DATAFLEX-RL: AN EVALUATION PLATFORM FOR RLVR DATA POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Data policies for reinforcement learning with verifiable rewards (RLVR) change which rollouts are used, how strongly they are weighted, or which domains supply the next batch. We introduce DATAFLEX-RL, an evaluation platform for comparing these choices under the same GRPO recipe. Our primary experiment evaluates 13 configurations with 12 matched seeds on Qwen2.5-7B-base and 12 math, logic, and science benchmarks. Uniform GRPO improves the domain-balanced average accuracy (Overall) by 7.76 points over the untrained checkpoint. None of the eight selection or reweighting methods has a paired 95% confidence interval excluding zero relative to uniform sampling, and none of the three adaptive mixtures improves over a fixed equal mixture at that precision. A corrected 12-seed Llama-3.1-8B-base extension puts the additional methods on the same score scale as the original controls, without producing a common winner in their observed means. We also quantify evaluation sensitivity by recomputing nine Qwen2.5-7B-Instruct runs with a math-heavy six-benchmark summary—five math benchmarks plus GPQA-Diamond, with no logic benchmark—and with the domain-balanced 12-benchmark summary. Their rankings are negatively correlated ($\rho = -0.33$), while summaries retaining all 12 benchmarks largely agree. Across the controlled settings studied here, changing the data policy measurably changes the training process but does not yield a reproducible improvement over uniform training.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) is widely used to post-train reasoning models. In Group Relative Policy Optimization (GRPO) Shao et al. (2024), each update samples prompts, generates multiple responses, scores them with a verifier, and constructs group-relative advantages. This makes the training distribution an online choice: a data policy decides which prompts receive rollout compute, how domains are mixed, which groups survive filtering, and how strongly their responses contribute to the update.

Prompts that are solved by every rollout or by none of them yield degenerate groups, while prompts near the policy's decision boundary may provide informative comparisons. This observation has produced a diverse set of interventions: solve-rate filtering in DAPO Yu et al. (2025), high-variance down-sampling in PODS Xu et al. (2026), probability-based Advantage Reweighting Yang et al. (2025), advantage weighting inspired by prioritized experience replay Schaul et al. (2016), and adaptive domain curricula such as DUMP Wang et al. (2025) and teacher-student curriculum learning Matiisen et al. (2017). These methods alter different parts of the training loop, but all try to concentrate rollout or optimization effort on data judged more useful.

The appeal of these methods rests on a broader premise: directing training toward more useful data should produce gains that survive reasonable changes in model and training condition. Existing evidence does not yet establish that premise. Data-processing methods are often introduced as one component of a larger RLVR recipe, alongside changes to the base model, prompt template, verifier, rollout budget, and optimization settings. Even methods driven by the same signal may use it differently: advantage magnitude can define either a continuous loss weight or a hard selection rule. A reported gain can therefore reflect the signal, the intervention, or the surrounding training stack.

On-policy evaluation adds two further complications. The score attached to a prompt changes as the policy learns, so the data utility signal is both noisy and endogenous to training. At the same

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.

time, training-seed variation and measurement noise on small reasoning benchmarks can match or exceed the reported gap between methods. The evaluation summary can introduce another choice: omitting a domain or weighting benchmarks differently may change which method appears best. A training–evaluation format mismatch can also leave optimization traces apparently normal while making benchmark scores invalid. A useful comparison must control the training recipe, cover the intended evaluation domains, align the verifier and evaluation formats, and preserve matched seeds from training through final evaluation.

We therefore ask whether the tested RLVR data policies provide reproducible gains over uniform sampling when the surrounding GRPO recipe is held fixed. The evaluation begins with a complete 12-seed comparison on Qwen2.5-7B-base, where GRPO itself produces a large gain. This experiment tests all selection, reweighting, and mixture configurations in a setting with substantial training headroom. We then extend the comparison to Llama base models and several Qwen base-model sizes, and test whether the conclusion depends on evaluation coverage. Separate controls examine whether selection benefits come from the targeting signal, from changing the number of tokens used for optimization, or from intervention strength.

DATAFLEX-RL is an evaluation platform built around this comparison. It separates what a policy changes from the signal it uses and transfers the selection–reweighting–mixture abstraction of DATAFLEX Liang et al. (2026) from supervised LLM training to on-policy RL. Selection decides which generated responses enter the update, reweighting changes their continuous loss contribution, and mixture adaptation changes which domains supply future prompts. Solve rate, reward, advantage, and token probability are then recorded as signals rather than treated as method categories. This distinction matters experimentally: two methods may use similar scores while changing different quantities, including the effective number of update tokens. The platform pairs this taxonomy with a standardized math/logic/science corpus, fixed GRPO training, domain-specific evaluation harnesses, and a calibration smoke test for training–evaluation alignment.

The release links each configuration to its training log and 12-benchmark record (Figure 1). The experiments then test whether the primary result survives changes in model family, model scale, and evaluation coverage.

Contributions. We make three contributions:

1. **No clear advantage over uniform sampling in the tested setting.** We evaluate 13 selection, reweighting, and mixture configurations with 12 matched seeds on Qwen2.5-7B-base. Uniform GRPO improves by 7.76 points, but no selection or reweighting method shows a clear improvement over uniform sampling, and no adaptive mixture improves over a fixed equal mixture at the measured precision (Section 5).
2. **A direct measurement of evaluation sensitivity.** On the same nine Qwen2.5-7B-Instruct configurations, a math-heavy six-benchmark summary (Math-Heavy-6; five math sets plus GPQA-Diamond) and the domain-balanced 12-benchmark summary have negatively correlated rankings ($\rho = -0.33$), with method spread changing from 0.90 under Math-Heavy-6 to 3.31 under DB-12. Summaries that retain all 12 benchmarks largely agree, showing that omitting the logic domain can change the apparent winner (Section 6).
3. **A platform for fair comparisons.** DATAFLEX-RL isolates the data policy from the surrounding GRPO recipe through a shared rollout, verification, optimization, and evaluation stack, matched seeds, and a common 12-benchmark score format. The intervention taxonomy and shared driver let future policies be compared under the same protocol without rebuilding the pipeline (Section 4).

