GRPO, Dr. GRPO, and DAPO Are Three Operations on One Number: The Group-Standard-Deviation Identity

Yong Yi Bay* **Kathleen A. Yearick***

PhD, University of Illinois at Urbana-Champaign

ABSTRACT

Three of the most popular methods for training language models to reason look like three different tricks. They are not. All three adjust a single number: *standard deviation*, reflecting how much a prompt's sampled answers disagree. When such a model is trained, it answers each problem many times, and an automatic checker marks every answer right or wrong. The standard deviation of those marks measures the disagreement: largest when the answers split evenly between right and wrong, and zero when they all agree. Group Relative Policy Optimization (GRPO) divides by this number, GRPO Done Right (Dr. GRPO) drops the division, and Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) discards the groups where it is zero. Each is presented as its own fix, yet this paper proves they are three settings of one dial. That dial is not cosmetic: for right-or-wrong rewards, the disagreement is exactly the size of the training update, the *group-standard-deviation identity*. A split group teaches the most, while a unanimous group teaches nothing and falls silent. The same result says which problems deserve the most weight and how many tries each one needs. This paper confirms the intuition on a large real difficulty dataset (Big-Math) and in a controlled training run. What looks like a harmless normalization step is the dial that decides where learning happens and how strongly.

Keywords GRPO · reward normalization · silent groups · difficulty bias · group size · dynamic sampling · RLVR · LLM reasoning

1 Introduction

Teaching a language model to reason starts with a step that looks wasteful: the same prompt is answered many times over, on purpose. This happens during reinforcement learning, not when a user chats with the model. The model produces a group of candidate answers, an external verifier marks each one as correct or incorrect, and the optimizer updates the model so that rewarded answer paths become more likely. The repeated answers are not redundant outputs for a user; they are measurements of the model's current uncertainty on that prompt.

A simple example captures the mechanism. Suppose a model attempts the same math problem eight times. If all eight attempts are wrong, there is no successful attempt to imitate. If all eight are right, there is no failed attempt to move away from. The useful training case is mixed: some attempts are right and some are wrong. Only then can the training rule compare the two sides. In this sense, a prompt teaches through its *within-group disagreement*.

Group Relative Policy Optimization [GRPO; 1, 2], the workhorse of current verifiable reasoning training, is built around this comparison. GRPO operates in the *reinforcement learning with verifiable rewards* setting

*Equal contribution. Correspondence: {yongyibay, kallie.a.yearick}@gmail.com.

(RLVR), where an automatic checker returns a reward, usually 1 for a correct final answer and 0 for an incorrect one. The model supplies the candidate answers; the verifier supplies the rewards; GRPO supplies the rule that converts those rewards into *advantages*; the optimizer changes the model parameters.

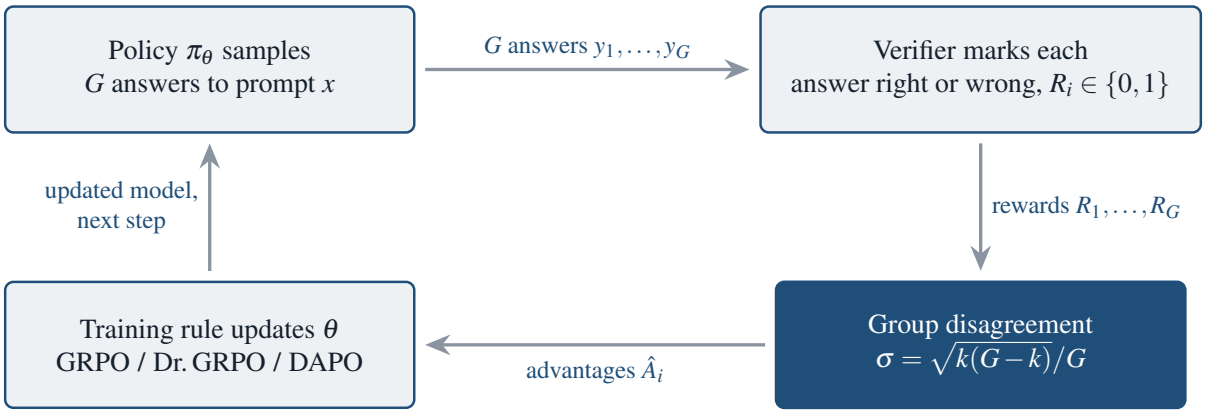

Figure 1: The training-time loop studied in this paper. The trainer samples one prompt many times to compare its correct and incorrect attempts, and the group reward standard deviation σ measures whether they disagree.

The mean subtraction inside GRPO needs little controversy: subtracting any action-independent baseline preserves the policy gradient while reducing variance [3]. The contested step is the next one, division by the group standard deviation. It is often dismissed as a normalization detail, yet Liu et al. [4] identify it as the source of a *question-level difficulty bias* and remove it. Its large-group effect is already understood: for binary rewards it makes GRPO ascend not the raw success rate p, but the *arcsine transform* $\mathbb{E}_x[2\arcsin\sqrt{p_x}]$, the classical variance-stabilizing transform of a binomial proportion [5, 6]. Thrampoulidis et al. [7] established this surrogate-reward view.

> **The lens.** The central object is the group reward standard deviation σ. It is the amount of disagreement in the verifier's marks inside one prompt's sampled group. GRPO divides by σ, Dr. GRPO drops that division, and DAPO's dynamic sampling discards groups with $\sigma = 0$. The paper's claim is that these are not separate tricks but three operations on the same number.

This paper. Real training does not take the large-group limit; it chooses a finite group size G and updates from the sampled group it receives. The paper's main claim is the exact finite-group accounting behind that update. For binary rewards, if k of G sampled answers are correct, then a single GRPO step on that prompt is, exactly and in any dimension,

$$g \;=\; \frac{1}{G}\sum_i \hat{A}_i \nabla_\theta \log \pi_\theta(y_i) \;=\; \sigma\left(\bar{s}_+ - \bar{s}_-\right), \qquad \sigma = \frac{\sqrt{k(G-k)}}{G}, \quad 0 < k < G, \tag{1}$$

independently of the baseline, where $\bar{s}_+$ and $\bar{s}_-$ are the mean scores of the correct and incorrect rollouts. The update has two plain pieces. The direction $\bar{s}_+ - \bar{s}_-$ says what to favor: correct rollouts over incorrect rollouts. The multiplier σ says how strongly to favor it: it vanishes for a unanimous group and peaks for an evenly split one. Equation (1) is the *group-standard-deviation identity*: what sits in the advantage's denominator is the length of the gradient itself. The scalar case $g(k) = \sqrt{k(G-k)}/G$ (where $\bar{s}_+ - \bar{s}_- = 1$) is the form used in the remaining analysis. Averaging (1) over groups recovers the arcsine gradient of Thrampoulidis et al. [7] as $G \to \infty$; the identity is the exact finite-G object underneath that limit. Figure 1 locates the quantity inside

the training loop, and Figure 2 gives the resulting method map: GRPO, Dr. GRPO, and DAPO act on one scalar quantity in three different ways.

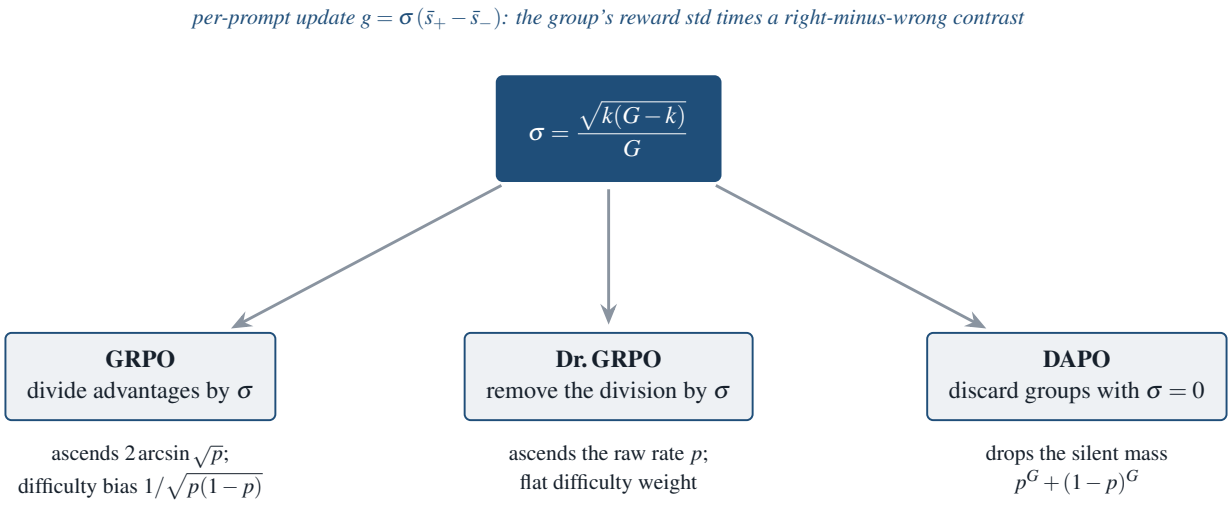

Figure 2: One object, three interventions. The group reward standard deviation $\sigma = \sqrt{k(G-k)}/G$ that GRPO divides by, Dr. GRPO drops, and DAPO filters to zero.

