle specifies the value assigned to local progress toward a target. Our scalar profiles describe weights determined by the current target probability; reference-dependent terms require separate treatment. We report reference-regularized comparisons in Appendix D.6.

Information geometry and natural-gradient learning. The Fisher–Rao metric supplies an established geometry for probability distributions. Invariance under congruent Markov embeddings characterizes it up to a common scale across finite probability simplices (Čencov, 1982; Ay et al., 2017). Natural-gradient methods use the Fisher information matrix to define steepest updates in this geometry (Amari, 1998). We use the metric to compare token objectives on the Bernoulli distribution that distinguishes the target from the rest of the vocabulary. Equal gain is an additional design principle, not a consequence of metric invariance alone. VRA applies detached scalar weights with a standard optimizer, rather than computing a full-model natural gradient.

Local policy gradients and on-policy learning. Relative-advantage methods provide another connection between reweighting and policy optimization. GRPO estimates relative advantages by normalizing rewards within groups of sampled responses and uses them in a clipped policy objective (Shao et al., 2024). One-Token Rollout (OTR) instead samples alternative next tokens at demonstration prefixes and assigns rewards according to agreement with the demonstrated token (Ming et al., 2025). Our policy-gradient reading of VRA (Appendix C) also uses a local target-match reward. Its population mean and variance are available analytically, so the standardized score-function expectation reduces exactly to the VRA weight without sampling alternative tokens. The reading is local to a fixed context and target, and its population normalization differs from OTR's sampled objective and GRPO's response groups.

For learning from accepted rollouts, the RAFT formulation studied by Xiong et al. (2025) maximizes their likelihood; RAFT++ adds importance correction and clipping when reusing sampled responses. Our on-policy variant replaces unit weighting on verifier-accepted targets with the same local VRA

rule. Appendix D.1 separately reports this variant and weight substitutions within the RAFT and RAFT++ recipes.

B Theoretical Details

This appendix specifies the statistical object measured by the progress profile, proves the equal-gain characterization, and derives the single-logit learning times. All local statements fix a context s and a target token $a^\star$, with $p = \pi_\theta(a^\star \mid s) \in (0, 1)$. External multipliers and calibration parameters are held fixed when a local profile is evaluated.

B.1 Reward Coarsening and the Statistical Metric

The statistic $r(a) = \mathbf{1}[a = a^\star]$ maps the vocabulary distribution to Bernoulli(p). This coarsening retains the probability of the observed target and discards distinctions among non-target tokens. Consequently, our distance measures progress in the target-versus-rest distribution. A change among non-target probabilities can move the full vocabulary distribution while leaving this distance zero.

The canonical-metric argument concerns a compatible family of metrics on all finite probability simplices. Specifically, consider smooth Riemannian metrics that contract under stochastic maps between finite sample spaces, including refinements and their sufficient-statistic inverses whenever the distributions remain in the interiors. Refinement followed by its inverse then preserves the metric. The classical characterization identifies this family with the Fisher–Rao metric up to a common positive scale; see Čencov (1982) and Ay et al. (2017). This is stronger than coordinate invariance alone, or contraction under maps between binary sample spaces alone. We use the usual Fisher-information normalization.

Restricting this metric to the Bernoulli family gives

$$d\ell^2 = \frac{dp^2}{p} + \frac{(-dp)^2}{1-p} = \frac{dp^2}{p(1-p)}, \qquad \ell(p) = 2 \arcsin \sqrt{p}.$$

Thus the Fisher–Rao distance between two Bernoulli states is $|\ell(p_2) - \ell(p_1)|$. Stochastic-map contraction ensures that coarsening does not enlarge the local statistical distance. It does not identify the coarsened distance with every change in the original vocabulary distribution.

B.2 A Utility Whose Differential Matches Detached Weighting

A detached weighted log-likelihood has a well-defined local utility gradient. Let $w : (0, 1) \to [0, \infty)$ be continuous, or piecewise continuous away from finitely many thresholds, and choose an interior reference point p_{r}. Define

$$U_w(p) = \int_{p_{\mathrm{r}}}^{p} \frac{w(u)}{u}\, du.$$

Where the derivative exists,

$$\nabla_\theta U_w(p) = w(p) \nabla_\theta \log p = -\nabla_\theta \big[- \operatorname{sg}(w(p)) \log p \big].$$

The equality describes gradients at the current state. The numerical value of the detached weighted loss is generally not $-U_w(p)$, and differentiating $w(p) \log p$ without detachment would add an unwanted term. If a rule depends on a calibration statistic that changes across training steps, this construction is its conditional local utility with that statistic fixed; it need not define a single global potential for the joint training process.

Proposition 2 (Three equivalent readings of the progress profile)**.** *For the positively oriented Bernoulli arc-length coordinate, the local marginal utility per unit statistical distance is*

$$C_w(p) = \frac{dU_w}{d\ell} = w(p) \sqrt{\frac{1-p}{p}}.$$

Equivalently, it is half the derivative with respect to the angle of the square-root embedding, or the target-logit signal divided by the Bernoulli standard deviation.

Proof. First, the chain rule gives

$$\frac{dU_w}{d\ell} = \frac{w(p)/p}{1/\sqrt{p(1-p)}} = w(p) \sqrt{\frac{1-p}{p}}.$$

Second, the square-root embedding $v(p) = (\sqrt{1-p}, \sqrt{p}) = (\cos\alpha, \sin\alpha)$ lies on the unit quarter circle with $\alpha = \arcsin\sqrt{p}$. Its Euclidean arc length is $d\alpha$, whereas the Fisher–Rao arc length is $d\ell = 2\,d\alpha$. Hence $C_w = \frac{1}{2}\,dU_w/d\alpha$; omitting this factor of two would change the normalization. Third, holding the other softmax logits fixed, the target-logit signal is

$$s_w(p) = w(p)\frac{\partial \log p}{\partial z_{a^\star}} = w(p)(1-p).$$

With $\sigma(p) = \sqrt{p(1-p)}$, we have $s_w(p)/\sigma(p) = C_w(p)$. □

The ratio $dU_w/d\ell$ is unchanged by a smooth reparameterization of the same Bernoulli state. A profile can therefore be parameterization invariant without being constant across probabilities. Its value is an objective's marginal evaluation of progress, rather than the statistical distance traversed by one training step.

