RACS: RESPONSE ACTIVATION CONSISTENCY-GUIDED DATA SELECTION FOR REINFORCEMENT LEARNING WITH VERIFIABLE REWARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) is widely used to improve the reasoning abilities of language models. Training requires many sampled responses, so selecting useful prompts is important for making effective use of the training budget. A common approach to RLVR data selection relies on prompt difficulty estimated from pass rates. Pass rate summarizes response correctness but does not capture the current policy's behavior during generation. Unlike prior work, we study *response activation consistency*, the directional agreement among hidden-state vectors of multiple responses sampled from the current policy for the same prompt. In our pilot study, training on prompts with higher response activation consistency yields better test accuracy. Inspired by this, we propose Response Activation Consistency-guided Data Selection, which combines response activation consistency scoring, reference-based score estimation, and consistency-guided prompt sampling. Each epoch, it measures consistency on a small reference set containing fewer than 2% of the corpus, estimates scores for the rest, and favors high-scoring prompts for training. On five mathematical reasoning benchmarks, our method improves pooled accuracy on Qwen2.5-Math-1.5B by 3.24 percentage points over random selection and 2.92 points over the strongest evaluated baseline under the same training-update budget. Further analysis shows that high-consistency prompts tend to have more separable correct and incorrect response representations, suggesting a possible explanation for the observed gains. Our work identifies response activation consistency across sampled responses as a useful signal for RLVR data selection and provides a practical procedure for using it during training.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) improves the reasoning abilities of language models by rewarding answers that can be checked automatically (Shao et al., 2024; Yang et al., 2024; Luo et al., 2025). Training repeatedly samples responses from the current policy, which incurs substantial computational cost (Sun et al., 2025; Tang et al., 2026). Under a limited training budget, selecting which prompts receive these rollouts and updates is therefore an important part of the training procedure. A common approach selects prompts according to their difficulty under the current policy (Sun et al., 2025; Bae et al., 2026b). Difficulty is typically estimated by the pass rate, the fraction of sampled responses that pass the verifier. These methods favor prompts of intermediate difficulty, where reward variation across sampled responses produces stronger and more informative policy gradients for Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Bae et al., 2026b).

Pass-rate-based methods rely primarily on response correctness to guide prompt selection. They do not capture the relationships among the policy's internal representations of its different responses. We study whether these relationships can help identify useful training prompts. For a given prompt, we define *response activation consistency score (RACS)* as the directional agreement among the hidden-state vectors of multiple responses sampled from the current policy (Section 3.2). A prompt has high consistency when the internal representations of its sampled responses are closely aligned. The comparison is among responses to the same prompt, and the score can change as the policy

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

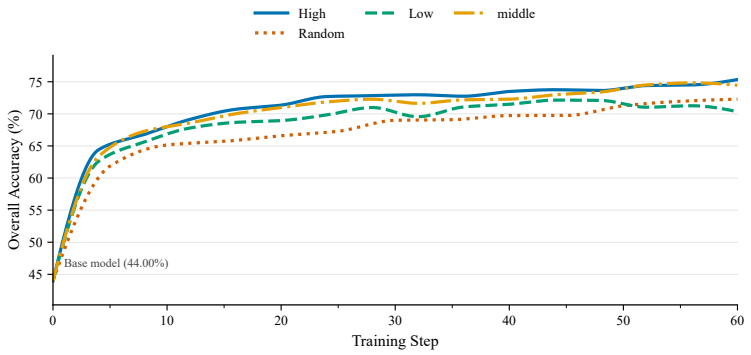


Figure 1: **Training with different levels of response activation consistency.** Qwen2.5-Math-1.5B policies are trained with GRPO on prompts selected from high-, medium-, or low-consistency groups. Scores are refreshed and prompts are resampled each epoch. The curves show aggregate accuracy on 2,161 mathematical reasoning problems.

changes. Our corpus analysis shows that prompts of similar difficulty can elicit substantially different levels of response activation consistency (Section 6.2). A selection rule based only on pass rate does not distinguish such differences.