Why the exact form matters. A practitioner does not choose an asymptotic limit; a practitioner chooses G and decides which prompts to keep. The finite-group identity turns both choices into closed forms. The resulting contributions are:

1. **The group-standard-deviation identity** (§3, Theorem 1). The per-prompt GRPO update is the contrastive score direction scaled by the group's reward standard deviation $\sqrt{k(G-k)}/G$, exactly, baseline-free, and in any dimension. Its expectation over $k \sim \text{Binomial}(G,p)$ is an exact binomial sum; the large-group arcsine gradient $\sqrt{p(1-p)}$ and its finite-G attenuation $1 - 1/(8Gp(1-p))$ are its limit and first correction.

2. **A closed-form group-size law** (§4, Corollary 2). A group of size G realizes a fraction $\varphi \approx 1 - 1/(8Gp(1-p))$ of the large-group gradient, so reaching fidelity $1-\varepsilon$ needs $G \gtrsim 1/(8\varepsilon p(1-p))$, the square of the difficulty weight: a coin-flip prompt is faithful by $G \approx 10$, a prompt at 5% success needs $G \approx 70$ (Table 2). The budget is read off difficulty, not swept.

3. **The silent-group rate** (§5). A group is silent (zero advantage everywhere) with probability $p^G + (1-p)^G$; this is exactly the $\sigma = 0$ mass that DAPO's dynamic sampling [8] over-samples and discards, and its logged all-correct fraction is the same functional family $\mathbb{E}_p[p^n]$, whose shape and sub-one plateau the closed form reproduces.

4. **The difficulty bias, exactly** (§6). The rate at which GRPO converts success probability into objective is $\partial_p 2\arcsin\sqrt{p} = 1/\sqrt{p(1-p)}$, the very $1/\sigma$ the advantage divides by; this is the question-level difficulty bias of Liu et al. [4], and deleting σ reverts the objective from $\arcsin\sqrt{p}$ to p. Group-mean centering is the leave-one-out (RLOO) advantage up to the constant $G/(G-1)$, so the division is the only objective-changing step.

5. **Validation on real difficulty distributions** (§7). On Big-Math [9], $N = 215{,}608$ problems with empirical solve rates, the standardization moves $13.9\% \to 24.7\%$ of the implicit objective's gradient mass onto

extreme prompts; the silent-group rate is 44% at the common group size $G = 8$ and matches direct subsampling of the logged rollouts to within two points.

6. **The predictions in a controlled run** (§8). A real GRPO loop over 6,000 Bernoulli-logit prompts confirms the closed forms as training dynamics: the silent-group rate tracks the measured wasted-group fraction step by step ($R^2 = 0.999$), the realized gradient mass matches the finite-G reweighting, and the difficulty bias is visible as GRPO lifting the hardest prompts where Dr. GRPO stalls.

These results do not require a new algorithm or a large training run. They follow from exact accounting of one sampled group, with the data-dependent quantities read off published rollout statistics.

2 Three Methods, One Operation Apart

This section names the objects in the training loop and spells out the three method formulas. The setting is a single prompt x during training. The policy π_θ samples responses, the verifier assigns rewards, and the training rule converts the reward pattern into a parameter update. All comparisons below differ only in how they handle the group reward standard deviation σ.

Acronyms and scope. The three names are Group Relative Policy Optimization (GRPO), GRPO Done Right (Dr. GRPO), and Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO). Dr. GRPO also discusses length normalization, and DAPO contains additional engineering choices. The present analysis isolates the common axis relevant to all three: the standard-deviation operation on a group of verifier rewards.

GRPO advantages. For a prompt x, GRPO draws a group of G responses $y_1, \ldots, y_G \sim \pi_\theta(\cdot \mid x)$ with scalar rewards $R_1, \ldots, R_G$, and forms the standardized advantage

$$\hat{A}_i = \frac{R_i - \mu}{\sigma}, \qquad \mu = \frac{1}{G}\sum_j R_j, \quad \sigma = \sqrt{\frac{1}{G}\sum_j (R_j - \mu)^2}, \tag{2}$$

which is broadcast to every token of y_i and plugged into the usual clipped surrogate with a KL penalty to a reference policy. The analysis here concerns the advantage construction; the clipping and KL terms are standard regularizers that do not affect the advantage's expectation at the first, on-policy step (ratio $= 1$), and are held aside throughout.

Two updates, one difference. Dropping the division by σ from (2) leaves the mean-centered advantage $\hat{A}_i = R_i - \mu$. This second update is neither new nor specific to one method: it is the update used by Dr. GRPO [4], and it equals the leave-one-out (RLOO) advantage [10, 11] and baselined REINFORCE [3] up to a learning-rate constant (Proposition 1). The equivalence is straightforward: each method subtracts a baseline from the reward and stops there. GRPO alone adds one further operation, division by σ. Thus the objective-changing distinction is not the baseline; it is the standard-deviation operation. Table 1 makes this explicit after the binary-reward notation is introduced.

Binary rewards. In RLVR the reward is the verifier's verdict, $R_i \in \{0, 1\}$. The verifier is the checker: for math it may compare final answers or symbolic forms, for code it may run tests, and for structured tasks it may apply a parser or schema. GRPO does not decide truth by itself; it receives the reward vector $R_1, \ldots, R_G$ from this checker. Write $p = p_x(\theta) = \mathbb{P}_{y \sim \pi_\theta(\cdot|x)}[R = 1]$ for the policy's success probability on x, and let

$k = \sum_i R_i$ be the number of correct samples in a group. Then $\mu = k/G$ and, because rewards are Bernoulli, $\sigma = \sqrt{\mu(1-\mu)} = \sqrt{k(G-k)}/G$ *exactly*. When $k = 0$ all sampled answers are wrong; when $k = G$ all sampled answers are right. In both cases $\sigma = 0$, so there is no right-versus-wrong contrast inside the group.

Method	**Operation on σ**	**Per-prompt update g**	**In words**
GRPO	Divide by σ	$g = \sigma\,\Delta s$	The update size is the group's disagreement.
Dr. GRPO	Remove the division	$g = \sigma^2\,\Delta s$	Ascends raw accuracy, with a flat difficulty weight.
DAPO	Drop the $\sigma = 0$ groups	$g = \mathbf{1}\{0 < k < G\}\,\sigma\,\Delta s$	Discards groups with no right-versus-wrong contrast.

Table 1: The three methods as operations on one number. Here $\Delta s = \bar{s}_+ - \bar{s}_-$ and $\sigma = \sqrt{k(G-k)}/G$; the advantage is $\hat{A}_i = (R_i - \mu)/\sigma$ for GRPO and $\hat{A}_i = R_i - \mu$ for Dr. GRPO. GRPO scales the update by σ, Dr. GRPO removes that scaling, and DAPO changes which groups contribute.

The score identity. The argument uses the elementary policy-gradient fact that for any baseline b independent of y,

$$\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\big[(R(y) - b)\,\nabla_\theta \log \pi_\theta(y \mid x)\big] = \nabla_\theta\,\mathbb{E}_y[R(y)] = \nabla_\theta\, p_x(\theta). \tag{3}$$

Mean-centering is leave-one-out, rescaled. The mean subtraction has a familiar form. Excluding the sample being scored gives the leave-one-out (RLOO) baseline $b_i = \frac{1}{G-1}\sum_{j\neq i} R_j$ [10, 11], whose advantage $R_i - b_i$ is unbiased by (3), and subtracting the group mean is the same thing up to a constant.