B.3 Endpoint Profiles and the Equal-Gain Characterization

The endpoint behavior follows directly from the common metric. For unit weighting and DFT,

$$C_{\text{unit}}(p) = \sqrt{\frac{1-p}{p}}, \qquad C_{\text{DFT}}(p) = \sqrt{p(1-p)}.$$

The first diverges as $p \downarrow 0$; the second vanishes at both endpoints. For the InfoSFT rule with a fixed calibration probability $\bar{p} \in (0,1)$, write $a = \operatorname{logit}\bar{p}$ and

$$w_{\text{Info}}(p) = p[a - \operatorname{logit} p]_+, \qquad C_{\text{Info}}(p) = \sqrt{p(1-p)}[a - \operatorname{logit} p]_+.$$

Then $C_{\text{Info}}(p) \sim \sqrt{p}\log(1/p) \to 0$ as $p \downarrow 0$, and the profile is zero for $p \geq \bar{p}$. These are endpoint statements; the finite-probability comparison depends on the actual calibration and overall scale. A threshold rule $w(p) = \mathbf{1}[p > \tau]$, for $\tau \in (0,1)$, instead sets the entire interval $(0, \tau]$ to zero.

Proposition 3 (Unique normalized equal-gain weight)**.** *Suppose a nonnegative weight satisfies $C_w(p) = c$ for every $p \in (0,1)$ and $w(1/2) = 1$. Then $c = 1$, and its unique value at every interior state is*

$$w(p) = \sqrt{\frac{p}{1-p}}.$$

Conversely, this weight satisfies both conditions. Its utility is $U_w(p) = 2\arcsin\sqrt{p} + \text{constant}$.

Proof. At $p = 1/2$, the profile equals the weight, so $c = 1$. Solving $w(p)\sqrt{(1-p)/p} = 1$ gives the claimed expression pointwise. Its utility derivative is $1/\sqrt{p(1-p)}$, whose antiderivative is the stated arcsine coordinate. □

The uniqueness is conditional on the equal-gain principle. The weaker requirement that a low-probability profile be finite and nonzero permits other weights. For example, $w(p) = \sqrt{2p}$ has $w(1/2) = 1$ and $C_w(p) = \sqrt{2(1-p)}$; it meets that endpoint requirement but is not equal-gain. The task motivation supports the design principle without turning it into an optimality theorem for every data distribution.

B.4 Numerical Stabilization and the Local Logit Bound

Proposition 4 (Stabilized profile and gradient)**.** *For $\epsilon > 0$, define*

$$w_\epsilon(p) = \sqrt{\frac{p}{1-p+\epsilon}}.$$

Its profile, local utility, and target-logit ascent signal are

$$C_{w_\epsilon}(p) = \sqrt{\frac{1-p}{1-p+\epsilon}},$$

$$U_{w_\epsilon}(p) = 2\arcsin\sqrt{\frac{p}{1+\epsilon}} + \textit{constant},$$

$$s_{w_\epsilon}(p) = \sqrt{p(1-p)}\sqrt{\frac{1-p}{1-p+\epsilon}} \leq \sqrt{p(1-p)} \leq \frac{1}{2}.$$

Proof. Substitution in Proposition 2 gives the first expression. Differentiating the stated utility gives $1/\sqrt{p(1-p+\epsilon)} = w_\epsilon(p)/p$. The final identity follows from $s_w = w(1-p)$; the first square-root correction is at most one, and $p(1-p) \le 1/4$. □

Stabilization therefore preserves an approximately constant profile away from the extreme high-confidence region, while making the profile vanish as $p \uparrow 1$. In particular, $C_{w_\epsilon}(p) \to 1/\sqrt{1+\epsilon}$ at the low-confidence endpoint. The implemented weight has $w_\epsilon(1/2) = 1/\sqrt{1+2\epsilon}$, so the normalization in Proposition 3 applies exactly to the ideal weight. The bound above is on a local logit derivative; it does not assert a bound on a shared-parameter update.

B.5 Learning Times in a Single-Logit Model

The learning-time comparison uses an explicit one-dimensional dynamics. Let $p = \mathrm{sigmoid}(z)$, hold the competing logit fixed, and perform Euclidean gradient ascent on U_w with unit time scale. Then

$$\dot{z} = w(p)(1-p), \qquad \dot{p} = w(p)p(1-p)^2.$$

For $0 < p_0 < p_\star < 1$, the time to reach $p_\star$, whenever the weight is positive over this interval, is

$$T_w(p_0, p_\star) = \int_{p_0}^{p_\star} \frac{dp}{w(p)p(1-p)^2}.$$

The target $p_\star$ is fixed as $p_0 \downarrow 0$. Constant rescaling of the weight or the learning-rate time unit rescales all corresponding times.

Proposition 5 (Closed-form hitting times)**.** *For unit weighting, DFT, and ideal VRA, write $T_w(p_0, p_\star) = F_w(p_\star) - F_w(p_0)$, where*

$$F_{\text{unit}}(p) = \log\frac{p}{1-p} + \frac{1}{1-p},$$
$$F_{\text{DFT}}(p) = 2\log\frac{p}{1-p} - \frac{1}{p} + \frac{1}{1-p},$$
$$F_{\text{VRA}}(p) = \frac{2(2p-1)}{\sqrt{p(1-p)}}.$$

Consequently,

$$T_{\text{unit}}(p_0, p_\star) = \log(1/p_0) + O(1),$$
$$T_{\text{DFT}}(p_0, p_\star) = p_0^{-1} + 2\log(1/p_0) + O(1),$$
$$T_{\text{VRA}}(p_0, p_\star) = 2p_0^{-1/2} + O(1).$$

Proof. Differentiation of the three expressions gives respectively $1/[p(1-p)^2]$, $1/[p^2(1-p)^2]$, and $1/[p^{3/2}(1-p)^{3/2}]$, which are the hitting-time integrands. Expanding the antiderivatives at $p_0 = 0$ proves the asymptotic expressions. □

For InfoSFT with fixed $\bar{p}$ and $p_\star < \bar{p}$, the corresponding integral is

$$T_{\text{Info}}(p_0, p_\star) = \int_{p_0}^{p_\star} \frac{dp}{p^2(1-p)^2[a - \operatorname{logit} p]}.$$

Since $a - \operatorname{logit} p \sim \log(1/p)$ near zero,

$$T_{\text{Info}}(p_0, p_\star) \sim \frac{1}{p_0 \log(1/p_0)}.$$

For completeness, the derivative of $1/[p\log(1/p)]$ is $-(\log(1/p)-1)/[p^2\log^2(1/p)]$; comparison with the derivative of the integral with respect to its lower limit proves the equivalence by l'Hôpital's rule. Near the upper threshold,

$$\dot{p} \sim \bar{p}(1-\bar{p})(\bar{p}-p) \qquad (p \uparrow \bar{p}).$$

Thus the continuous flow approaches $\bar{p}$ only asymptotically and cannot cross it. A trajectory initialized at or above $\bar{p}$ is stationary under this local fitting term. These conclusions hold for a fixed calibration threshold; a changing threshold is a different dynamics.