To test whether this signal helps identify useful training prompts, we conduct a pilot study with Qwen2.5-Math-1.5B. We train three policies with GRPO, selecting prompts that elicit high, medium, or low response activation consistency scores from each policy at its current training stage. Under the same update budget, high-consistency selection achieves higher aggregate test accuracy than both alternatives (Figure 1). This comparison supports the value of considering response activation consistency when selecting training prompts.

Based on these findings, we introduce RACS-guided data selection, an online method for RLVR prompt selection (Section 3). Its *response activation consistency scoring* module measures agreement among the current policy's response internal representations for each prompt. However, scoring every candidate directly would require generating multiple responses for each prompt in the corpus, adding substantial computational cost to data selection. To reduce this cost, we use *reference-based score estimation* to predict scores for the corpus from a small reference set. The estimator uses similarities between frozen prompt representations to transfer these measurements to other prompts. *Consistency-guided prompt sampling* then favors prompts with high estimated consistency for GRPO updates. Reference scores and sampling weights are refreshed each epoch to follow changes in the policy.

Our experiments show that RACS improves reasoning accuracy under a fixed training-update budget. We evaluate our method with GRPO on five mathematical reasoning benchmarks. On Qwen2.5-Math-1.5B, RACS improves aggregate accuracy by 3.24 percentage points over random selection and by 2.92 points over the strongest baseline. Further analysis shows that high-consistency prompts tend to have more separable correct and incorrect response representations. We quantify separability using the Fisher discriminant ratio, which compares the distance between group centroids with the spread within each group. On the 1.5B model, high-consistency prompts have higher ratios than low-consistency prompts at six of eight measured training stages (Section 6.3). The difference is approximately 10% in early training. This association offers a possible explanation for the gains from high-consistency selection.

Our contributions are as follows:

- We find that response activation consistency is informative about prompt training value in RLVR. Prioritizing high-consistency prompts yields better downstream performance.
- We propose RACS-guided data selection, an online RLVR data-selection method that integrates response activation consistency scoring, reference-based score estimation, and consistency-guided prompt sampling.
- We demonstrate accuracy gains under a fixed GRPO update budget on five math benchmarks and across four models. We also analyze score estimation, prompt difficulty, and representation separability to examine when and why consistency can be useful.

2 Related Work

Data selection for RLVR. RLVR data selection often favors prompts with intermediate pass rates (Sun et al., 2025; Bae et al., 2026b). These prompts are less likely to yield identical rewards and zero GRPO advantages (Shao et al., 2024). DOTS (Sun et al., 2025) estimates difficulty online using rollouts from a small reference set and similarity-based prediction for other prompts. DEPO (Tang et al., 2026) combines offline selection based on diversity, influence, and difficulty with online filtering based on exploration potential. Other methods consider learning dynamics or uncertainty. LIMR (Li et al., 2025) favors prompts whose reward trajectories align with the model's learning progress. InSight (Zhou et al., 2026) combines difficulty and uncertainty in estimated success rates using weighted mutual information. Unlike difficulty-based methods, SHIFT (Wu et al., 2026) uses the difference between the initial and final hidden states of a single reasoning trace for offline prompt selection. In contrast to these approaches, our online method focuses on the directional consistency of response activations across multiple responses to the same prompt.

Gradient-based data selection. Gradient-based methods assess data utility through the parameter updates that training examples would induce. LearnAlign (Li et al., 2026) combines gradient similarity within the training corpus with a success-rate-based measure of learnability. GradAlign (Yang et al., 2026) instead compares candidate policy gradients with gradients from a small, trusted validation set. Computing candidate gradients requires additional backward passes, which can be costly when selecting from large training corpora. Refreshing these gradients as the policy changes adds further overhead to online selection. For validation-based methods, the choice of reference data also determines which update directions are favored.