Proposition 1 (Group-mean centering is rescaled RLOO)**.** *For any group $R_1, \ldots, R_G$ with $G \geq 2$ and RLOO baseline $b_i = \frac{1}{G-1}\sum_{j\neq i} R_j$,*

$$R_i - \mu \;=\; \frac{G-1}{G}\,(R_i - b_i) \qquad \textit{for every } i.$$

Proof. $\sum_j R_j = R_i + (G-1)b_i$, so $\mu = \frac{1}{G}\big(R_i + (G-1)b_i\big)$ and $R_i - \mu = \frac{G-1}{G}(R_i - b_i)$. □

Mean-centering is therefore the unbiased RLOO advantage up to the constant $G/(G-1)$, which a learning rate absorbs; both updates share it, so it is not where they differ. The division by σ, the step that does change the objective, is analyzed next.

3 The Update Is the Group's Standard Deviation

The main result is the mathematical version of the training story above. Take one prompt, sample G answers, and let the verifier split them into correct and incorrect sets. GRPO moves the model toward the correct side and away from the incorrect side. The theorem below states that the length of this move is exactly the group reward standard deviation. Thus the same scalar that appears in the denominator of the advantage is also the scalar that measures how much learning signal the prompt actually produced. Everything downstream (the group-size budget, the silent fraction, and the difficulty bias) is read from this one form. The statement holds for any policy in any dimension, not only for a scalar model.

Theorem 1 (The group-standard-deviation identity)**.** *Fix a prompt and a group of G responses with binary rewards, k of them correct, $0 < k < G$. Let $s_i = \nabla_\theta \log \pi_\theta(y_i \mid x)$ be the score of response i, and let $\bar{s}_+ = \frac{1}{k}\sum_{i:R_i=1} s_i$ and $\bar{s}_- = \frac{1}{G-k}\sum_{i:R_i=0} s_i$ be the mean scores of its correct and incorrect responses. The per-prompt GRPO update is, exactly,*

$$g \;=\; \frac{1}{G}\sum_i \hat{A}_i s_i \;=\; \sigma\left(\bar{s}_+ - \bar{s}_-\right), \qquad \sigma = \frac{\sqrt{k(G-k)}}{G}, \tag{4}$$

independently of any baseline*, and $g = 0$ when $k \in \{0, G\}$. The scalar coefficient is the group's empirical reward standard deviation; the direction $\bar{s}_+ - \bar{s}_-$ contrasts the correct and incorrect responses. For a one-dimensional Bernoulli-logit prompt $p = \varsigma(\theta)$, where $s_i = y_i - p$ and $\bar{s}_+ - \bar{s}_- = 1$, this is the scalar form*

$$g(k) \;=\; \frac{\sqrt{k(G-k)}}{G} \;=\; \sigma. \tag{5}$$

Proof. With binary rewards the standardized advantage takes two values, $\hat{A}_+ = (1-\mu)/\sigma = (G-k)/(G\sigma)$ on the k correct responses and $\hat{A}_- = -\mu/\sigma = -k/(G\sigma)$ on the $G-k$ incorrect ones, with $\mu = k/G$. Since $\sum_{i:R_i=1} s_i = k\,\bar{s}_+$ and $\sum_{i:R_i=0} s_i = (G-k)\,\bar{s}_-$,

$$g = \frac{1}{G}\left(\hat{A}_+ k\,\bar{s}_+ + \hat{A}_-\,(G-k)\,\bar{s}_-\right) = \frac{k(G-k)}{G^2\sigma}\,(\bar{s}_+ - \bar{s}_-) = \sigma\,(\bar{s}_+ - \bar{s}_-),$$

the last step using $k(G-k)/(G^2\sigma) = \sigma$ from $\sigma = \sqrt{k(G-k)}/G$. Adding any baseline b to the rewards shifts every $\hat{A}_i$ equally and cancels because $\sum_i \hat{A}_i = 0$. The scalar form follows from $\bar{s}_+ - \bar{s}_- = (1-p) - (-p) = 1$. □

> **The group-standard-deviation identity.** The update factors cleanly. Its direction $\bar{s}_+ - \bar{s}_-$ points from the incorrect rollouts toward the correct ones; its length is the group's reward standard deviation $\sigma = \sqrt{k(G-k)}/G$, zero for a unanimous group and largest for an even split. This is the paper's central lens: for binary rewards the standard deviation is not a denominator used for normalization but the prompt's learning signal.

Averaging the identity over random groups gives an *exact* expected gradient, whose limit is the established large-group arcsine reading.

Proposition 2 (Exact expected gradient)**.** *For the scalar prompt* (5) *with $k \sim \mathrm{Binomial}(G, p)$, the expected per-prompt gradient is the exact finite sum*

$$\mathbb{E}[g] = \frac{1}{G}\sum_{k=1}^{G-1} \binom{G}{k} p^k (1-p)^{G-k} \sqrt{k(G-k)}. \tag{6}$$

Corollary 1 (Large-group attenuation, asymptotic)**.** *As $G \to \infty$,*

$$\mathbb{E}[g] = \sqrt{p(1-p)}\left(1 - \frac{1}{8Gp(1-p)} + O(G^{-2})\right), \tag{7}$$

the arcsine gradient $\sqrt{p(1-p)}$ attenuated by the relative factor $1/(8Gp(1-p))$. This is the interior expansion of the exact sum (6)*; it loses accuracy near $p \in \{0, 1\}$, where the unanimous mass $p^G + (1-p)^G$ is not negligible. At $G = 8$, $p = 0.05$ the exact realized fraction $\mathbb{E}[g]/\sqrt{p(1-p)}$ is* 0.54*, against the expansion's* 0.67.

Proof. A second-order delta-method expansion of $f(\mu) = \sqrt{\mu(1-\mu)}$ about $\mu = p$, with $f''(p) = -\frac{1}{4}(p(1-p))^{-3/2}$ and $\mathrm{Var}(\mu) = p(1-p)/G$, gives $\mathbb{E}[f(\mu)] \approx \sqrt{p(1-p)} - \frac{1}{8G}(p(1-p))^{-1/2}$; the excluded unanimous terms are $O(p^G + (1-p)^G)$. □

The number σ is computed from the sampled group alone; it does not require knowledge of the policy's true success probability p. It is therefore an observable training diagnostic: before estimating any global difficulty, the sampled rollouts already reveal how much signal the prompt produced. Figure 3 (left) confirms the scalar form: the gradient lands on $\sqrt{k(G-k)}/G$, and the Monte-Carlo markers at three baselines coincide because the baseline cancels in (4). Averaging over groups gives the arcsine gradient $\sqrt{p(1-p)}$, attenuated at finite G by Corollary 1. The identity itself is the finite-G statement used by an actual training step.

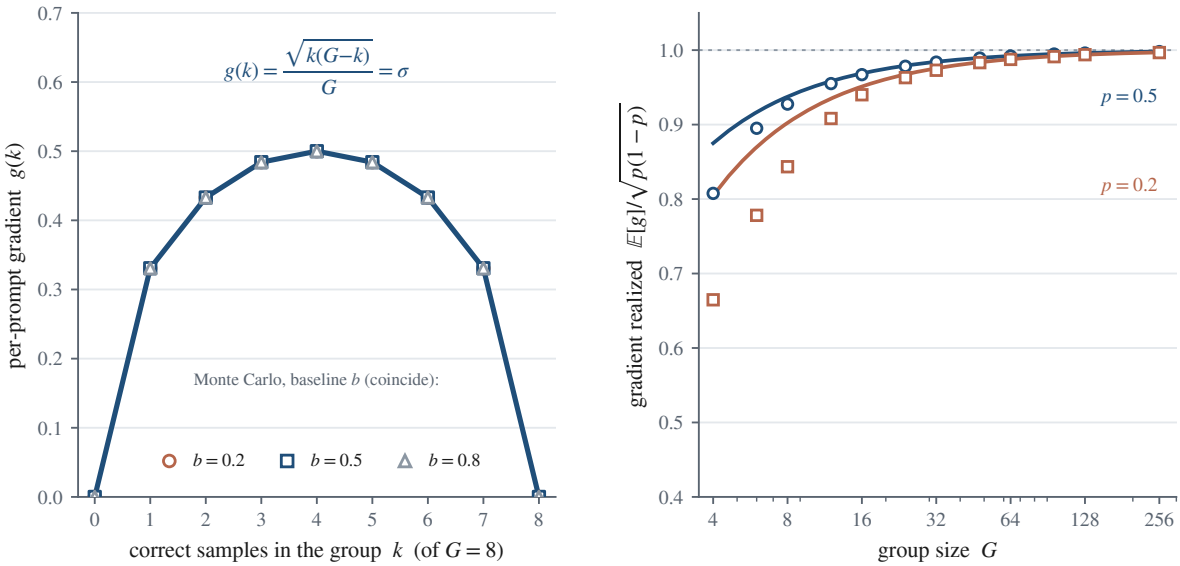

Figure 3: The group-standard-deviation identity. *Left:* the per-prompt gradient at $G = 8$, zero when all samples agree and largest at an even split, landing on $\sqrt{k(G-k)}/G$; Monte-Carlo markers at three baselines $b \in \{0.2, 0.5, 0.8\}$. *Right:* the realized fraction $\mathbb{E}[g]/\sqrt{p(1-p)}$ against the asymptotic law $1 - 1/(8Gp(1-p))$ of Corollary 1.