For ProFit's strict mask $w(p) = \mathbf{1}[p > \tau]$, with $0 < \tau < p_\star < 1$, the zero-update region and the SFT flow above the threshold give

$$T_{\text{ProFit}}(p_0, p_\star) = \begin{cases} \infty, & 0 < p_0 \le \tau, \\ T_{\text{SFT}}(p_0, p_\star), & \tau < p_0 < p_\star. \end{cases}$$

The stationary result is local: in a shared language model, other targets' updates can change a masked target's probability.

The times quantify a specific tradeoff: ideal VRA slows the unit-weight low-probability flow while retaining a smaller asymptotic learning time than DFT or the fixed-threshold InfoSFT flow. They do not rank task performance or describe full-model training. Unlike the profile definition, these times depend on the stated Euclidean logit dynamics.

C From DFT to VRA: A Policy-Gradient Reading

The main text obtains VRA from equal gain per unit Fisher–Rao distance. The same weight also admits a policy-gradient reading, which is the source of its name: we start from the policy-gradient view of SFT in DFT (Wu et al., 2026), standardize its reward in the manner of GRPO (Shao et al., 2024), and evaluate the resulting expectation over the vocabulary in closed form.

Setting. Fix a prefix $s = (x, y^\star_{<t})$ and its target $a^\star = y^\star_t$, with $p = \pi_\theta(a^\star \mid s)$. In gold training the prefix is the teacher-forced demonstration; in the on-policy variant of Appendix D.1 it comes from a verifier-accepted rollout. Only the next token is treated as an action. Draw $a \sim \pi_\theta(\cdot \mid s)$ and define the target-match reward $r(a) = \mathbf{1}[a = a^\star]$. Under the current policy, r is Bernoulli(p), with mean p and standard deviation $\sigma(p) = \sqrt{p(1-p)}$.

The DFT view of SFT. DFT writes the SFT gradient as an on-policy policy gradient with the target-match reward and an importance weight $1/\pi_\theta$, which becomes large when the target is unlikely (Wu et al., 2026). At a single token position,

$$\nabla_\theta \log p = \mathbb{E}_{a\sim\pi_\theta}\left[\frac{r(a)}{\pi_\theta(a \mid s)} \nabla_\theta \log \pi_\theta(a \mid s)\right].$$

Multiplying by the detached probability $\text{sg}(p)$ cancels the importance weight and leaves the policy gradient of the target-match reward itself,

$$g_{\text{DFT}} = \mathbb{E}_{a\sim\pi_\theta}[r(a)\, \nabla_\theta \log \pi_\theta(a \mid s)] = p\, \nabla_\theta \log p = \nabla_\theta p,$$

so every demonstrated target receives the same unit reward.

Standardizing the reward over the vocabulary. DFT likens this uniform reward to verification-based reinforcement learning (Wu et al., 2026), where GRPO further standardizes rewards within a group of sampled responses (Shao et al., 2024). At a single token position, the whole next-token distribution can serve as the group. Its reward mean and standard deviation are p and $\sigma(p)$, so the standardized advantage is available in closed form, without sampling:

$$A(a) = \frac{r(a) - p}{\sqrt{p(1-p)}}, \qquad A_+ = \sqrt{\frac{1-p}{p}} \quad (a = a^\star), \qquad A_- = -\sqrt{\frac{p}{1-p}} \quad (a \neq a^\star).$$

All non-target tokens share the advantage A_- because all of them receive zero reward.

The vocabulary expectation. With the advantage detached, the standardized policy gradient separates into a target term and a non-target term,

$$g_{\text{VRA}} = \mathbb{E}_{a\sim\pi_\theta}[\text{sg}(A(a))\, \nabla_\theta \log \pi_\theta(a \mid s)] = pA_+ \nabla_\theta \log p + A_- \sum_{a\neq a^\star} \pi_\theta(a \mid s) \nabla_\theta \log \pi_\theta(a \mid s).$$

The score identity $\sum_a \pi_\theta(a \mid s) \nabla_\theta \log \pi_\theta(a \mid s) = \nabla_\theta \sum_a \pi_\theta(a \mid s) = 0$ turns the non-target sum into $-p\, \nabla_\theta \log p$. Since $A_+ - A_- = 1/\sqrt{p(1-p)}$,

$$g_{\text{VRA}} = p\,(A_+ - A_-)\, \nabla_\theta \log p = \sqrt{\frac{p}{1-p}}\, \nabla_\theta \log p.$$

The target and non-target terms contribute $\sqrt{p(1-p)}$ and $\sqrt{p^3/(1-p)}$ to this coefficient; Appendix D.2 trains with each term alone. The full expectation is therefore reproduced by a cross-entropy loss on the target token alone, $-\,\mathrm{sg}\big(\sqrt{p/(1-p)}\big)\log p$, whose stabilized form is the implemented weight w_ϵ of Appendix F.1. Because the advantage is formed relative to the vocabulary distribution rather than to a group of sampled responses, we call the rule Vocab Relative Advantage.

Relation to equal gain. At a single token position the vocabulary probabilities are available analytically, so the policy gradient expands exactly over the vocabulary. In this expansion the mean subtraction leaves the total weight unchanged; it determines how the weight is shared between the target and non-target terms. The resulting gradient is the DFT gradient rescaled by the target-match standard deviation $\sigma(p)$ of Section 2, $g_{\mathrm{VRA}} = g_{\mathrm{DFT}}/\sigma(p) = \nabla_\theta p/\sigma(p)$. Rescaling a change in p by $1/\sigma(p)$ converts it into Fisher–Rao arc length, $d\ell = dp/\sqrt{p(1-p)}$, so $g_{\mathrm{VRA}} = \nabla_\theta \ell(p)$, the gradient of the equal-gain potential U_{VRA} in Section 4. The policy-gradient reading and the equal-gain principle thus arrive at the same coefficient.