2 RELATED WORK

2.1 DATA PROCESSING FOR RLVR

RL with verifiable rewards. DeepSeekMath introduced GRPO, which computes group-relative advantages from multiple responses to one prompt Shao et al. (2024). DeepSeek-R1 and Qwen2.5-Math demonstrated strong reasoning improvements from outcome-verifiable RL DeepSeek-AI et al.

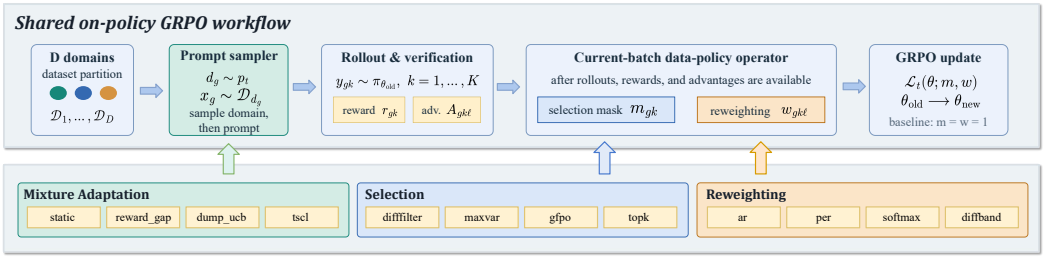

Figure 1: The shared GRPO workflow and the intervention point of each data-policy family. Mixture adaptation changes the domains supplying future prompts; selection and reweighting act after rollout and verification by modifying the current update. Uniform sampling with unit mask and weight is the common baseline.

(2025); Yang et al. (2024), and DAPO provided an open large-scale recipe Yu et al. (2025). These systems evaluate complete stacks; DATAFLEX-RL isolates the data-processing component.

Sample selection and reweighting. DAPO filters degenerate groups Yu et al. (2025); PODS retains a reward-variance-maximizing subset Xu et al. (2026); and GFPO uses reward-per-token Shrivastava et al. (2025). Advantage Reweighting changes token contributions Yang et al. (2025), while prioritized replay emphasizes high-surprise transitions Schaul et al. (2016). We compare these distinct interventions under matched models, data, rollout budgets, and seeds.

2.2 DATA-CENTRIC TRAINING

Offline selection and mixture optimization. DSIR, LESS, and related curation methods resample or score examples before training Xie et al. (2023b); Xia et al. (2024); Sorscher et al. (2022); Li et al. (2024). DoReMi and RegMix learn domain mixtures with proxy runs Xie et al. (2023a); Liu et al. (2024). These policies are fixed before the main run and do not respond to on-policy rewards.

Online curricula and unified systems. Skill-It, teacher–student curricula, and DUMP adapt sampling from observed progress Chen et al. (2023); Matiisen et al. (2017); Wang et al. (2025). DATAFLEX unifies selection, mixture optimization, and reweighting for supervised LLM training Liang et al. (2026); DATAFLEX-RL brings the abstraction to noisy, policy-dependent RL.

2.3 EVALUATION AND REPRODUCIBILITY

Evaluation infrastructure. The LM Evaluation Harness, HELM, and OpenCompass standardize prompts and scoring Gao et al. (2021); Liang et al. (2023); Cao et al. (2026). We retain their task-specific protocols for MATH, GSM8K, GPQA, MMLU-Pro, and ZebraLogic Yang et al. (2024); Hendrycks et al. (2021); Cobbe et al. (2021); Rein et al. (2023); Wang et al. (2024b); Lin et al. (2025), but compare matched training interventions rather than unrelated checkpoints.

Reproducible empirical RL. Task choice, implementation details, and seeds can change benchmark conclusions Dehghani et al. (2021); Henderson et al. (2018). Reliable comparisons therefore report variance, intervals, robust aggregation, and complete procedures Agarwal et al. (2021); Patterson et al. (2024); Pineau et al. (2021); our platform supplies these elements with run-level records.

3 THREE DATA-POLICY INTERVENTIONS

RLVR data policies differ first in the part of training they change. *Selection* removes generated responses or prompt groups from the current update. *Reweighting* keeps those responses but changes their loss contribution. *Mixture adaptation* changes the domain distribution used for subsequent prompt sampling. We describe these intervention points before discussing the signals that drive them.

3.1 SELECTION, REWEIGHTING, AND MIXTURE ADAPTATION

The baseline training step is simple: it samples math, logic, and science prompts in equal proportions, generates $K = 5$ responses for each prompt, verifies their rewards, and applies the standard GRPO loss to every valid response token. Each policy family below changes one part of this process.

Selection: choose which responses enter the update. Selection is applied after the responses and rewards have been generated. Let $m_{gk} \in \{0, 1\}$ indicate whether response k to prompt g is used for optimization. A zero mask removes that response from the update; a one mask leaves its GRPO loss unchanged. A group-level selector uses the same decision for all responses from one prompt. With L_{gk} denoting response length and $\ell_{gk\ell}^{\mathrm{GRPO}}$ the standard loss for token ℓ, the selected loss is

$$\mathcal{L}_{\mathrm{sel}} = \frac{1}{GK} \sum_{g=1}^{G} \sum_{k=1}^{K} \frac{m_{gk}}{L_{gk}} \sum_{\ell=1}^{L_{gk}} \ell_{gk\ell}^{\mathrm{GRPO}}. \tag{1}$$

The baseline sets every $m_{gk} = 1$. Our selection configurations are `diffilter`, `maxvar`, `gfpo`, and `topk`.