4 How Many Samples a Prompt Needs

Group size is usually fixed before training, often by convention ($G = 8$ or 16) rather than by calculation. Corollary 1 turns this choice into a difficulty-dependent quantity. The question becomes: how many samples are needed before a finite group behaves like the large-group limit? Define the *gradient fidelity*

$$\varphi(G,p) \;=\; \frac{\mathbb{E}[g]}{\sqrt{p(1-p)}} \;\in\; [0,1], \tag{8}$$

as the fraction of the large-group (arcsine) gradient realized by a group of size G at difficulty p. It approaches 1 as $G \to \infty$, and the interior expansion (7) gives $\varphi \approx 1 - 1/(8Gp(1-p))$. Solving this expression for G gives the group-size law.

Corollary 2 (The group-size law)**.** *A group of size G realizes interior fidelity $\varphi \geq 1 - \varepsilon$ at difficulty p once*

$$G \;\geq\; G^\star(\varepsilon,p) \;=\; \frac{1}{8\,\varepsilon\,p(1-p)} \;=\; \frac{w(p)^2}{8\,\varepsilon}, \qquad w(p) = \frac{1}{\sqrt{p(1-p)}}. \tag{9}$$

The budget is the mid-difficulty cost $1/(2\varepsilon)$ *times a difficulty penalty* $1/[4p(1-p)] \geq 1$*: the requirement grows as the* square *of the difficulty weight* $w(p)$ *that returns as GRPO's reweighting in §6. The penalty is* 1 *at* $p = \frac{1}{2}$*,* 2.8 *at* $p = 0.1$*, and* 5.3 *at* $p = 0.05$*. Near* $p \in \{0,1\}$ *the unanimous mass* $p^G + (1-p)^G$ *(§5) costs more than the interior expansion accounts for, so* $G^\star$ *understates the requirement there; the exact group size sits beside* $G^\star$ *in Table 2.*

Table 2: The group-size law, exact against closed form. Each entry is the exact group size at which the gradient fidelity φ of (8) reaches the column target, with the closed-form budget $G^\star = 1/(8\varepsilon p(1-p))$ of (9) in parentheses.

	group size G for fidelity $\varphi \geq 1-\varepsilon$		
difficulty p	90%	95%	99%
0.50	7 (5)	11 (10)	51 (50)
0.30 / 0.70	9 (6)	14 (12)	61 (60)
0.10 / 0.90	22 (14)	36 (28)	144 (139)
0.05 / 0.95	42 (26)	69 (53)	273 (263)

The law can be read from a table rather than swept. Three cases summarize the effect. A coin-flip prompt is sample-efficient: $G = 11$ already realizes 95% of the large-group gradient, and the conventional $G = 8$ realizes 93%. A very hard prompt is more expensive: a prompt solved 5% of the time needs $G = 69$ for the same 95% fidelity, roughly six times the coin-flip budget, and at $G = 8$ realizes only about half of its large-group gradient ($\varphi = 0.54$). This is the same lens again: if the group rarely contains both right and wrong answers, more samples are needed before the standard deviation reveals a stable learning signal. The closed form is most accurate in the high-fidelity regime that forces large groups: at 99% fidelity and $p = 0.05$, $G^\star$ gives 263 against the exact requirement 273. The discrepancy is concentrated at small groups near the extremes, where unanimous groups remain common. Figure 4 shows both views: fidelity rising to one with G (left), and the budget's difficulty bathtub (right). The practical reading is simple: one uniform group size spends too much on mid-difficulty prompts and too little on the hard tails, precisely where the silent-group rate of §5 is largest. The same budgeting question returns one stage later at inference, governed by a different mechanism: when a fixed prompt is answered by drawing many responses and returning one, the limiting factor is not within-group difficulty but how strongly the responses agree, so the independent-draw accounting here gives way to a correlation ceiling that caps selection from extra samples even while coverage keeps climbing [12].

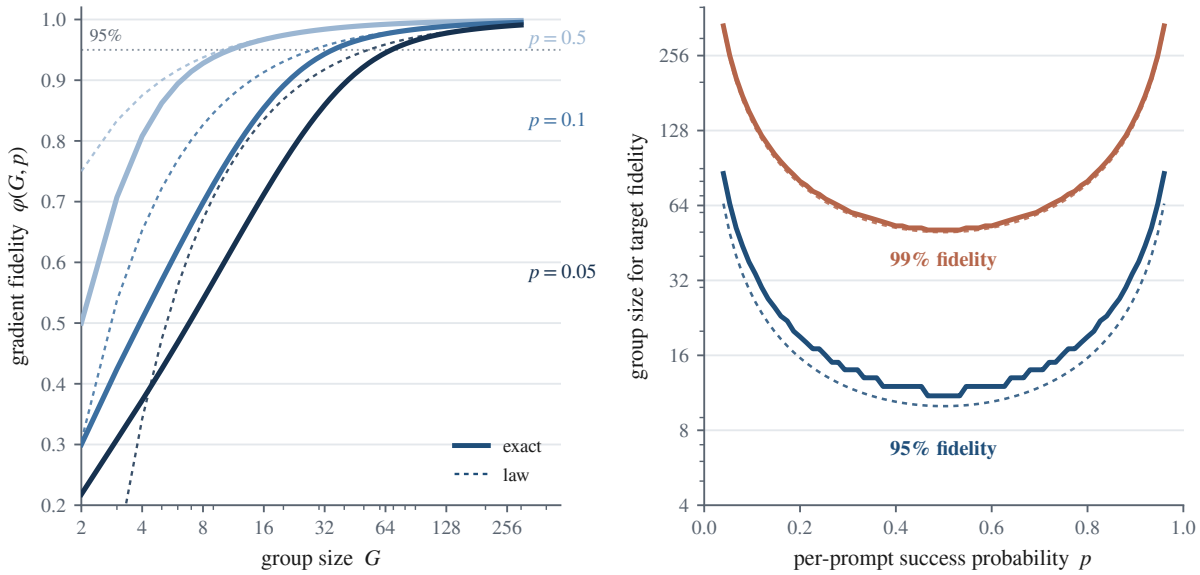

Figure 4: The group-size law. *Left:* the gradient fidelity $\varphi(G,p) = \mathbb{E}[g]/\sqrt{p(1-p)}$ against group size for four difficulties, with the closed form $1 - 1/(8Gp(1-p))$ dashed. *Right:* the group size required for 95% and 99% fidelity against difficulty, exact (solid) and the law $G^\star$ (dashed).

5 Silent Groups, and What DAPO Discards

The identity (4) sets $g = 0$ exactly when a group is unanimous. A *silent group* does not mean that the prompt is unimportant; it means that this particular sampled group produced no contrast between right and wrong responses. In the student analogy, all attempts are either failures or successes, so there is no within-prompt comparison to learn from. Thus the second practical decision, which groups to keep, also has a closed form. If every sampled answer is correct or every sampled answer is wrong, the reward standard deviation is zero, the standardized advantage is undefined, and the prompt contributes no gradient. The probability of this event, the *silent-group rate*, is

$$\mathbb{P}[\text{group silent}] \;=\; p^G + (1-p)^G, \tag{10}$$

which is large for easy or hard prompts. At any interior p it decays geometrically in G, but at $p \in \{0,1\}$ it is pinned at 1: no group size can produce signal when the policy is always wrong or always right (Figure 5). This is exactly the failure mode targeted by DAPO's *dynamic sampling* [8]: DAPO over-samples and discards prompts whose group accuracy is 0 or 1, keeping only groups with $0 < k < G$. In this notation, dynamic sampling is simply a keep rule on the same scalar: retain the group when $\sigma > 0$ and replace it when $\sigma = 0$. Discarding is not the only response to a silent group: a fixed-reference sign advantage instead assigns a nonzero update to a unanimous group, scoring each response against a constant rather than the group mean [13].