D Additional Training Regimes and Ablations

The main experiments isolate token reweighting on fixed gold demonstrations without a reference-policy penalty. This appendix first applies the same token rule to verifier-accepted on-policy targets (Appendix D.1) and examines that setting through ablations of the coefficient's analytic components, coefficient scale, and rollout count, and through transfer to code (Appendices D.2–D.5); the ablations and the code study use Qwen2.5-7B. It then reports gold-data fine-tuning with reference-based regularization (Appendix D.6). The mathematics tables use the five benchmarks and the mean@8 definition from Section 5, with AVG their unweighted mean.

D.1 Learning from Verifier-Accepted Rollouts

Training objective and baselines. On-policy VRA generates responses with the current model, retains those accepted by a correctness verifier, and applies the same detached token weight to their response-token log-likelihoods. Incorrect responses are masked out. We compare with GRPO (Shao et al., 2024), GSPO (Zheng et al., 2025), and the RAFT/RAFT++ formulations of Xiong et al. (2025). RAFT maximizes likelihood on accepted responses; RAFT++ additionally uses token-level behavior-policy importance ratios and PPO-style clipping when optimizing a rollout buffer. The "w/ DFT" and "w/ VRA" rows replace the token weight within the indicated recipe. They retain that recipe's policy corrections, so a RAFT++ weight substitution is distinct from the default on-policy VRA objective.

Configuration and selection. Qwen2.5-7B and Qwen3-4B/8B train on 17,917 DAPO-Math prompts (Yu et al., 2025). Rewards are binary, obtained by normalized string matching of the last answer enclosed in `\boxed{}`. Training uses verl. Adam uses learning rate 10^{-6}, $\beta = (0.9, 0.999)$, weight decay 0.1, and gradient clipping at 1.0, with a constant schedule after 10 linear-warmup steps. The global batch size is 256 with bfloat16 precision. Each prompt has $N = 8$ responses sampled at temperature 1.0 and top-$p = 1.0$, with maximum response length 2,048. For VRA, the coefficient is $\mathrm{sg}\left(\sqrt{p/(1-p+10^{-3})}\right)$, computed from the current policy, with token-mean aggregation over accepted response tokens in the mini-batch. GRPO centers binary rewards within each group and divides by the population standard deviation (divisor N); homogeneous groups have zero centered reward-gradient coefficients. Each method selects its highest-AVG checkpoint within the common 0–200-step window.

Results and recipe dependence. Default VRA attains the highest AVG at all three model sizes in Table 4, exceeding GRPO by 1.91, 2.89, and 1.71 points on Qwen2.5-7B, Qwen3-4B, and Qwen3-8B, respectively. It exceeds GRPO on 13 of 15 benchmark–model pairs; the exceptions are MATH-500 on Qwen2.5-7B and Minerva Math on Qwen3-8B. Thus the coefficient transfers to accepted on-policy targets. The effect of a weight substitution depends on the surrounding recipe: RAFT++ with VRA reaches 40.27 AVG on Qwen3-8B, below RAFT++ with DFT at 44.59 and default VRA at 48.01.

D.2 Analytic Components

The policy-gradient reading in Appendix C splits the ideal coefficient into a target-action term and an analytically marginalized non-target-action term:

$$w_{\text{pos}}(p) = \sqrt{p(1-p)}, \qquad w_{\text{neg}}(p) = \sqrt{\frac{p^3}{1-p}}, \qquad w_{\text{pos}}(p) + w_{\text{neg}}(p) = \sqrt{\frac{p}{1-p}}.$$

For the implemented weight $w_\epsilon(p) = \sqrt{p/(1-p+\epsilon)}$, with $\epsilon = 10^{-3}$, the corresponding algebra-preserving split is

$$w^\epsilon_{\text{pos}}(p) = (1-p)w_\epsilon(p), \qquad w^\epsilon_{\text{neg}}(p) = p\, w_\epsilon(p), \qquad w^\epsilon_{\text{pos}}(p) + w^\epsilon_{\text{neg}}(p) = w_\epsilon(p).$$

Each ablation replaces the full detached weight by one component while retaining the accepted-response loss. "Negative only" therefore refers to non-target *vocabulary actions*; it does not train on rejected responses.

Neither isolated component matches the full coefficient's AVG: positive-only and negative-only training trail by 2.57 and 2.39 points, respectively (Table 5). Negative-only training leads on MATH-500,

Table 4: On-policy results on mathematics benchmarks. Scores are mean@8 percentages; AVG is the unweighted mean over the five benchmarks. Bold and underline denote the best and second-best values within each model block, excluding Base; ties are retained. Checkpoints are selected within steps 0–200.