Reweighting: change how strongly each token or response is learned. Reweighting keeps generated responses in the update but changes how strongly they contribute. Let $w_{gk\ell} \geq 0$ be the weight on token ℓ. A response-level method uses one weight for all tokens in a response; a token-level method can assign different weights within that response:

$$\mathcal{L}_{\mathrm{rew}} = \frac{1}{GK} \sum_{g=1}^{G} \sum_{k=1}^{K} \frac{1}{L_{gk}} \sum_{\ell=1}^{L_{gk}} w_{gk\ell}\, \ell_{gk\ell}^{\mathrm{GRPO}}. \tag{2}$$

The baseline sets every $w_{gk\ell} = 1$; our reweighting methods normalize weights to have mean one over the relevant batch units. The reweighting configurations are `ar`, `per`, `softmax`, and `diffband`.

Mixture adaptation: change which domains supply future prompts. Mixture methods act before the next rollout. Let $\mathcal{D}_d$ be the prompt distribution for domain d, and let $p_t(d)$ be its sampling probability at step t, with $p_t(d) \geq 0$ and $\sum_{d=1}^{D} p_t(d) = 1$. The GRPO objective under this training-data mixture is

$$\mathcal{L}_{\mathrm{mix},t}(\theta) = \sum_{d=1}^{D} p_t(d)\, \mathbb{E}_{\substack{x \sim \mathcal{D}_d, \\ y_{1:K} \sim \pi_{\theta_{\mathrm{old}}}(\cdot|x)}} \left[\frac{1}{K} \sum_{k=1}^{K} \frac{1}{L_k} \sum_{\ell=1}^{L_k} \ell_{k\ell}^{\mathrm{GRPO}}(\theta) \right]. \tag{3}$$

The fixed control uses $p_t(d) = 1/D$ throughout training; with our three domains, this is $(1/3, 1/3, 1/3)$. A mixture method changes p_t over time but leaves the per-response GRPO loss unchanged. Thus, selection and reweighting modify the current update after rollout, whereas mixture adaptation changes which prompts generate the next set of rollouts. The next subsection and Table 1 specify how each implemented mixture method computes p_t.

The timing also clarifies the compute comparison. Selection and reweighting occur after rollout, so they do not reduce generation cost. Selection is not renormalized and can reduce the number of tokens used in the update, whereas reweighting preserves a mean weight of one. Mixture adaptation leaves the per-response update unchanged and changes only the source of future prompts.

3.2 IMPLEMENTED METHODS AND CONTROLLED COMPARISONS

The intervention family describes *what* a method changes; the signal describes *how* it chooses what to change. For example, a method may use the fraction of the five responses to a prompt that are correct, which we call the group solve rate q_g. Another method may score each response by its mean absolute GRPO advantage a_{gk}. Formally,

$$a_{gk} = \frac{1}{L_{gk}} \sum_{\ell=1}^{L_{gk}} |A_{gk\ell}|, \qquad q_g = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}[r_{gk} > 0.5], \tag{4}$$

Configuration	Family	Signal / operation
`baseline`	uniform control	unit token weight
Selection (fixed equal domain mixture)		
`difffilter`	selection	group solve rate q_g
`maxvar`	selection	within-group reward variance
`gfpo`	selection	reward per response token
`topk`	selection	mean absolute advantage
Reweighting (fixed equal domain mixture)		
`ar`	reweighting	token probability
`per`	reweighting	mean absolute advantage
`softmax`	reweighting	mean absolute advantage
`diffband`	reweighting	outcome reward
Mixture (no within-domain filtering)		
`static`	uniform control	$p_t(d) = 1/3$
`reward_gap`	mixture	rolling domain reward
`dump_ucb`	mixture	advantage and exploration count
`tscl`	mixture	domain reward slope

Table 1: The 13 configurations compared in the paper. Full intervention rules are specified in Section 3; implementation details are released with the run records.

where $A_{gk\ell}$ is the advantage of token ℓ and r_{gk} is the verified reward. We also use the rollout policy's token probability $\pi_{gk\ell}$ and the efficiency score r_{gk}/L_{gk}. Mixture methods summarize each domain over a rolling 50-observation window using mean reward $\bar{r}_{t,d}$, mean absolute advantage $\bar{a}_{t,d}$, or reward slope $s_{t,d}$. Here $n_{t,d}$ is the number of prompts sampled from domain d and $N_t = \sum_d n_{t,d}$ is the total.

This distinction gives us a direct comparison. The methods `per`, `softmax`, and `topk` all rank responses using a_{gk}. The first two convert that score into a continuous loss weight; `topk` uses it to keep or discard a response. Comparing them asks whether the same signal is more useful for reweighting or for selection.

Table 1 lists the complete set of released configurations. Several implement or closely adapt published proposals: `ar` follows Advantage Reweighting Yang et al. (2025), `difffilter` is a stricter DAPO-style filter Yu et al. (2025), `maxvar` follows PODS Xu et al. (2026), and `gfpo` uses GFPO's reward-per-token criterion Shrivastava et al. (2025). The mixture methods draw on DoReMi, DUMP, and Teacher–Student Curriculum Learning Xie et al. (2023a); Wang et al. (2025); Matiisen et al. (2017). The remaining configurations are controlled variants built from the same signals. For compactness, the table writes i for a response indexed by a prompt–response pair (g, k).

The canonical `dump_ucb` and `tscl` implementations use the distinct signals shown in the table. Twelve earlier proxy runs that supplied reward levels to both methods are retained for provenance but excluded from every canonical mean.

4 Experimental Setup

Within each comparison, we keep the model, corpus, verifier, optimizer, rollout budget, and evaluation protocol fixed. This section describes the common setup used throughout the experiments.