Consistency-based decoding and uncertainty estimation. Agreement among sampled responses has been used both to improve decoding and to estimate uncertainty. Self-consistency (Wang et al., 2023) selects an answer by aggregating predictions across multiple sampled reasoning paths. Xiao et al. (2025) formalize and empirically examine the relationship between generation similarity and correctness. CoCoA (Vashurin et al., 2025) combines semantic agreement across outputs with confidence derived from token probabilities. In these approaches, consistency is assessed through final answers or generated text to support inference-time decisions. The connection between response agreement and prediction reliability motivates examining whether consistency can also help identify useful training prompts.

3 Method

Unlike difficulty-based criteria derived from pass rates (Sun et al., 2025), we compare the model's internal representations across sampled responses. Our method selects prompts according to the activation consistency of the current policy's responses to each prompt. The **response activation consistency scoring** module defines the Response Activation Consistency Score (RACS) from these responses (Section 3.2). **Reference-based score estimation** predicts RACS across the corpus using measurements on a small reference set (Section 3.3). **Consistency-guided prompt sampling** uses the estimates to choose candidates and retains mixed-reward response groups for GRPO (Section 3.4). Figure 2 shows how the modules interact during training.

3.1 Problem setup

Let $\mathcal{D} = \{x_i\}_{i=1}^{N}$ be a corpus of N training prompts and π_θ the current policy. At each epoch, we select a subset $\mathcal{S} \subseteq \mathcal{D}$ of B prompts for training. We use Group Relative Policy Optimization (GRPO; Shao et al., 2024) with binary verifiable rewards. For a prompt x, the policy generates K responses $r_k \sim \pi_\theta(\cdot \mid x)$, each receiving a reward $R_k = R(x, r_k) \in \{0, 1\}$.

3.2 Response Activation Consistency Scoring

For a prompt x, the scoring module measures directional agreement among the representations of its K responses sampled from the current policy π_θ. We compute a score at each selected model layer and average the resulting values.

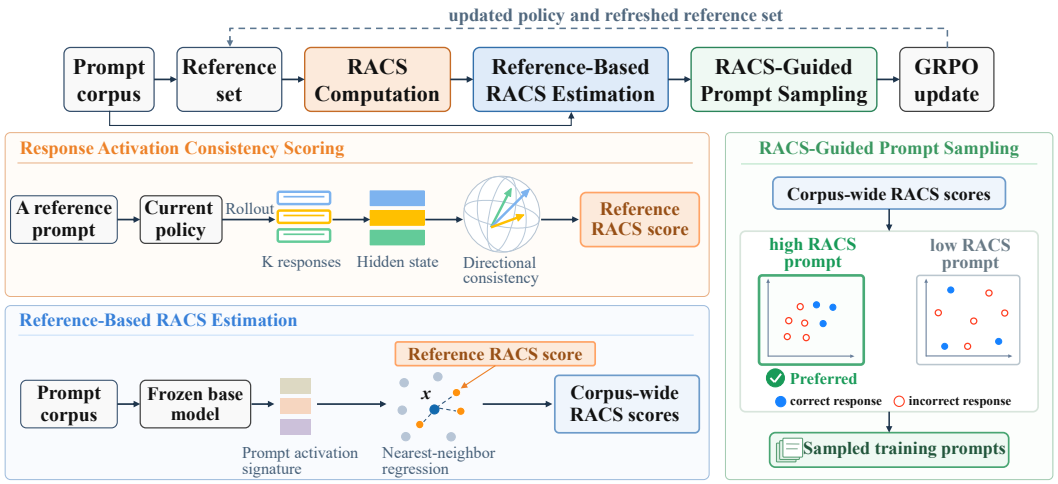


Figure 2: **Overview of RACS-guided data selection.** At each epoch, the current policy samples responses for a small reference set, and response activation consistency scoring computes a score for each prompt. Reference-based RACS estimation then uses these scores and frozen prompt activation signatures to predict scores for the remaining corpus. The resulting estimates determine the sampling weights in RACS-guided prompt sampling, which favors high-consistency prompts for GRPO updates. The updated policy is used to score a refreshed reference set in the next epoch, allowing selection to adapt as training progresses.