Model	Method	AIME24	MATH-500	Minerva	Olympiad	AMC23	AVG ↑
Qwen2.5-7B	Base	3.74	43.27	15.35	15.16	22.19	19.94
	GRPO	9.18	70.80	32.64	32.49	<u>50.31</u>	39.08
	GSPO	10.82	<u>70.83</u>	31.39	32.34	47.50	38.58
	RAFT	2.08	39.83	13.80	15.70	21.88	18.66
	RAFT++	5.84	55.25	19.90	23.32	37.81	28.43
	RAFT w/ DFT	**12.93**	70.58	30.74	**33.66**	48.75	39.33
	RAFT++ w/ DFT	11.66	66.62	30.21	30.40	43.12	36.41
	RAFT++ w/ VRA	10.00	**71.47**	<u>33.50</u>	32.54	49.69	<u>39.44</u>
	VRA (ours)	<u>12.91</u>	69.95	**34.80**	<u>33.56</u>	**53.75**	**40.99**
Qwen3-4B	Base	4.99	37.65	12.84	17.24	25.31	19.61
	GRPO	11.66	<u>74.97</u>	37.26	36.14	48.75	41.76
	GSPO	10.00	74.53	<u>41.69</u>	36.58	<u>53.75</u>	43.31
	RAFT	2.92	39.75	18.66	16.80	26.56	20.94
	RAFT++	4.99	45.90	18.39	21.20	31.88	24.47
	RAFT w/ DFT	**13.74**	74.92	39.06	36.54	52.50	<u>43.35</u>
	RAFT++ w/ DFT	10.41	73.33	37.59	<u>37.66</u>	50.94	41.98
	RAFT++ w/ VRA	<u>11.68</u>	74.60	36.04	37.62	**54.38**	42.86
	VRA (ours)	<u>11.68</u>	**76.67**	**41.75**	**38.77**	**54.38**	**44.65**
Qwen3-8B	Base	8.34	58.10	23.48	26.62	38.44	31.00
	GRPO	13.32	77.25	**45.41**	38.62	<u>56.88</u>	46.30
	GSPO	13.75	76.00	41.67	39.73	**58.75**	45.98
	RAFT	7.91	57.48	22.07	25.71	35.31	29.70
	RAFT++	4.56	43.60	16.81	19.71	20.00	20.94
	RAFT w/ DFT	<u>14.16</u>	<u>77.58</u>	<u>44.41</u>	<u>40.41</u>	55.94	<u>46.50</u>
	RAFT++ w/ DFT	12.09	76.28	38.60	39.73	56.25	44.59
	RAFT++ w/ VRA	11.66	71.55	34.75	35.27	48.12	40.27
	VRA (ours)	**18.34**	**78.25**	43.67	**41.04**	**58.75**	**48.01**

Table 5: Analytic-component ablation on Qwen2.5-7B. All columns report mean@8 percentages; higher is better. Bold marks the highest AVG.

Variant	AIME24	MATH-500	Minerva	Olympiad	AMC23	AVG ↑
Positive only	10.41	68.62	31.27	31.79	50.00	38.42
Negative only	12.90	71.08	32.54	32.10	44.38	38.60
Full VRA	12.91	69.95	34.80	33.56	53.75	**40.99**

so the benefit of the full coefficient is an aggregate result. The comparison supports retaining both terms in this setting.

D.3 Coefficient Scale

We test whether the VRA–GRPO AVG ordering persists when their worst-case coefficient magnitudes are brought to similar scales. For a group of N binary rewards with K successes, using the group-population standard deviation gives, for $0 < K < N$,

$$A_+ = \sqrt{\frac{N-K}{K}}, \qquad A_- = -\sqrt{\frac{K}{N-K}}, \qquad |A_i| \le \sqrt{N-1}.$$

Homogeneous groups have zero centered reward coefficients. At $N = 8$, the GRPO advantage bound is $\sqrt{7} \approx 2.646$, whereas

$$\sup_{0<p<1} w_\epsilon(p) = \epsilon^{-1/2} \approx 31.62.$$

We compare GRPO $\times 1$ with VRA $\times 0.1$, and GRPO $\times 10$ with VRA $\times 1$. The respective bound pairs are $(2.646, 3.162)$ and $(26.46, 31.62)$. This matches the bounds only approximately: the VRA bound remains about 19.5% larger in each pair.

Table 6: Coefficient-scale control on Qwen2.5-7B with $N = 8$. Bound denotes the approximate upper bound on the absolute reward or token coefficient. Scores are mean@8 percentages. Bold marks the higher AVG within each pair.

Variant	Bound	AIME24	MATH-500	Minerva	Olympiad	AMC23	AVG ↑
GRPO $\times 1$	2.646	9.18	70.80	32.64	32.49	50.31	39.08
VRA $\times 0.1$	3.162	9.57	72.22	35.44	32.16	51.25	**40.13**
GRPO $\times 10$	26.46	9.99	71.60	33.24	33.36	50.00	39.64
VRA $\times 1$	31.62	12.91	69.95	34.80	33.56	53.75	**40.99**

The AVG ordering persists in both comparisons: VRA $\times 0.1$ exceeds GRPO $\times 1$ by 1.05 points, and VRA $\times 1$ exceeds GRPO $\times 10$ by 1.35 points (Table 6). These controls show that the observed ordering survives these constant rescalings. They do not establish independence from learning rate or coefficient scale in general.

D.4 Rollout Count

The number of responses per prompt changes the available on-policy data. More responses provide additional samples for GRPO's group-relative statistics and additional opportunities for VRA to collect accepted trajectories. Table 7 varies this count from $N = 4$ to $N = 16$. These comparisons do not hold the number of generated responses or total computation fixed.

Table 7: Rollout-count sensitivity on Qwen2.5-7B. N is the number of sampled responses per prompt. Scores are mean@8 percentages; bold marks the higher AVG at each N.

N	Method	AIME24	MATH-500	Minerva	Olympiad	AMC23	AVG ↑
4	GRPO	8.34	69.53	28.11	30.12	49.69	37.16
	VRA	11.68	71.65	32.56	33.52	51.56	**40.20**
8	GRPO	9.18	70.80	32.64	32.49	50.31	39.08
	VRA	12.91	69.95	34.80	33.56	53.75	**40.99**
16	GRPO	11.68	71.67	34.46	32.55	52.19	40.51
	VRA	12.91	73.58	35.58	34.41	54.38	**42.17**

Increasing N from 4 to 16 improves AVG for both methods: GRPO rises from 37.16 to 40.51, and VRA from 40.20 to 42.17. VRA leads by 3.04, 1.91, and 1.66 points at $N = 4, 8, 16$, respectively. Thus the favorable ordering is present at all three sampled group sizes, while the gap narrows as the rollout count increases.

D.5 Transfer of the Training Rule to Code

We train Qwen2.5-7B on CodeRL+ (Jiang et al., 2026), transferring the training rule to code rather than evaluating a math-trained checkpoint. Training follows the on-policy settings in Appendix D.1, with code data and verification. The five evaluation tasks are HumanEval (Chen et al., 2021), the 180-item DeepSeek-Coder LeetCode Contest benchmark (Guo et al., 2024), and LiveCodeBench Code Generation, Code Execution, and Output Prediction (Jain et al., 2025).

Table 8: CodeRL+ on-policy study with Qwen2.5-7B. All task columns report pass@1 percentages; AVG is their unweighted mean. Bold marks the highest AVG. LCB denotes LiveCodeBench.