All runs use one shared driver for rollout, verification, GRPO optimization, and 12-benchmark evaluation; only the data-policy intervention varies. Each method is registered by its intervention point, signal, operator, and hyperparameters, while the evaluator writes every run to a common record format. Implementation and run-inventory details are provided in Appendix A.1.

4.1 Data, training, and evaluation

The training corpus contains $15{,}000$ prompts, split equally among math, logic, and science. Math combines `math_dapo`, DeepScaler, and GSM8K prompts with boxed-answer verification; logic

uses procedurally generated Knights&Knaves puzzles with an assignment checker; science uses SciQ multiple-choice questions with exact letter matching. The released records retain source and domain metadata for every prompt.

All campaigns use verl v0.5+ and GRPO with five rollouts per prompt, a KL coefficient of 10^{-3}, and prompt and response limits of 1024 and 8192 tokens. Runs use 300 optimizer steps and checkpoint every 100 steps unless the campaign explicitly studies 1000-step training. Comparisons use matched seeds; the breadth grid uses seeds $\{1, 2, 3\}$, while the Qwen-base primary matrix and higher-seed replications use seeds $\{1, \ldots, 12\}$.

We evaluate every run on 12 benchmarks: five math sets (MATH-500 Lightman et al. (2024), AIME-2024, OlympiadBench He et al. (2024), MinervaMath Lewkowycz et al. (2022), and GSM8K Cobbe et al. (2021)), four logic sets (Knights&Knaves Xie et al. (2025), two BBH tasks Suzgun et al. (2023), and ZebraLogic Lin et al. (2025)), and three science sets (MMLU-Pro Chemistry, MMLU-Pro Physics Wang et al. (2024a), and GPQA-Diamond Rein et al. (2023)). Decoding is deterministic. Math and GPQA allow up to 8192 output tokens; the shared logic and MMLU-Pro evaluator uses 4096.

Training and evaluation formats can disagree without an obvious training failure, so we run a calibration smoke test for boxed-answer and multiple-choice parsing before each campaign and audit the training prompts against all evaluation items. No exact, normalized, or 13-gram Jaccard match above 0.5 was found; detailed audit outputs are provided in Appendix A.1.

The **Overall** score averages within MATH, LOGIC, and SCIENCE and then gives the three domains equal weight. The main Qwen-base table exposes all 12 benchmark means; compact tables report Overall and paired intervals.

Primary comparisons pair methods by training seed and report seed-level 95% intervals; the three-seed extensions are used to describe coverage and direction rather than to support standalone significance claims.

Because each run retains its benchmark-level scores, we can recompute alternative evaluation summaries on the same trained checkpoints without introducing another training comparison.

The release spans 591 scientific runs, including the complete 13-configuration, 12-seed Qwen2.5-7B-base block and the corrected Llama and Qwen-base extensions; the full inventory and record schema are provided in Appendix A.1.

5 Main Result on a Base Model with Training Headroom

Our main experiment uses Qwen2.5-7B-base, whose untrained checkpoint leaves substantial room for post-training. All 13 configurations are evaluated with 12 matched seeds. The comparison has two parts: selection and reweighting methods are measured against uniform sampling, while adaptive mixtures are measured against a fixed, equal mixture of math, logic, and science data.

5.1 Training Headroom

Before comparing data policies, we measure how much the shared GRPO recipe changes each representative starting model. Table 2 reports the untrained checkpoint and the uniform-GRPO control under the same deterministic 12-benchmark evaluation. Both base-model rows use 12 matched seeds; the interval is shown for the Qwen2.5-7B-base comparison, which is the primary inferential setting.

Base Model	Seeds	Untrained	Uniform GRPO	Δ	95% CI
Qwen2.5-7B-base	12	42.01	49.77	**+7.76**	[7.28, 8.25]
Llama-3.1-8B-base	12	10.87	21.12	**+10.25**	[7.75, 12.33]

Table 2: Training headroom in two representative base-model settings. Uniform GRPO is compared with the corresponding untrained checkpoint. Untrained scores are deterministic evaluations of fixed checkpoints; the Qwen2.5-7B-base interval reflects variation across its 12 trained baseline seeds.

	MATH					LOGIC				SCIENCE			SUMMARY	
Method	M500	A24	Oly	Min	GSM8K	K&K	LD	Track	Zebra	GPQA	Chem	Phys	**Overall**	95% CI
baseline	84.45	14.44	38.06	34.93	91.42	20.46	65.77	78.75	29.78	33.71	53.17	57.05	49.77	[49.29, 50.26]
difffilter	84.23	15.01	37.78	35.06	91.10	22.38	65.76	76.97	32.28	32.58	53.75	56.42	49.85	[49.10, 50.61]
maxvar	84.18	14.16	37.77	35.81	90.91	18.38	65.13	78.13	29.47	31.65	53.13	56.93	49.19	[48.57, 49.82]
gfpo	83.78	13.88	37.88	34.94	91.31	17.62	64.80	76.16	30.53	31.73	53.23	56.03	48.88	[48.03, 49.73]
topk	84.30	14.17	37.82	34.69	91.42	20.12	66.31	76.69	30.13	32.79	53.02	56.33	49.39	[48.67, 50.11]
ar	84.63	14.45	37.30	35.24	91.18	19.79	66.23	77.21	31.55	33.29	52.63	55.78	49.50	[49.00, 49.99]
per	81.93	12.51	36.59	33.48	90.11	22.83	66.10	73.18	30.82	33.54	52.53	56.72	48.92	[46.35, 51.48]
softmax	83.77	13.88	37.28	35.36	90.92	21.12	66.04	76.62	30.93	33.04	53.95	56.13	49.54	[48.59, 50.50]
diffband	84.13	15.83	37.75	35.54	91.07	18.79	67.23	77.33	30.07	33.80	53.73	56.58	49.75	[49.06, 50.45]
static	84.60	15.00	37.51	35.02	91.27	19.00	66.04	75.11	29.05	33.16	52.48	55.43	49.00	[48.52, 49.48]
reward_gap	83.95	11.38	37.66	33.85	91.37	21.25	66.36	77.21	31.58	32.66	53.35	56.87	49.46	[49.00, 49.91]
dump_ucb	84.67	13.62	37.79	34.60	91.21	19.42	66.79	78.02	31.25	31.90	53.22	57.67	49.61	[49.16, 50.07]
tscl	84.13	14.72	37.68	34.19	91.27	16.54	66.08	78.29	30.10	32.24	53.47	56.27	49.16	[48.27, 50.04]