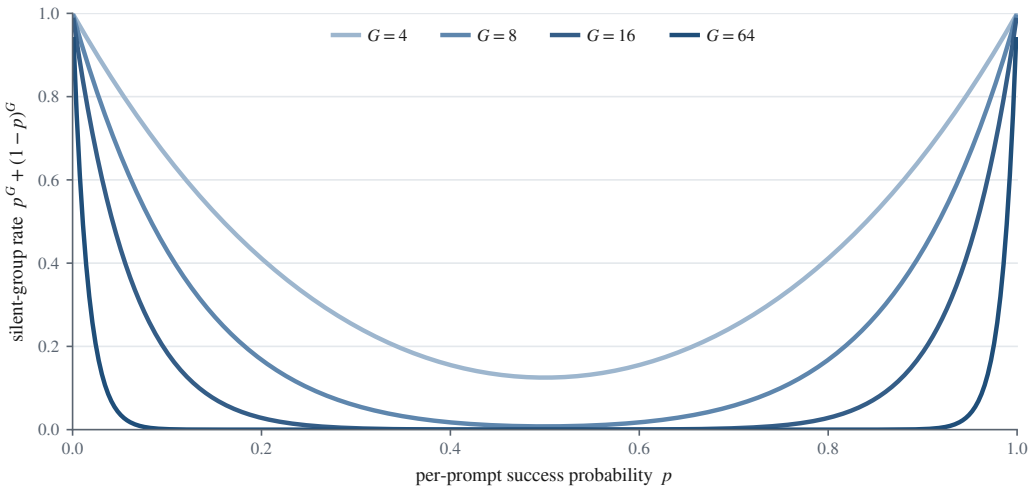

Figure 5: The silent-group rate $p^G + (1-p)^G$ of (10). Groups are most often silent near the easy and hard extremes, where all sampled answers tend to share the same reward. Larger G reduces the silent mass in the interior but cannot remove the endpoints $p = 0$ and $p = 1$.

The same accounting on a real run. The comparison with DAPO has two levels. The first is structural and requires no fitting. DAPO [8] keeps only groups with $0 < k < G$, so the discarded groups are exactly the silent mass in (10). The fraction logged in Fig. 3b of DAPO (the avg@32 all-correct rate) is the all-correct component $\mathbb{E}_p[p^{32}]$, and the corresponding training-time discarded component at $G = 16$ is $\mathbb{E}_p[p^{16}]$. The second level concerns the curve over training time. As training moves the Big-Math difficulty distribution toward mastery and the mean solve rate increases, $\mathbb{E}_p[p^{32}]$ traces the shape of the logged curve with one free timescale, at $R^2 = 0.92$ (bootstrap 0.91–0.96; Figure 6, left). This is a consistency check rather than a unique fit: a generic two-parameter saturating curve fits the same points at $R^2 = 0.98$. The closed form nevertheless explains the qualitative shape without another mechanism. The discarded fraction rises because more prompts become all-correct, yet it saturates below one because a persistent contested mass remains (Figure 6, right).

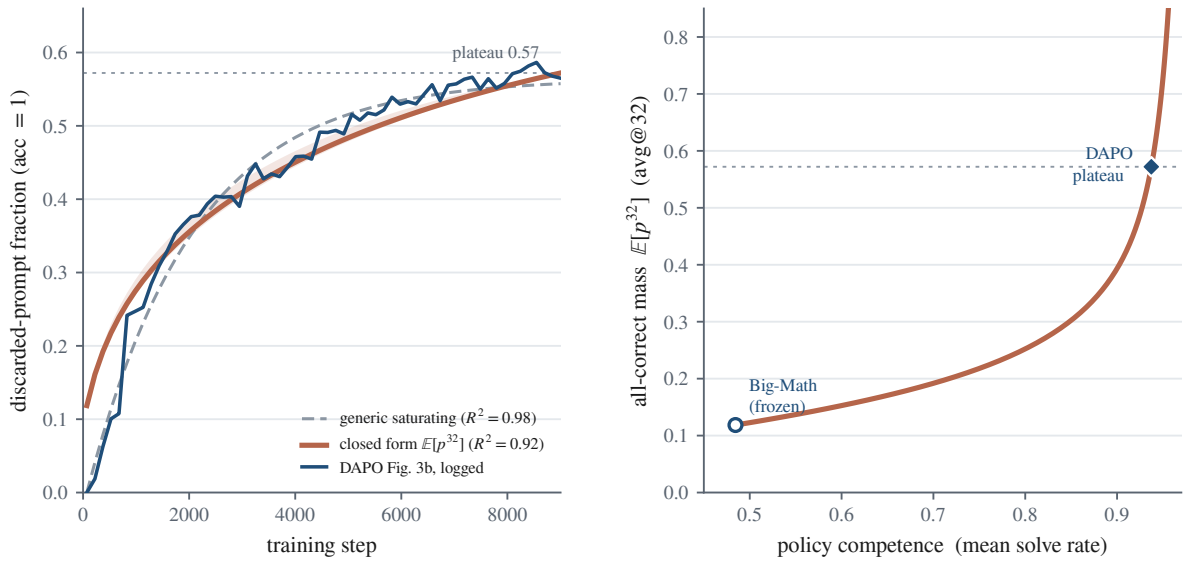

Figure 6: DAPO's discarded-prompt fraction and the closed form. *Left:* DAPO's logged accuracy-1 fraction (avg@32, Fig. 3b [8]), the closed-form $\mathbb{E}_p[p^{32}]$ fit with bootstrap 10–90% band, and a generic saturating baseline. *Right:* the closed-form all-correct mass $\mathbb{E}_p[p^{32}]$ versus policy competence on Big-Math, with the frozen anchor and DAPO's plateau.

6 What the Division Optimizes

Averaging the identity over groups recovers the known large-group picture and makes the difference between GRPO and Dr. GRPO exact. The single operation "divide by σ" does not merely rescale a step; it changes the implicit objective. By (6), as $G \to \infty$ the per-prompt GRPO gradient tends to $\sqrt{p(1-p)}$, the gradient of $2\arcsin\sqrt{p}$; Dr. GRPO tends to $p(1-p)$, the gradient of p. This is the surrogate-reward reading [7]: GRPO ascends $\mathbb{E}_x[2\arcsin\sqrt{p_x}]$, while Dr. GRPO ascends the raw success rate $\mathbb{E}_x[p_x]$. Both objectives have the same endpoint (all prompts solved), but they allocate training pressure differently before that endpoint is reached. Figure 7 plots both: the left panel contrasts the two objectives, and the right overlays Monte-Carlo gradients on the closed forms. The remaining gap on the GRPO curve is precisely the finite-G attenuation of (6).

The difficulty bias is the derivative. Liu et al. [4] (Dr. GRPO) observed empirically that standard-deviation normalization induces a "question-level difficulty bias": very easy and very hard prompts are over-weighted relative to medium ones. Removing the normalization removes that bias. The large-group limit makes the mechanism exact. The weight GRPO places on an incremental improvement at difficulty p is the derivative of the transform,

$$w(p) \;=\; \frac{\partial}{\partial p}\, 2\arcsin\sqrt{p} \;=\; \frac{1}{\sqrt{p(1-p)}}, \tag{11}$$

the same $1/\sigma$ that appears in the advantage (2). This weight has a bathtub shape (Figure 8): it is smallest at $p = \frac{1}{2}$, where $w = 2$, and diverges like $p^{-1/2}$ and $(1-p)^{-1/2}$ near $p = 0$ and $p = 1$. A prompt solved 5% or 95% of the time receives $w \approx 4.6$, more than twice the weight of a coin-flip prompt. Dr. GRPO sets $w \equiv 1$, which integrates back to the raw objective p. The difficulty bias is therefore not an accidental side effect. It is the price of using the variance-stabilizing transform: estimator variance is equalized by assigning extra weight to the extremes, where the raw success rate changes slowly.