Method	HumanEval	LeetCode	LCB CodeGen	LCB CodeExec	LCB OutPred	AVG ↑
Base	50.61	38.33	24.87	53.03	43.62	42.09
GRPO	65.85	52.22	27.73	53.65	45.48	48.99
VRA	74.39	48.89	29.03	60.96	45.54	**51.76**

VRA improves AVG by 9.67 points over the base model and by 2.77 points over GRPO. It leads on four of the five tasks; GRPO leads on LeetCode by 3.33 points (Table 8). This supports utility beyond math at one model scale.

D.6 Gold-Data Fine-Tuning with Reference Regularization

Setting. Table 9 reports the reference-regularized runs on Qwen2.5-7B and Qwen3-4B/8B using the NuminaMath-CoT 30K demonstrations. SFT with KL and VRA with KL add a reference-policy KL term; ASFT (Zhu et al., 2026a) is the reference-based comparison. The VRA KL coefficient is 0.05. The gold-data setup is given in Appendix F; these runs are reported separately because they require a reference model. Each method reports its highest-AVG checkpoint over the full run.

Results. Within the reference-based comparisons, VRA with KL has the highest AVG at all three model sizes, exceeding ASFT by 2.14, 0.48, and 0.60 points, respectively. The reference-based results extend the evaluation to another training regime while preserving the reference-free method as the main comparison.

Table 9: Gold-data fine-tuning with a reference policy. All three methods in each model block use reference-based regularization. Scores are mean@8 percentages; AVG is the five-benchmark macro-average. Bold and underline mark the best and second-best values within each model block; ties are retained.

Model	Method	AIME24	MATH-500	Minerva	Olympiad	AMC23	AVG ↑
Qwen2.5-7B	SFT w/ KL	3.76	50.30	13.91	18.19	31.25	23.48
	ASFT	8.32	60.17	**22.44**	26.96	36.56	30.89
	VRA w/ KL	**10.00**	**63.62**	22.18	**29.02**	**40.31**	**33.03**
Qwen3-4B	SFT w/ KL	2.08	50.15	18.56	21.49	30.94	24.64
	ASFT	**3.74**	59.23	**24.45**	25.90	32.19	29.10
	VRA w/ KL	**3.74**	**61.35**	20.73	**27.07**	**35.00**	**29.58**
Qwen3-8B	SFT w/ KL	2.92	60.15	22.29	21.81	33.44	28.12
	ASFT	7.91	68.53	29.76	33.31	**48.44**	37.59
	VRA w/ KL	**10.84**	**69.90**	**29.96**	**33.70**	46.56	**38.19**

E Gold-Data Analysis

This appendix gives the protocols behind Section 6 and two further measurements on the same gold data: acquisition of the demonstrated targets during training, and a held-out response behavior.

E.1 Census and Matched One-Step Updates

Probability census. Section 6 uses 4,000 examples sampled from the NuminaMath-CoT 30K training subset. The Qwen3-8B base model scores the gold responses by teacher forcing in the training format: the Alpaca prompt, a trailing response EOS, and a maximum total length of 2,048 tokens. The census contains 1,720,740 target tokens; 10 sequences are truncated at the length limit. Figure 3 reports pooled-token exposure: each target token contributes equally to the probability histogram. For a probability region R and token weight w, its nominal direct target-logit signal share is

$$Q_w(R) = \frac{\sum_t \mathbf{1}[p_t \in R] w(p_t)(1-p_t)}{\sum_t w(p_t)(1-p_t)}.$$

Figure 3 evaluates this quantity for SFT, DFT, InfoSFT, ProFit, and VRA using their main-run weights. Each method is normalized separately, so the figure compares allocation across probability regions but not absolute signal scales.

Matched one-step updates. The comparisons of Section 6.2 restore the full training state of the SFT run at steps 100 and 300, including parameters, optimizer moments, and random-number state. From each state, we take one optimizer step with gradient clipping at 1.0 and the run's scheduled learning rate at that step (8.27×10^{-6} and 4.89×10^{-6}), once with the token-mean log-likelihood and once with the detached VRA weight ($\epsilon = 10^{-3}$). Both updates use the same batch, and we repeat the comparison for three disjoint 256-example batches drawn from the training set. After each step, we rescore all 1,720,740 census targets. Groups are assigned by the target probabilities at each starting checkpoint (82,074 / 420,500 / 1,218,166 tokens at step 100; 65,699 / 345,923 / 1,309,118 at step 300), so they differ from the Base groups used elsewhere. Before clipping, the unit-weight gradient has norm 16.7–19.8 and the VRA gradient 75.7–99.2, so clipping is active in every case. Table 10 lists the per-batch contrasts summarized in Figure 4a. For Figure 4b, a prompt moves in the predicted direction when its average contrast within a region is negative in the low region or positive in the middle or high region; the numbers of prompts with targets in the three regions are 3,724 / 4,000 / 4,000 at step 100 and 3,295 / 3,935 / 4,000 at step 300.

Table 10: Per-batch contrasts underlying Figure 4: prompt-averaged signed Fisher–Rao change after one VRA step minus one unit-weight step.

Checkpoint	Batch	$p_0 < 0.1$	$0.1 \le p_0 < 0.9$	$p_0 \ge 0.9$
Step 100	0	-0.0091	$+0.0113$	$+0.0071$
	1	-0.0093	$+0.0114$	$+0.0069$
	2	-0.0095	$+0.0121$	$+0.0075$
Step 300	0	-0.0079	$+0.0070$	$+0.0039$
	1	-0.0075	$+0.0061$	$+0.0034$
	2	-0.0070	$+0.0070$	$+0.0037$

E.2 Same-Target Acquisition on Gold Data

Acquisition during training. We score each run's checkpoints at steps 50 through 300 on the same census, with the same Base groups. Each point in Figure 6 is a measured checkpoint.

The trajectories show how the regional differences develop. SFT rises fastest and highest in the low group, whereas VRA already has 0.652 middle-group and 0.141 high-group acquisition at step 50. Across checkpoints, VRA remains near the upper envelope in the middle group; by step 300 its 0.8855 mean is nearly identical to ProFit's 0.8854. DFT and VRA are close to 0.14 in the high group throughout, while SFT and InfoSFT remain below that level. The trajectories therefore show persistent allocation differences rather than a transient crossing.