Table 3: Twelve-benchmark means for the complete Qwen2.5-7B-base primary experiment. Each row averages 12 matched seeds; Overall gives equal weight to the MATH, LOGIC, and SCIENCE domain means. The final column is the 95% t interval for the seed-level Overall mean; paired method comparisons are reported in Figure 2.

Uniform GRPO improves by $+7.76$ points on Qwen-base and $+10.25$ points on Llama-base. These gains make it unlikely that the null data-policy result is simply due to the model failing to learn. They also motivate the Qwen2.5-7B-base block as the primary test: it combines substantial headroom with complete method coverage and 12 matched seeds.

For a direct view of the primary experiment, Table 3 reports the mean score on each benchmark after averaging the 12 matched seeds. The differences are not concentrated in one domain: methods that are relatively strong on LOGIC are not uniformly strong on MATH or SCIENCE. Overall therefore summarizes a genuinely multi-domain comparison rather than a single benchmark group.

5.2 Selection and Reweighting Add No Reproducible Gain

The upper two blocks of Figure 2 compare uniform sampling with all four selection and all four reweighting configurations, pairing each difference by training seed. This is the paper's primary method comparison: it combines complete method coverage with enough seeds to estimate the size of the remaining differences.

None of the eight paired intervals excludes zero. Including the baseline, the nine policy means span 0.97 point, about one eighth of the 7.76-point gain from uniform GRPO. The selector ordering also changes when the comparison uses all 12 seeds: `topk` leads in the original three-seed subset, whereas `difffilter` has the highest 12-seed mean and `topk` falls below the baseline. The four reweighting estimates range from -0.86 to -0.02, and every interval crosses zero. **At the precision of this experiment, changing the policy adds no reproducible gain, even though GRPO itself is clearly effective.**

5.3 Adaptive Mixtures Show No Reproducible Gain

The bottom block of Figure 2 compares three adaptive mixtures with a `static` control. The checkpoint and training recipe are identical within this campaign; only the proportions of math, logic, and science prompts change during training. The `static` row is a campaign-local control, so its mean need not equal the primary uniform-baseline mean; all mixture differences are paired within this campaign.

All three adaptive mixtures have positive point estimates relative to the fixed mixture, but every paired interval includes zero and the means span only 0.61 point. Training logs confirm that the methods changed the realized domain proportions. **In this setting, changing the mixture changes which data the model sees, but it does not produce a reproducible improvement in accuracy.**

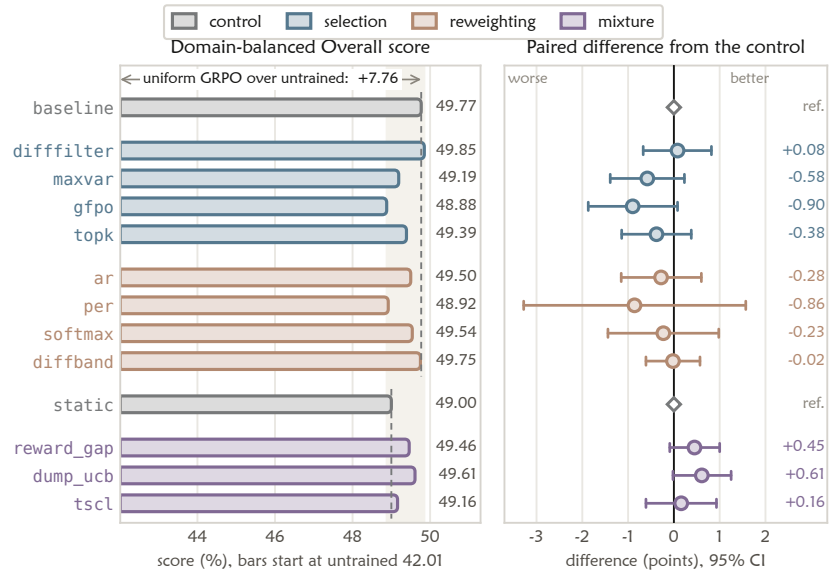

Figure 2: All 13 Qwen2.5-7B-base configurations with 12 matched seeds. **Left:** bars start at the untrained checkpoint (42.01), so their length is the combined GRPO and data-policy gain; the band spans all 13 means (0.97 points). **Right:** paired difference from each block's control—uniform sampling above, the `static` mixture below—with its 95% t interval.

6 Cross-Model and Evaluation Checks

Section 5 establishes the main result on Qwen2.5-7B-base with 12 seeds and complete method coverage. We then examine three possible sources of variation: model family, model scale, and evaluation coverage. The first two checks use the corrected base-model extension. The last check recomputes the same runs under alternative benchmark summaries.

6.1 Llama-3.1-8B-base Evaluation

The corrected 12-seed Llama-3.1-8B-base evaluation uses the same protocol as the Qwen-base comparison. The untrained checkpoint scores 10.87 Overall, while uniform GRPO reaches 21.12, a gain of 10.25 points (95% CI $[7.75, 12.33]$). Thus the common GRPO recipe is effective on Llama as well; the comparison is whether a data policy adds to that gain. The complete benchmark-level means are listed in Appendix A.4 (Table 6).