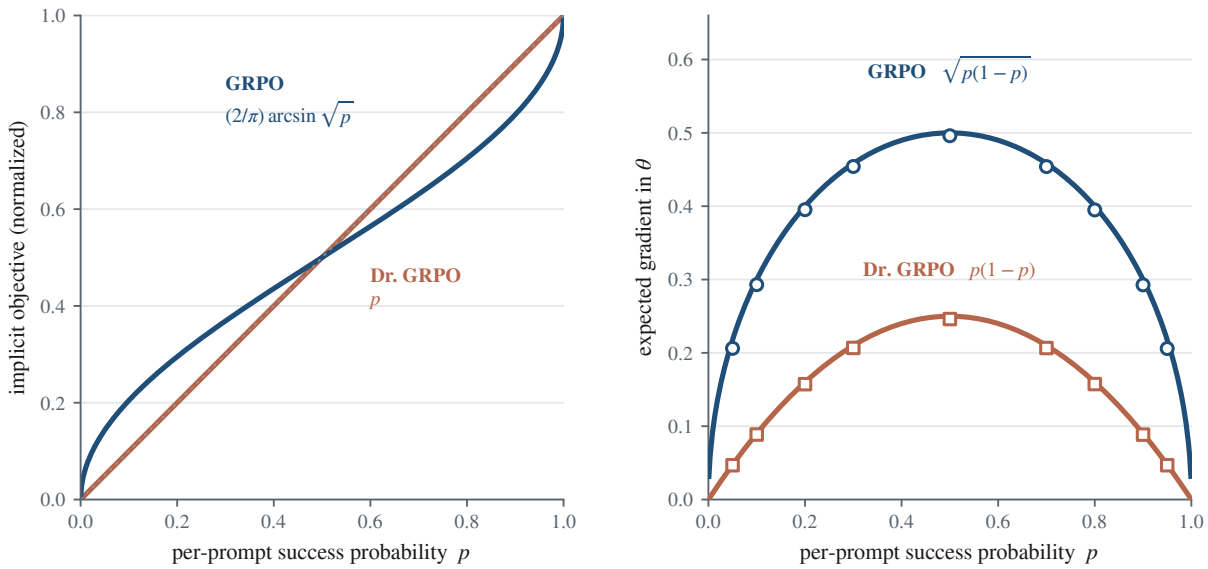

Figure 7: The large-group limit of the identity. *Left:* raw success rate p (Dr. GRPO) against the arcsine transform $\frac{2}{\pi}\arcsin\sqrt{p}$ (GRPO). *Right:* expected per-prompt gradient: closed-form curves with Monte-Carlo markers over groups of size $G = 64$ on a Bernoulli-logit prompt.

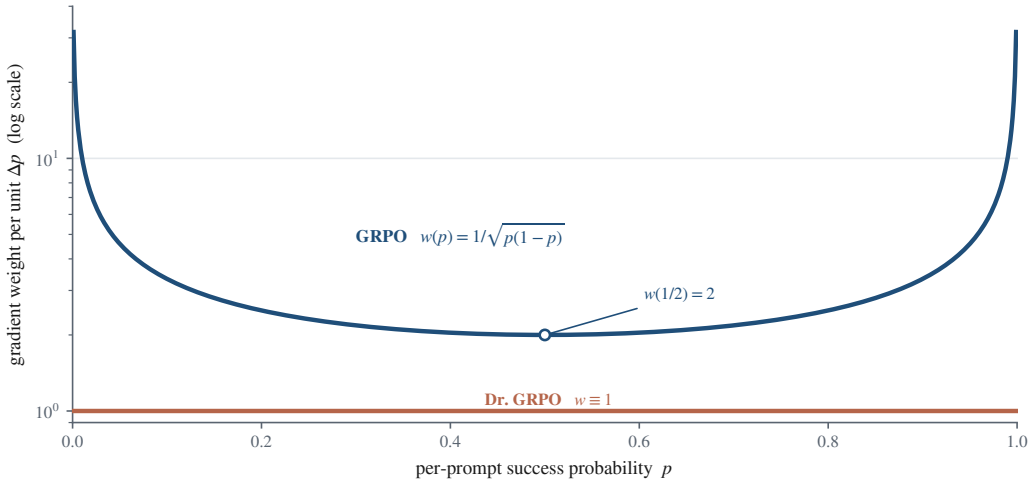

Figure 8: The difficulty weight $w(p) = \partial_p 2\arcsin\sqrt{p} = 1/\sqrt{p(1-p)}$ of (11) (log scale), against the flat weight $w \equiv 1$ of Dr. GRPO. GRPO assigns extra marginal weight to the easiest and hardest prompts, while Dr. GRPO weights each unit of raw success-rate improvement equally.

7 The Lens on Real Difficulty Data

The closed forms above are functions of difficulty p, so their practical effect depends on the *distribution* of p in real data. Big-Math [9] provides such a distribution: $N = 215{,}608$ competition and textbook problems, each annotated with the empirical solve rate $\hat{p} = k/64$ of Llama-3.1-8B over 64 rollouts.[1] The distribution

[1]The ungated `open-r1/Big-Math-RL-Verified-Processed` mirror is used. Llama-3.1-8B's per-prompt solve rate is treated as a stand-in for a policy's success probability p_x; it is a realistic, fixed difficulty distribution, not a specific training run. All

(Figure 9, left) is sharply bimodal: 4.0% of problems are never solved ($\hat{p} = 0$) and 7.2% are always solved ($\hat{p} = 1$), with broad mass in between. Thus the extreme-difficulty regime is not a theoretical corner case; it occupies a visible part of the corpus, precisely where reweighting and silent groups matter most.

The reweighting, quantified. In the large-group limit (7), each prompt contributes per-prompt gradient mass $\propto \sqrt{p(1-p)}$ under GRPO and $\propto p(1-p)$ under Dr. GRPO, the gradients of the two implicit objectives. Normalizing each over the corpus gives the share of the total gradient budget spent at each difficulty (Figure 9, right, and Table 3). Relative to Dr. GRPO, GRPO's standardization nearly doubles the share allocated to extreme-difficulty prompts (13.9% $\rightarrow$ 24.7%, a factor 1.78) and reduces the share allocated to medium-difficulty prompts (22.8% $\rightarrow$ 17.5%). At finite G the realized shift is milder, because the attenuation of Corollary 1 discounts the same extremes where silent groups are common; the controlled run of §8 measures 14.3% $\rightarrow$ 17.0% at $G = 8$, approaching the large-group 24.7% as G grows. In either view, the difficulty bias is not marginal on this corpus; it reallocates a visible share of the training signal.

Silent groups, quantified. Using the silent-group rate (10) on the same $\hat{p}$ histogram, Table 4 reports the fraction of prompts that yield no signal as a function of G. At the common choice $G = 8$, 44% of prompts produce no GRPO gradient; even at $G = 64$, 17% do not. An irreducible 11.2% ($\hat{p} \in \{0, 1\}$) are silent at every G: no amount of additional sampling creates a right-versus-wrong contrast for prompts that are always wrong or always right under the logged policy. This is the mass that dynamic sampling [8] must replace by over-sampling. The closed form is not only an assumption: drawing size-G groups directly from the 64 logged rollouts per problem, with no Bernoulli model, gives a measured silent fraction within two points of $p^G + (1-p)^G$ for G well below the rollout budget (43% against 44% at $G = 8$; Table 4, lower row), the small gap being the finite-pool correction.