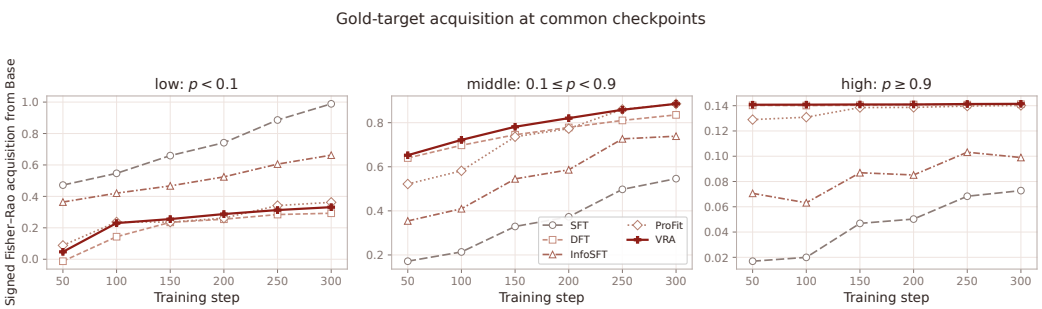


Figure 6: **Acquisition during training.** Prompt-averaged signed Fisher–Rao change from Base in the low, middle, and high groups at training steps 50–300.

Same-target comparison at step 300. We next score the same 1,720,740 gold target events under the base model and the five fine-tuned models at step 300, the last checkpoint in Figure 6. ProFit uses a fixed threshold $p > 0.1$, InfoSFT uses fixed calibration 0.93, and VRA uses detached $\sqrt{p/(1-p+0.001)}$ weights with no KL term; target log-probabilities are computed in FP32. Probability groups are assigned once from the Base scores and held fixed for every method; the eight groups contain 12,449; 20,484; 72,282; 180,010; 257,501; 280,304; 292,210; and 605,500 tokens. For method m, the signed change is

$$\Delta\ell_{m,t} = 2\arcsin\sqrt{p_{m,t}} - 2\arcsin\sqrt{p_{0,t}}.$$

Within each group, we first average over a prompt's target tokens and then over the prompts with targets in that group. Uncertainty intervals use 2,000 bootstrap resamples of prompts, shared across methods.

In the three Base groups, the coarse means for SFT, DFT, InfoSFT, ProFit, and VRA are 0.9889, 0.2928, 0.6623, 0.3623, and 0.3314 in the low group; 0.5461, 0.8354, 0.7387, 0.8854, and 0.8855 in the middle group; and 0.0728, 0.1411, 0.0990, 0.1402, and 0.1415 in the high group. All values are positive; larger values mean more movement toward the demonstrated target. Figure 7 resolves the same comparison into the eight census bins.

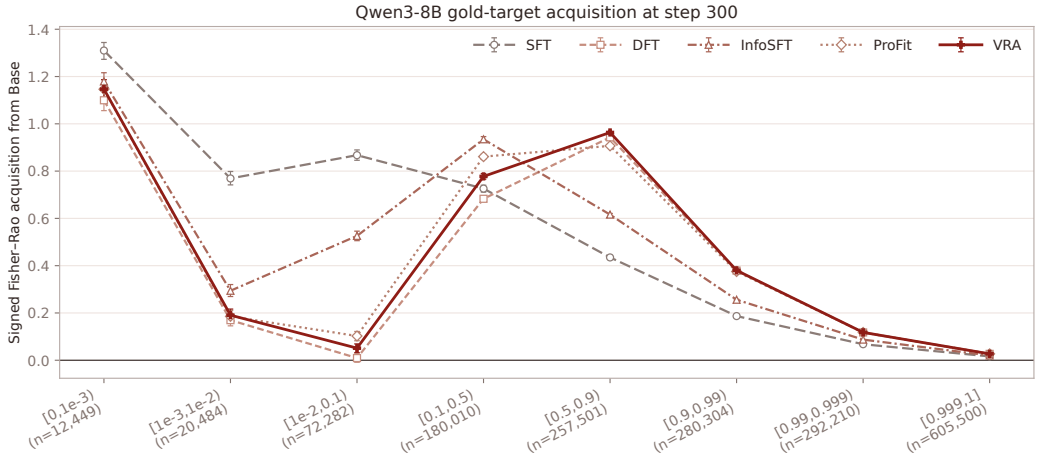


Figure 7: **Acquisition at step 300 by base-probability bin.** Points show prompt-averaged signed Fisher–Rao changes from Base; error bars are 95% intervals from 2,000 paired bootstrap resamples of prompts. The token count of each Base group is printed below its bin.

In the low group, the realized ordering SFT > VRA > DFT matches the scalar-flow acquisition-time ordering of Section 4 (Figure 7): VRA is below SFT and InfoSFT, below ProFit by -0.0308 (95% interval $[-0.0371, -0.0243]$), and above DFT by $+0.0387$ ($[+0.0325, +0.0450]$). In the middle group, it exceeds SFT, DFT, and InfoSFT, and is indistinguishable from ProFit under this interval:

the VRA-minus-ProFit contrast is $+0.0001$ ($[-0.0033, +0.0035]$). In the high group, VRA has the largest mean, with margins of $+0.0003$ over DFT and $+0.0012$ over ProFit. This measurement thus shows the predicted redistribution across the base probability axis; no single method leads in every region.

Two properties of this measurement delimit what it can show. It separates the rules precisely where their gain profiles differ most—in the low group, the likelihood-side and suppression-side methods split, and the VRA–DFT contrast excludes zero—and it coincides where indirect parameter coupling dominates: ProFit receives no direct signal below its threshold yet moves the low group by 0.3623. Moreover, the middle-group means of VRA and ProFit agree to 10^{-4} while their AVG mean@8 differs by 2.77 points (Table 2), so the amount of gold acquisition does not by itself explain the task-level differences; the one-step contrasts of Section 6.2 and properties invisible to this coarse measurement, such as non-target reallocation, must carry them. Because these are training-set gold prefixes, the measurement is conditional acquisition rather than free-generation transfer.

E.3 Which Useful Behaviors Are Acquired?

The task-level motivation in Section 1 concerns useful unfamiliar targets, not unfamiliarity by itself. We therefore fix a narrow, objective response behavior in advance: the solution ends with a nonempty, balanced `\boxed{...}` (or `\fbox{...}`) answer, followed only by closing math markup or punctuation. This makes the proposed answer unambiguous and extractable, but does not certify its mathematical correctness.