Relative to the baseline, the paired Overall differences for the three original selectors are -1.09 for `difffilter` (95% CI $[-2.84, 0.65]$), -0.71 for `maxvar` ($[-2.18, 0.76]$), and -0.26 for `topk` ($[-1.76, 1.24]$); all three intervals include zero. The corrected extension places the nine additional methods on the same scale as the controls, with means from 18.66 (`softmax`) to 21.98 (`diffband`) versus 21.12 for the reusable baseline. The appendix intervals are seed-level means, not paired method comparisons, and do not identify a universal winner. Earlier 14–16 point Llama values came from an evaluator that silently scored unsupported math tasks as zero and are excluded.

6.2 Base-Model Scale and Family Coverage

The corrected extension covers Qwen2.5 base sizes from 1.5B to 14B and Llama-3.2-3B. Auxiliary rows use three matched seeds; the Qwen 7B row is the 12-seed anchor from Section 5. Appendix A.5 reports a planned six-configuration sweep (baseline, two selectors, one reweighting method, and two mixture controls); these rows describe coverage and direction rather than standalone significance tests.

The signs vary across models rather than following a common pattern. For example, `difffilter` is below the uniform baseline at 1.5B and 3B but slightly above it at 7B and 14B, while `maxvar`

is below the baseline in every row. The mixture comparison also changes direction: `tscl` is below its fixed control at 1.5B but above it at the larger Qwen sizes and on Llama-3.2-3B. These three-seed rows therefore show why a single cross-model winner is not evident; the complete inferential comparison remains the Qwen2.5-7B-base anchor.

6.3 Method Rankings Depend on Evaluation-Domain Coverage

We ask whether the leading method remains the same when the evaluation covers all three training domains. It does not: a math-heavy six-benchmark summary (five math sets plus GPQA-Diamond) and the domain-balanced 12-benchmark summary select different leading methods, whereas alternative summaries retaining all 12 benchmarks largely agree.

The reason is that the methods exhibit domain trade-offs. On Qwen2.5-7B-Instruct, for example, `diffband` has the highest LOGIC mean (59.0) and the lowest SCIENCE mean (44.2). An evaluation that omits LOGIC therefore measures a materially different profile rather than merely averaging fewer benchmarks.

For the same nine Qwen2.5-7B-Instruct configurations, we compare four summaries. Math-Heavy-6 averages five math benchmarks and GPQA-Diamond and contains no logic benchmark. DB-12 first averages within MATH, LOGIC, and SCIENCE and gives the three domains equal weight; Macro-12 weights benchmarks equally, while Item-12 weights them by evaluation-set size.

Table 4 summarizes the ranking and spread changes across these evaluation summaries.

Evaluation summary	Rank ρ vs. DB-12	Spread (points)
DB-12 (domain-balanced)	1.00	3.31
Macro-12	0.88	2.79
Item-12	0.88	2.17
Math-Heavy-6	-0.33	0.90

Table 4: Sensitivity of the nine Qwen2.5-7B-Instruct configuration means to evaluation-domain coverage and aggregation. Rank correlation is Spearman's ρ relative to DB-12; spread is the highest minus lowest configuration mean in percentage points. Each configuration mean uses the same three trained seeds.

Math-Heavy-6 ranks `topk` and `diffband` first, whereas DB-12 ranks `gfpo` and `difffilter` first. The rankings are negatively correlated ($\rho = -0.33$), and the between-method spread changes from 0.90 to 3.31 points. By contrast, Macro-12 and Item-12 both correlate strongly with DB-12 ($\rho = 0.88$), and leave-one-out checks range from 0.78 to 0.98 (Appendix A.2). The disagreement is therefore driven mainly by the omitted logic domain, not by small changes in weighting once all 12 benchmarks are retained.

These checks leave the primary conclusion unchanged. The corrected Llama evaluation places the additional configurations on the same score scale as the original controls and finds no common winner in their observed means. The base-model scale and family sweep likewise shows no monotone advantage for any method. The aggregation analysis adds a separate caution: if a domain is omitted, the apparent winner can change even when the underlying runs are identical.

7 Conclusion

DataFlex-RL asks whether RLVR data policies add to a shared GRPO recipe. On Qwen2.5-7B-base, uniform GRPO improves Overall accuracy by 7.76 points; none of the eight selection or reweighting methods has a paired interval excluding zero, and none of the three adaptive mixtures improves over its fixed control (mean spreads 0.97 and 0.61 points). Llama-3.1-8B-base shows the same separation between a large GRPO gain (10.25 points; 95% CI $[7.75, 12.33]$) and no common data-policy winner. The cross-model checks do not reveal a consistent advantage, while omitting an evaluation domain can change the apparent winner. Under the controlled recipes studied here, the tested policies therefore add no reproducible improvement over uniform training at the achieved precision. The released records and reconstruction scripts make these comparisons auditable and reusable.

AI USE STATEMENT

In preparing this work, we used generative AI tools for language editing, translation, and assistance with code implementation. The authors reviewed the AI-assisted text for accuracy and consistency with the intended meaning, and checked the AI-assisted code against the intended implementation. The authors retain full responsibility for the methodology, experiments, results, and final manuscript.

ETHICS STATEMENT

This work studies data-policy choices for reinforcement learning with verifiable rewards using existing benchmarks, publicly available models, and procedurally generated tasks. It does not involve human subjects or the collection of personal data. We report negative and inconclusive results alongside positive findings to reduce selective reporting and to support transparent comparison of RLVR data policies. All datasets and models should be used in accordance with their respective licenses and terms of use.

REPRODUCIBILITY STATEMENT

Section 3 describes the data-policy interventions, and Section 4 documents the shared training and evaluation setup, including data sources, verification procedures, seed settings, and score aggregation. Appendix A.1 summarizes the experimental run inventory and associated artifacts. Appendices A.2–A.5 provide evaluation-sensitivity analyses, selector diagnostics, and additional benchmark results. These descriptions support reproduction of the controlled comparisons and examination of the scope of our conclusions.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html.