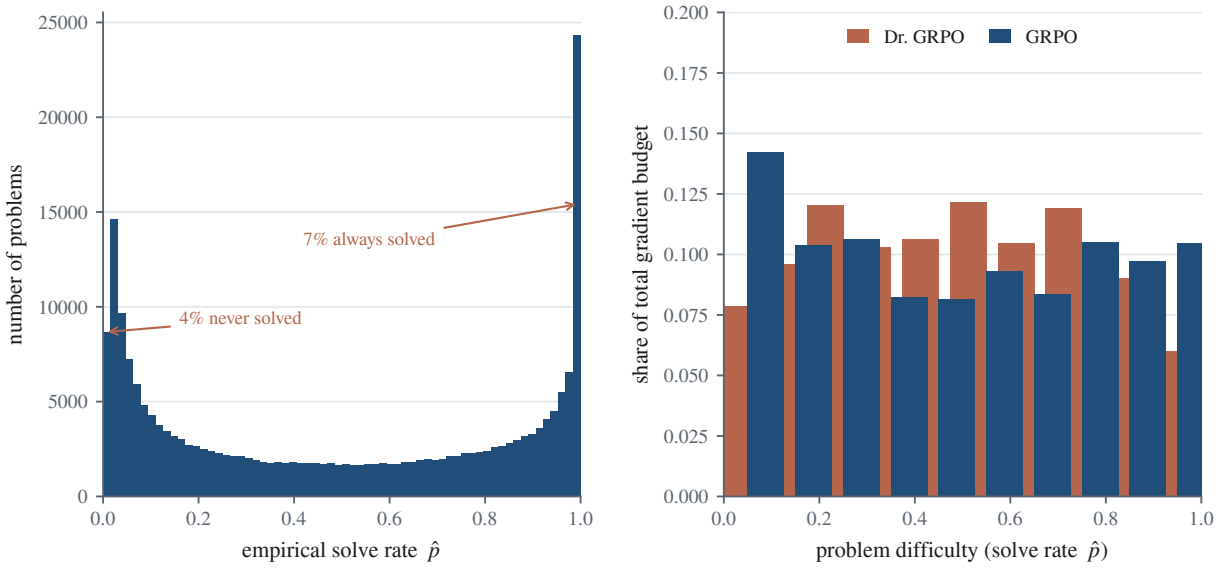

Figure 9: Real difficulty distribution and the standardization reweighting (Big-Math, $N = 215{,}608$). *Left:* histogram of empirical solve rates $\hat{p}$ (Llama-3.1-8B, 64 rollouts). *Right:* share of per-prompt gradient budget by difficulty under Dr. GRPO ($\propto p(1-p)$) and GRPO ($\propto \sqrt{p(1-p)}$).

numbers below are exact functions of the published $\hat{p}$ histogram.

Table 3: Share of total gradient mass by prompt difficulty on Big-Math, GRPO versus Dr. GRPO.

gradient budget on	Dr. GRPO $p(1-p)$	GRPO $\sqrt{p(1-p)}$
extreme ($\hat{p}<.1$ or $>.9$)	13.9%	24.7%
medium ($.4\leq\hat{p}\leq.6$)	22.8%	17.5%

Table 4: Fraction of Big-Math prompts whose group is silent ($\sigma = 0$), by group size G: the closed form (10) against direct subsampling of the 64 logged rollouts.

group size G	4	8	16	32	64
closed form	59%	44%	32%	23%	17%
subsampled	59%	43%	30%	–	–

8 The Lens in a Live Training Run

The preceding sections use static accounting: for a fixed difficulty p, the formulas predict gradient mass, silent groups, and group-size requirements. This section checks whether the same accounting remains visible during training. The setup is fully reproducible: $M = 6{,}000$ prompts, each a one-dimensional Bernoulli-logit policy $p_x = \varsigma(\theta_x)$ with initial difficulty drawn from the real Big-Math solve-rate distribution, are trained for 150 steps at group size $G = 8$ under three advantage rules: GRPO (divide by σ), Dr. GRPO (do not divide), and DAPO (resample degenerate groups). At each step every prompt draws a fresh group, forms the advantage, and updates θ_x by the actual sampled gradient. Three predictions are compared with the measured run in Figure 10.

The silent-group rate predicts wasted groups. Across the whole run the measured fraction of unanimous groups tracks the closed form $\mathbb{E}[p^G + (1-p)^G]$ evaluated at the current difficulty, with $R^2 = 0.999$ (Figure 10a). The fraction *rises* toward one as training proceeds, because mastered prompts ($p \to 1$) increasingly become all-correct. The same all-correct mass that climbs in DAPO's logged run (§5) is reproduced here from the identity alone.

The reweighting predicts which difficulties dominate. Binning the realized gradient mass by difficulty, GRPO and Dr. GRPO match their finite-G closed forms exactly (Figure 10b): GRPO spends 17.0% of its gradient mass on the extreme prompts ($\hat{p} < 0.1$ or > 0.9) against Dr. GRPO's 14.3%. This is the finite-G realized version of the large-group 24.7% of §7; the gap is the attenuation of Corollary 1, which discounts the very extremes where silent groups concentrate.

The difficulty bias is visible in the trajectories. Because the GRPO step scales like σ while the Dr. GRPO step scales like σ^2, GRPO moves relatively faster where σ is small, namely near the unanimous extremes. The effect is dynamic (Figure 10c): GRPO lifts the initially-hardest quartile of prompts to a mean solve rate of 0.99 over the run, whereas Dr. GRPO reaches 0.88; DAPO, which never spends an update on a silent group, is fastest, at the cost of $3.5\times$ oversampling concentrated on those same extremes. Thus the bias is not only a static reallocation of gradient budget. It changes the learning trajectory, especially for prompts that begin near the hard end of the difficulty scale.

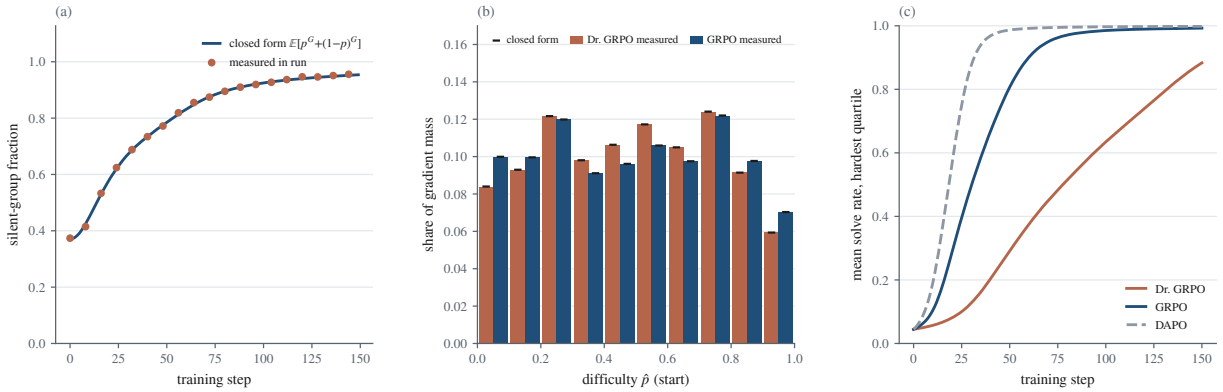

Figure 10: The identity's predictions in a controlled GRPO run ($M = 6{,}000$ Bernoulli-logit prompts, Big-Math initial difficulty, $G = 8$). *(a)* measured silent-group fraction against the closed form over training. *(b)* realized gradient mass by difficulty, measured against the finite-G closed form. *(c)* mean solve rate of the initially-hardest quartile under GRPO, Dr. GRPO, and DAPO.

9 Related Work

GRPO and its critic-free relatives. Group Relative Policy Optimization (GRPO) was introduced for mathematical reasoning [1] and popularized by R1-style training [2]. Its closest relatives drop the critic in favor of group or leave-one-out baselines: RLOO and "back-to-basics" REINFORCE [10, 11], which the identity (4) subsumes through the rescaling of Proposition 1. Liu et al. [4] propose Dr. GRPO, identifying a length bias and a question-level difficulty bias in GRPO and removing both the length and standard-deviation normalizers; §6 above expresses their difficulty bias exactly as $\partial_p 2\arcsin\sqrt{p}$. DAPO [8] adds dynamic sampling, which §5 writes as removing the silent-group mass (10). A parallel line keeps the group but replaces the group-relative center with a constant reference, so unanimous groups still produce a signal [13]; a related reading recasts the same correct-minus-incorrect contrast as an implicit preference objective [14].

The large-group surrogate-reward view. The asymptotics of the standard-deviation division are well understood. Thrampoulidis et al. [7], in a study of Pass@K advantage shaping, establish that GRPO is (up to clipping) RLOO applied to the surrogate reward $2\arcsin\sqrt{p}$, and identify this with the binomial variance-stabilizing transform [5, 6]; related analyses read GRPO as a process reward model [15], give a local-curvature/adaptive-gradient account of the normalizer [16], and derive REINFORCE, PPO, and GRPO from a common expected-reward objective [17]. The group-standard-deviation identity (4) supplies the exact finite-group object whose group average (6) becomes this surrogate gradient. It holds for a single group of any size G in any dimension, depends on no baseline, and turns two open design choices into closed forms: the group-size law (9) and the silent-group rate (10). Those quantities are then measured on a real 215,608-problem corpus in §7 and confirmed as training dynamics in §8.