For each sampled response r_k, we run the current policy on the concatenated prompt–response sequence (x, r_k). Let $\mathcal{L} = \{\ell_1, \ldots, \ell_L\}$ index L transformer blocks evenly spaced across the model depth. At each selected block ℓ, we extract its output at the last valid response token, denoted by $\boldsymbol{h}_k^{(\ell)} \in \mathbb{R}^d$, where d is the hidden-state dimension. This position can attend to the prompt and preceding response tokens, so we use its hidden state as a response-level representation.

To quantify consistency across responses, we focus on the directions of their activation vectors. Studies of linear representations show that high-level concepts can be encoded as directions in representation space (Park et al., 2023). We therefore normalize each vector as $\mathbf{u}_k^{(\ell)} = \boldsymbol{h}_k^{(\ell)} / \|\boldsymbol{h}_k^{(\ell)}\|_2$, so that the consistency measure reflects directional alignment rather than vector magnitudes.

We define the layer-level score as the mean resultant length of the normalized response representations:

$$\mathrm{RACS}^{(\ell)}(x) = \left\| \frac{1}{K} \sum_{k=1}^{K} \mathbf{u}_k^{(\ell)} \right\|_2 \in [0, 1]. \tag{1}$$

The score is 1 when all response representations point in the same direction. Lower values indicate less alignment, with a score near 0 when the directions largely cancel.

The mean resultant length has a standard interpretation under the von Mises–Fisher (vMF) distribution, which models directional data on the unit sphere (Banerjee et al., 2005):

$$f(\mathbf{u}; \boldsymbol{\mu}, \lambda) = C_d(\lambda) \exp\left(\lambda \boldsymbol{\mu}^\top \mathbf{u}\right), \qquad \|\boldsymbol{\mu}\|_2 = 1, \quad \lambda \geq 0, \tag{2}$$

where $\boldsymbol{\mu}$ is the mean direction, λ is the concentration parameter, and $C_d(\lambda)$ is the normalizing constant. For K independent directions under this model, maximizing the log-likelihood over $\boldsymbol{\mu}$ gives

$$\max_{\|\boldsymbol{\mu}\|_2 = 1} \sum_{k=1}^{K} \log f(\mathbf{u}_k^{(\ell)}; \boldsymbol{\mu}, \lambda) = K \log C_d(\lambda) + K\lambda\, \mathrm{RACS}^{(\ell)}(x). \tag{3}$$

Thus, the mean resultant length is the sample statistic used to estimate vMF concentration.

The final score is the average over the selected layers:

$$\mathrm{RACS}(x) = \frac{1}{L} \sum_{\ell \in \mathcal{L}} \mathrm{RACS}^{(\ell)}(x). \tag{4}$$

The score is computed solely from response activations and does not use verifier rewards.

3.3 REFERENCE-BASED SCORE ESTIMATION

Directly scoring the corpus requires generating K responses for each of the N prompts, or $K \times N$ responses in total. Reference-set prediction has also been used to reduce rollout costs in difficulty-based data selection (Sun et al., 2025). For RACS, we compute response-based scores for a small reference set and estimate scores for the remaining prompts using similarity-weighted nearest-neighbor regression over frozen prompt activation signatures. This estimator requires no separately trained prediction model.

We use a small reference set to obtain response-based scores for estimating RACS over the corpus. At the start of epoch t, we sample M prompts uniformly from $\mathcal{D}$ to form $\mathcal{R}_t$. For each $x \in \mathcal{R}_t$, we generate K responses with the current policy and compute $\mathrm{RACS}(x)$ directly from their activations using equation 4. The reference set and its scores are refreshed at each epoch to track changes in the policy.

We use prompt activation signatures as features for predicting the scores of non-reference prompts. Probing studies show that prompt representations contain information about subsequent response properties, including response length, reasoning steps, and answer confidence (Dong et al., 2025).