Maosong Cao, Kai Chen, Haodong Duan, Yixiao Fang, Tong Gao, Ge Jiaye, Mo Li, Hongwei Liu, Junnan Liu, Yuan Liu, et al. OpenCompass: A universal evaluation platform for large language models. *arXiv preprint arXiv:2605.19276*, 2026. URL https://arxiv.org/abs/2605.19276.

Mayee F. Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! a data-driven skills framework for understanding and training language models. *arXiv preprint arXiv:2307.14430*, 2023. URL https://arxiv.org/abs/2307.14430.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Mostafa Dehghani, Yi Tay, Alexey A. Gritsenko, Zhe Zhao, Neil Houlsby, Fernando Diaz, Donald Metzler, and Oriol Vinyals. The benchmark lottery. *arXiv preprint arXiv:2107.07002*, 2021. URL https://arxiv.org/abs/2107.07002.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. A framework for few-shot language model evaluation, 2021. URL https://doi.org/10.5281/zenodo.5371629.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3207–3214, 2018. doi: 10.1609/aaai.v32i1.11694.

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. *arXiv preprint arXiv:2103.03874*, 2021.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

Hao Liang, Zhengyang Zhao, Meiyi Qiang, Mingrui Chen, Lu Ma, Rongyi Yu, Hengyi Feng, Shixuan Sun, Zimo Meng, Xiaochen Ma, et al. DataFlex: A unified framework for data-centric dynamic training of large language models. *arXiv preprint arXiv:2603.26164*, 2026. URL https://arxiv.org/abs/2603.26164.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2211.09110.

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*, volume 2024, pp. 39578–39601, 2024.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. ZebraLogic: On the scaling limits of LLMs for logical reasoning. *arXiv preprint arXiv:2502.01100*, 2025. URL https://arxiv.org/abs/2502.01100.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. RegMix: Data mixture as regression for language model pre-training. *arXiv preprint arXiv:2407.01492*, 2024. URL https://arxiv.org/abs/2407.01492.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. *arXiv preprint arXiv:1707.00183*, 2017. URL https://arxiv.org/abs/1707.00183.

Andrew Patterson, Samuel Neumann, Martha White, and Adam White. Empirical design in reinforcement learning. *Journal of Machine Learning Research*, 25(318):1–63, 2024. URL https://www.jmlr.org/papers/v25/23-0183.html.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research: A report from the NeurIPS 2019 reproducibility program. *Journal of Machine Learning Research*, 22(164): 1–20, 2021. URL https://www.jmlr.org/papers/v22/20-303.html.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023. URL https://arxiv.org/abs/2311.12022.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.05952.

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

Vaishnavi Shrivastava, Ahmed Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. Sample more to think less: Group filtered policy optimization for concise reasoning. *arXiv preprint arXiv:2508.09726*, 2025. URL https://arxiv.org/abs/2508.09726.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35: 19523–19536, 2022.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024a.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024b. URL https://arxiv.org/abs/2406.01574.

Zhenting Wang, Guofeng Cui, Yu-Jhe Li, Kun Wan, and Wentian Zhao. DUMP: Automated distribution-level curriculum learning for RL-based LLM post-training. *arXiv preprint arXiv:2504.09710*, 2025. URL https://arxiv.org/abs/2504.09710.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024. URL https://arxiv.org/abs/2402.04333.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. On memorization of large language models in logical reasoning. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 2742–2785, 2025.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. *arXiv preprint arXiv:2305.10429*, 2023a. URL https://arxiv.org/abs/2305.10429.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. URL https://arxiv.org/abs/2302.03169.

Yixuan Even Xu, Yash Savani, Fei Fang, and J. Zico Kolter. Not all rollouts are useful: Down-sampling rollouts in LLM reinforcement learning. *Transactions on Machine Learning Research*, 2026. URL https://arxiv.org/abs/2504.13818.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024. URL https://arxiv.org/abs/2409.12122.

Zhihe Yang, Xufang Luo, Zilong Wang, Dongqi Han, Zhiyuan He, Dongsheng Li, and Yunjian Xu. Do not let low-probability tokens over-dominate in RL for LLMs. *arXiv preprint arXiv:2505.12929*, 2025. URL https://arxiv.org/abs/2505.12929.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

A Additional Analyses

A.1 Protocol and run inventory

Each method is registered at one of the three intervention points in Figure 1 by specifying its signal, operator, and hyperparameters. A campaign generator expands model–method–seed combinations into runs; the evaluator writes all 12 benchmark scores to a common record format, which feeds the released table-reconstruction and statistical-analysis scripts. The release contains 591 scientific runs, including the complete 13-configuration, 12-seed Qwen2.5-7B-base block and the corrected Llama and Qwen-base extensions; the original three-seed breadth and diagnostic runs are retained for coverage and mechanism checks. The detailed campaign inventory, record schema, calibration outputs, and contamination audit are released with the artifact.

A.2 Sensitivity to the Evaluation Summary

DB-12 is the mean of the MATH, LOGIC, and SCIENCE domain scores. We compare it with three alternatives on the nine-configuration Qwen2.5-7B-Instruct campaign. Math-Heavy-6 averages five math benchmarks with GPQA-Diamond and omits logic; Macro-12 gives every benchmark equal weight; Item-12 weights benchmarks by evaluation-set size. Math-Heavy-6 and DB-12 have negatively correlated method rankings (Spearman $\rho = -0.33$, Kendall $\tau = -0.28$), and the range of method means changes from 0.90 to 3.31 points. Macro-12 and Item-12 both correlate with DB-12 at $\rho = 0.88$. Leave-one-benchmark-out correlations range from 0.78 to 0.98. The ranking change is therefore driven primarily by the omitted logic domain rather than by the weighting used once all 12 benchmarks are retained.