Variance-stabilizing transforms. The arcsine transform $2\arcsin\sqrt{p}$ that stabilizes the variance of a binomial proportion is classical [5, 6] and standard in the analysis of binomial and count data. It appears here as the large-group limit (6) of the per-group standard deviation, the object the GRPO update is shown to equal.

Lineage. Methodologically this paper follows the tradition of explaining a widely used but under-theorized method by exhibiting a closed-form equivalence to a classical object, as Levy and Goldberg [18] did for word2vec [19], and as the closed-form, search-free treatment of Bay and Yearick [20] does for scaling laws.

10 Discussion

The practical message is not that one existing method is universally best. The identity separates three design choices that are often discussed together. First, the objective choice: dividing by σ gives GRPO the variance-stabilized arcsine objective, while removing the division gives Dr. GRPO the raw success-rate objective. Second, the compute choice: groups with $\sigma = 0$ have no right-versus-wrong contrast, so DAPO-style dynamic sampling avoids spending updates on them. Third, the sampling choice: the group-size law says how large G must be at a given difficulty before the finite group faithfully realizes the large-group signal.

- **The signal is disagreement.** By (4), the prompt's instantaneous learning signal is computable from the sampled rollouts alone. A mixed group teaches because it contains both sides of the comparison. A unanimous group is silent because there is no within-prompt contrast.
- **The methods are operations, not mysteries.** GRPO scales by σ, Dr. GRPO omits that scaling, and DAPO skips groups with $\sigma = 0$. This makes the algorithmic landscape easier to reason about than a list of unrelated heuristics.
- **Group size is a difficulty-dependent budget.** The fidelity (8) and silent-group rate (10) describe the two things G buys: how much of the large-group gradient a retained group realizes, and how often a usable mixed group appears. A single uniform G can be reasonable for simplicity, but the law explains why it underserves the easy and hard extremes.
- **Standardization is a modeling choice.** Dividing by σ trades raw-accuracy alignment for a variance-stabilized objective that gives more marginal weight to extreme difficulties. This can help rescue hard prompts, but it is also the source of the difficulty bias. The right choice depends on whether the training goal prioritizes raw success-rate alignment, hard-prompt pressure, or compute efficiency.

More broadly, reading GRPO through the group standard deviation suggests a template for analyzing other reward-shaping choices: start with one prompt, one sampled group, and one exact update. Rank-based advantages, quantile advantages, reward clipping, and length normalization should each admit the same kind of single-group accounting.

Limitations. The identity (4) concerns the advantage construction under binary rewards and an on-policy first step. It deliberately sets aside clipping, the KL penalty, off-policy staleness, and non-binary rewards, each of which merits the same single-group treatment. The controlled run of §8 validates the closed forms as dynamics on a tractable Bernoulli-logit policy, where the score is scalar. A full language-model training loop that logs silent groups and per-difficulty gradient mass across G is the natural next test; the theorem predicts what such a run should find because (4) holds in any dimension. Whether the extra weight that standardization places on the hardest prompts improves generalization beyond the trained difficulty range is a further, downstream question, continuous with the broader study of when models extrapolate past their training distribution [21]. The DAPO comparison (§5) rests on the exact structural identification of the discarded mass; its time-evolution fit is a consistency check, not a unique prediction. A logged per-prompt solve-rate history would allow the all-correct curve to be predicted rather than fit. None of these limitations affects the core identity (4), which is exact for any single group of any size and any policy dimension.

11 Conclusion

During RLVR training, a prompt teaches through the pattern of rewards assigned to its sampled answers. For binary rewards, this paper shows that one GRPO step on a prompt equals the reward standard deviation of its sampled group, $\sqrt{k(G-k)}/G$, times the contrast between correct and incorrect response scores. The result is finite-group, baseline-free, and valid in any policy dimension. A prompt pushes hardest when its answers are split and gives no signal when all sampled answers agree.

This identity makes the relationship among GRPO, Dr. GRPO, and DAPO concrete. GRPO divides by the group standard deviation and thereby follows the variance-stabilized arcsine objective. Dr. GRPO drops the division and returns to the raw success-rate objective. DAPO's dynamic sampling removes the groups where the same standard deviation is zero. The methods are therefore not three disconnected tricks; they are three operations on one number.

Because the identity holds at the finite group sizes used in training, two practical quantities become closed forms. The group-size law, $G \gtrsim 1/(8\varepsilon p(1-p))$, gives the number of samples needed to realize a target fraction of the large-group gradient. The silent-group rate, $p^G + (1-p)^G$, gives the fraction of groups that provide no right-versus-wrong contrast. Both are borne out in a controlled run and on a 215,608-problem difficulty corpus. The group standard deviation, long read as a normalizer, is the size of the learning signal itself.

Reproducibility. All code and data are available at `https://github.com/bay-yearick-lab/grpo-standard-deviation-identity`. All claims, including the general identity (4), are verified numerically (`scripts/checks.py`), and the closed forms are exposed as a small diagnostic API (`scripts/grpo_diagnostics.py`). The controlled run of §8 is `scripts/experiment_dynamics.py`; figures and tables are regenerated from the public Big-Math solve-rate annotations by `scripts/make_figures.py` and `scripts/analyze_rollouts.py`. The DAPO comparison (§5) is digitized from the published figure by `scripts/digitize_dapo.py` and stored with provenance in `data/dapo/`.

References

[1] 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.

[2] DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[3] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

[4] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[5] M. S. Bartlett. The square root transformation in analysis of variance. *Supplement to the Journal of the Royal Statistical Society*, 3(1):68–78, 1936.

[6] F. J. Anscombe. The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, 35 (3–4):246–254, 1948.

[7] Christos Thrampoulidis, Sadegh Mahdavi, and Wenlong Deng. Advantage shaping as surrogate reward maximization: Unifying Pass@K policy gradients. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=R1RhBFUk8t.

[8] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[9] Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, and Nick Haber. Big-Math: A large-scale, high-quality math dataset for reinforcement learning in language models. *arXiv preprint arXiv:2502.17387*, 2025.

[10] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 12248–12267, 2024.

[11] Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! *ICLR Deep RL Meets Structured Prediction Workshop*, 2019.

[12] Yong Yi Bay and Kathleen A. Yearick. When more sampling hurts: The modal ceiling and correlation ceiling of test-time scaling. *arXiv preprint arXiv:2606.28661*, 2026.

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

[14] Yihong Wu, Liheng Ma, Lei Ding, Muzhi Li, Xinyu Wang, Kejia Chen, Zhan Su, Zhanguang Zhang, Chenyang Huang, Yingxue Zhang, Mark Coates, and Jian-Yun Nie. It takes two: Your GRPO is secretly DPO. *arXiv preprint arXiv:2510.00977*, 2025.

[15] Michael Sullivan and Alexander Koller. GRPO is secretly a process reward model. *arXiv preprint arXiv:2509.21154*, 2025.

[16] Cheng Ge, Caitlyn Heqi Yin, Hao Liang, and Jiawei Zhang. Why GRPO needs normalization: A local-curvature perspective on adaptive gradients. *arXiv preprint arXiv:2601.23135*, 2026.

[17] Jianghan Shen, Siqi Luo, Yue Li, Jiyao Liu, Wanying Qu, Yi Zhang, Ziyan Huang, Tianbin Li, Ming Hu, Xiaohong Liu, Yirong Chen, and Junjun He. A first-principles derivation of LLM policy optimization: From expected reward to GRPO and its structural extensions. *arXiv preprint arXiv:2606.16733*, 2026.

[18] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, pages 2177–2185, 2014.

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[20] Yong Yi Bay and Kathleen A. Yearick. Solve for the hyperparameter, skip the search: Kolmogorov-optimal scaling laws for spline regression. *arXiv preprint arXiv:2606.23575*, 2026.

[21] Yong Yi Bay and Kathleen A. Yearick. Machine learning vs deep learning: The generalization problem. *arXiv preprint arXiv:2403.01621*, 2024.