We encode each prompt with a frozen copy of the base model. For each block index $\ell \in \mathcal{L}$ used in the response scoring step (Section 3.2), let $\mathbf{g}^{(\ell)}(x) \in \mathbb{R}^d$ denote the frozen model's output at the last valid prompt token. The prompt's *activation signature* concatenates these layer-wise vectors:

$$\varphi(x) = \left[\mathbf{g}^{(\ell_1)}(x)^\top, \ldots, \mathbf{g}^{(\ell_L)}(x)^\top\right]^\top \in \mathbb{R}^{Ld}. \tag{5}$$

The frozen encoder keeps the feature space fixed during training, while the reference scores are recomputed using the current policy. We cache the prompt features within each epoch.

Nearest-neighbor prediction. We estimate each candidate's score using nearby reference prompts. For $x \notin \mathcal{R}_t$, let $\mathcal{N}(x) \subseteq \mathcal{R}_t$ contain the n reference prompts most similar to x in activation-signature space. Using cosine similarity, we compute

$$\begin{aligned} \widehat{\mathrm{RACS}}(x) &= \sum_{x_j \in \mathcal{N}(x)} \alpha_j(x)\, \mathrm{RACS}(x_j), \\ \alpha_j(x) &= \frac{\exp(\cos(\varphi(x), \varphi(x_j)))}{\sum_{x_q \in \mathcal{N}(x)} \exp(\cos(\varphi(x), \varphi(x_q)))}, \end{aligned} \tag{6}$$

where $\cos(\cdot, \cdot)$ denotes cosine similarity. Reference prompts retain their directly computed scores, with $\widehat{\mathrm{RACS}}(x) = \mathrm{RACS}(x)$ for $x \in \mathcal{R}_t$. The estimator assumes that prompts close in activation signature space tend to have similar response activation consistency. Score estimation therefore requires $K \times M$ generated responses rather than $K \times N$, where $M \ll N$.

3.4 CONSISTENCY-GUIDED PROMPT SAMPLING

Given the corpus-wide scores, we sample training candidates with a preference for high consistency:

$$p(x) \propto \exp\left(\frac{\widehat{\mathrm{RACS}}(x) - \max_{x' \in \mathcal{D}} \widehat{\mathrm{RACS}}(x')}{\tau}\right), \qquad x \in \mathcal{D}, \tag{7}$$

where $\tau > 0$ is the selection temperature. Smaller values place more probability on high-scoring prompts.

At the start of each epoch, we refresh the RACS estimates under the current policy and update $p(x)$. We then sample from $p(x)$ until we collect B prompts whose sampled responses include both correct and incorrect answers, and use them for the subsequent GRPO updates. Pseudocode is provided in Algorithm 1 in Appendix A.

Table 1: Accuracy (%) on five math benchmarks after GRPO training with different data-selection methods. "All" denotes accuracy over all 2,161 test items. The best score in each model block is **bold** (ties included).

Model	Method	GSM8K	MATH500	Minerva	AIME24	AMC23	All
Qwen2.5-Math 1.5B	Random	83.62	72.80	24.26	**16.67**	50.00	72.10
	PPL-Top	84.31	71.60	25.74	10.00	55.00	72.42
	PPL-Middle	84.61	71.20	23.53	**16.67**	47.50	72.19
	DOTS+RR	78.92	69.20	18.75	6.67	50.00	67.56
	RACS	**86.81**	**76.40**	**26.84**	13.33	**60.00**	**75.34**
Qwen2.5-Math 7B	Random	91.96	77.80	30.15	13.33	50.00	79.04
	PPL-Top	89.92	79.20	27.21	20.00	62.50	78.07
	PPL-Middle	90.90	77.60	**30.51**	10.00	60.00	78.53
	DOTS+RR	85.90	77.20	23.16	20.00	55.00	74.50
	RACS	**92.42**	**83.60**	29.41	**26.67**	**65.00**	**81.03**
Qwen3-1.7B	Random	85.14	76.00	27.21	10.00	50.00	74.04
	PPL-Top	83.24	72.00	27.94	10.00	**52.50**	72.10
	PPL-Middle	82.79	70.20	22.79	6.67	47.50	70.62
	DOTS+RR	88.70	69.60	**30.88**	6.67	50.00	75.15
	RACS	**89.01**	**76.60**	29.04	**26.67**	**52.50**	**77.05**
Qwen3-4B	Random	94.39	88.20	38.24	33.33	72.50	84.64
	PPL-Top	94.01	87.00	38.60	23.33	82.50	84.22
	PPL-Middle	**95.00**	86.80	40.44	33.33	80.00	85.10
	DOTS+RR	93.86	86.80	**41.18**	36.67	75.00	84.45
	RACS	94.01	**91.60**	40.81	**50.00**	**90.00**	**86.07**