A.3 Selector Diagnostics

Table 5 compares each original selector with controls for retention count, update volume, and intervention strength. Matched random keeps the same number of groups or responses but chooses them randomly. Matched update tokens continues training until the run has used at least as many nonzero update tokens as the 300-step baseline. Alternative strengths widen the filtering band or change the keep fraction.

Selector	Targeted	Matched Random	Matched Update Tokens	Alternative Strength(s)
difffilter	53.81 ± 1.38	54.29 ± 1.85	54.33 ± 4.09	54.48 ± 0.90
maxvar	53.42 ± 0.56	50.93 ± 3.17	54.64 ± 1.67	54.30 ± 1.33 / 52.34 ± 0.79
topk	52.86 ± 0.66	52.29 ± 1.54	51.58 ± 0.11	53.42 ± 1.63 / 53.44 ± 1.25

Table 5: Three-seed selector diagnostics on Qwen2.5-7B-Instruct (mean $\pm$ sample standard deviation). Alternative strengths use the full nonzero solve-rate band for `difffilter` and keep fractions $0.25/0.75$ for `maxvar` and `topk`.

The controls move the three selectors in different directions. Targeted selection exceeds matched random for `maxvar` and `topk`, but not for `difffilter`; matching update tokens raises `difffilter` and `maxvar`, but lowers `topk`. No single targeting, update-volume, or strength effect explains all three methods. In a separate response-efficiency analysis, `gfpo` and the baseline have nearly identical observed accuracy on five math benchmarks (72.21% and 72.19%), while `gfpo` reduces mean response length from 1568 to 1497 characters and raises correct answers per thousand characters from 0.461 to 0.482.

A.4 Full Llama-base benchmark results

	MATH					LOGIC				SCIENCE			SUMMARY	
Method	M500	A24	Oly	Min	GSM8K	K&K	LD	Track	Zebra	GPQA	Chem	Phys	**Overall**	95% CI
baseline	32.18	0.00	4.71	7.38	47.09	3.71	39.87	23.95	19.13	27.49	19.35	23.45	21.12	[20.01, 22.23]
difffilter	29.12	0.27	5.05	8.49	39.36	4.42	40.60	24.07	18.07	25.85	18.23	21.47	20.03	[18.46, 21.60]
maxvar	32.98	0.55	4.85	8.12	44.87	3.92	38.31	23.57	16.52	25.42	19.65	22.07	20.41	[19.52, 21.30]
topk	31.22	0.00	4.66	7.54	44.55	4.29	41.13	26.23	17.22	25.76	19.75	22.82	20.86	[19.78, 21.95]
gfpo	31.23	0.27	4.92	7.94	46.77	4.21	38.99	27.89	18.55	27.53	19.77	22.93	21.35	[19.81, 22.89]
ar	31.10	0.27	4.70	8.57	42.68	4.46	39.43	27.27	18.38	24.92	18.32	20.83	20.40	[18.99, 21.82]
per	32.72	0.00	4.52	7.32	48.30	4.62	39.11	27.94	16.85	27.31	20.68	23.88	21.55	[20.59, 22.52]
softmax	29.32	0.00	4.47	7.13	38.36	3.29	34.26	22.09	12.55	25.72	19.27	21.27	18.66	[16.91, 20.42]
diffband	33.55	0.27	4.81	8.87	48.54	3.54	42.58	26.39	21.60	26.30	20.77	22.52	21.98	[20.76, 23.19]
static	32.67	0.00	4.78	8.18	47.80	3.83	41.94	26.12	19.12	26.77	20.82	23.68	21.73	[20.71, 22.76]
reward_gap	31.62	0.00	4.92	8.18	44.56	4.71	41.09	26.18	20.00	25.76	20.13	23.65	21.34	[19.99, 22.69]
dump_ucb	30.07	0.55	4.36	7.51	42.79	3.92	39.06	22.11	15.37	26.43	18.53	22.50	19.89	[18.00, 21.77]
tscl	33.78	0.82	5.53	7.71	48.92	4.00	38.64	22.09	17.08	26.05	18.65	21.80	20.66	[19.66, 21.65]

Table 6: Twelve-benchmark means for the Llama-3.1-8B-base evaluation. Each row averages 12 matched seeds; Overall gives equal weight to the MATH, LOGIC, and SCIENCE domain means. The final column is the 95% t interval for the seed-level Overall mean.

A.5 Base-model scale and family table

Base Model	Seeds	Uniform	Selection				Reweighting		Mixture		
		baseline	difffilter	ΔU	maxvar	ΔU	softmax	ΔU	static	tscl	ΔS
Qwen2.5-1.5B-base	3	28.14	27.31	-0.83	26.96	-1.18	27.83	-0.31	27.47	27.37	-0.10
Qwen2.5-3B-base	3	37.31	36.52	-0.79	36.76	-0.55	36.78	-0.53	36.83	37.06	+0.23
Qwen2.5-7B-base	12	49.77	49.85	+0.08	49.19	-0.58	49.53	-0.24	49.00	49.16	+0.16
Qwen2.5-14B-base	3	59.38	60.58	+1.20	58.54	-0.84	59.14	-0.24	58.22	59.65	+1.43
Llama-3.2-3B-base	3	13.16	12.80	-0.36	11.47	-1.69	13.45	+0.29	12.74	12.96	+0.22

Table 7: Overall means in the corrected base-model scale and family sweep. Each cell for a selector or reweighting method reports its mean followed by the difference from the uniform baseline; the `static` column is the fixed-mixture control, and the `tscl` difference is relative to that control. Auxiliary rows use three seeds, while Qwen2.5-7B-base uses the 12-seed anchor. Red indicates an increase and blue indicates a decrease.