The held-out pool contains 1,500 NuminaMath problems that do not appear in the 30K training subset, after removing duplicates and problems whose reference answers fail verification. We use 64 of them to define the criterion and evaluate on 256 others. Empty, unbalanced, continuing, and length-truncated outputs count as failures; answer correctness is judged separately by a symbolic grader, after normalizing formatting and mapping a single multiple-choice letter to its option. Each of the six models produces one greedy response (temperature 0) of at most 2,048 new tokens in the training prompt format. For span acquisition, the target is the final boxed span of each reference response; all 256 spans fit within the 2,048-token scoring window, and the statistic averages signed Fisher–Rao changes over a prompt's span tokens and then over prompts. Intervals come from 2,000 paired bootstrap resamples of prompts. Table 11 reports the structural criterion, answer correctness, and span acquisition.

Table 11: **Held-out final-answer behavior (256 prompts).** Completion and answer columns are rates. Span acquisition is the prompt-averaged change on the reference boxed span under the gold prefix.

Method	Boxed completion	Answer correct	Span acquisition
Base	54.3%	43.8%	0.000
SFT	65.6%	54.3%	0.279
DFT	64.8%	54.7%	0.385
InfoSFT	63.3%	58.2%	0.353
ProFit	64.8%	55.5%	0.375
VRA	63.7%	56.3%	0.392

Relative to Base, VRA raises boxed completion by 9.4 percentage points (95% paired prompt-bootstrap interval [2.0,17.2]) and answer correctness by 12.5 points ([5.9,19.5]). The structural differences between VRA and each trained baseline are small and all four intervals contain zero: -2.0 points versus SFT, -1.2 versus DFT, +0.4 versus InfoSFT, and -1.2 versus ProFit. On the reference boxed span, VRA has the largest mean conditional acquisition. Its differences are +0.113 versus SFT ([0.092,0.132]), +0.007 versus DFT ([-0.008,0.022]), +0.039 versus InfoSFT ([0.020,0.057]), and +0.017 versus ProFit ([0.000,0.034]). Within each trained method, prompts that pass the structural rule do not have larger mean span acquisition than failures; for VRA the respective means are 0.368 and 0.435. Thus the acquired targets transfer beyond Base; on this structural behavior the trained methods are statistically comparable.

A traced case. One case, selected by a fixed rule, asks for three-color, three-stripe flags with adjacent stripes unequal. The reference final span is $\boxed{12}$, with Base mean span probability 0.817. Base

derives 12 but ends in bold text without a box (structure fail, answer correct). SFT and VRA end in $\boxed{12}$ (both pass and are correct), whereas DFT ends in $\boxed{21}$ (passes the structure but is incorrect). InfoSFT gives the correct boxed value and then continues with substantive text, while ProFit similarly continues after an incorrect $\boxed{21}$; both fail the terminal-box criterion. Their span-acquisition values for Base, SFT, DFT, InfoSFT, ProFit, and VRA are 0.000, 0.332, 0.168, 0.375, 0.211, and 0.166. The example separates three quantities that can disagree: learning the reference target under a gold prefix, emitting the designated response structure, and solving the problem.

F Experimental Details

F.1 Teacher-Forced Implementation

For target positions $\mathcal{T}$ in gold responses, we use

$$w_\epsilon(p) = \sqrt{\frac{p}{1-p+\epsilon}}, \qquad \mathcal{L}(\theta) = -\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathrm{sg}(w_\epsilon(p_t)) \log p_t, \qquad \epsilon = 10^{-3}.$$

Prompt and padding positions are excluded. Detaching the weight is essential: differentiating through the product would introduce an additional term and would no longer implement the derived direction. The rule adds a scalar operation to ordinary token loss, requires no reference model, and has no tuned weight-shape parameter. The fixed stabilizer bounds the raw weight; its effect on the gain profile and the local gradient is derived in Appendix B.4. We report this implemented rule in all VRA experimental rows; exact equal gain and the closed-form time expressions refer to the ideal rule.

F.2 Training and Evaluation

Table 12 specifies the setup for the SFT, DFT, and VRA runs on the six backbones. MiMo-7B, OLMo3-7B, and Ling 3.0 Tiny 8B follow the same configuration as Qwen2.5-7B and Qwen3-8B, starting from the base checkpoints `XiaomiMiMo/MiMo-7B-Base`, `allenai/Olmo-3-1025-7B`, and `inclusionAI/Ling-3.0-tiny-base`. Ling 3.0 Tiny 8B is a mixture-of-experts model with 7.9B total and 1.3B activated parameters per token (Ling Team, 2026); the other five backbones are dense. InfoSFT uses Qwen3-8B with NuminaMath-CoT 30K, a five-epoch run, and calibration $\bar{p} = 0.93$.

Table 12: Gold-data training configuration for the six backbones, using SFT, DFT, and VRA.

Setting	Value
Training set	A fixed, randomly sampled 30K-example subset of NuminaMath-CoT, shared across the core methods and the six backbones.
Supervision	Alpaca-style prompt; prompt tokens masked, response tokens including the trailing EOS supervised.
Fine-tuning	Full parameters; five epochs; maximum sequence length 2,048; no packing; right truncation.
Optimizer	AdamW, $\beta = (0.9, 0.999)$, zero weight decay, gradient clipping at 1.0.
Schedule	Linear learning-rate decay, no warmup.
Learning rate	3×10^{-5} for Qwen3-4B; 10^{-5} for Qwen2.5-7B, Qwen3-8B, MiMo-7B, OLMo3-7B, and Ling 3.0 Tiny 8B.
Batch and precision	Global batch size 256; bfloat16.
VRA	Detached weights with $\epsilon = 10^{-3}$; no reference-policy KL penalty.

The five evaluation sets contain 30 AIME24, 500 MATH-500, 272 Minerva Math, 675 OlympiadBench, and 40 AMC23 problems. For a benchmark with n problems and binary correctness a_{ij} for sample j of problem i, mean@8 is defined as

$$\text{mean@8} = \frac{100}{8n}\sum_{i=1}^{n}\sum_{j=1}^{8} a_{ij}.$$

AVG macro-averages the five benchmark metrics, so each benchmark has the same weight regardless of its size.