4 Experimental Setup

Training data and models. We construct the training pool by merging the DeepScaleR preview dataset (Luo et al., 2025), which contains approximately 40K prompts, with the MATH training split (Hendrycks et al., 2021). After deduplication, the pool contains approximately 44K prompts. We use Qwen2.5-Math-1.5B (Yang et al., 2024) for the main experiments and analyses, and include Qwen2.5-Math-7B, Qwen3-1.7B and Qwen3-4B in the model comparison.

Baselines. Random samples prompts uniformly from the training pool. PPL-Top and PPL-Middle are adapted from perplexity-based data pruning (Ankner et al., 2024). We also compare with DOTS+RR (Sun et al., 2025), which combines difficulty-targeted online data selection with rollout replay. We use the official default settings and pretrained math difficulty estimator. Training protocol and RACS configuration are provided in Appendix A.

Evaluation. We report accuracy on five mathematical reasoning benchmarks: GSM8K (Cobbe et al., 2021), MATH500 (Lightman et al., 2024), Minerva Math (Lewkowycz et al., 2022), AIME24, and AMC23. Evaluation uses sampling temperature 0.6 and top-p 0.95. Alongside the individual benchmark scores, we report "All," the accuracy over the pooled set of 2,161 test problems.

5 Main Results

We evaluate whether RACS-guided data selection improves reinforcement learning across four base models and five mathematical reasoning benchmarks. Table 1 reports accuracy after training with GRPO under each selection method.

The results show that response activation consistency is an effective signal for selecting RLVR training prompts. On Qwen2.5-Math-1.5B, our method reaches 75.34% on the pooled test set, improving over Random by 3.24 percentage points and over the strongest baseline by 2.92 points. It achieves the best result on four of the five benchmarks, supporting response activation consistency as a useful criterion for prompt selection.

The advantage also generalizes across model scale and family. On Qwen2.5-Math-7B, our method obtains the highest aggregate accuracy at 81.03%, while on Qwen3-1.7B and Qwen3-4B it raises ag-

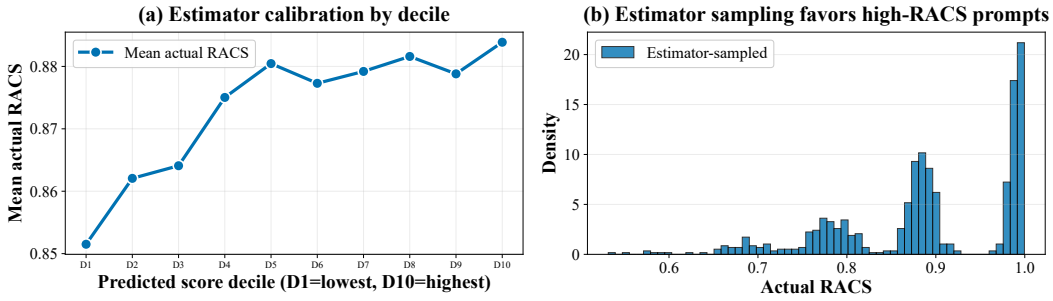


Figure 3: **(a)** Mean actual RACS within each predicted-score decile. The overall increase indicates coarse-grained rank preservation. **(b)** Actual scores of prompts selected using estimated-score weights. The distribution concentrates on high-consistency prompts.

gregate accuracy to 77.05% and achieves the best or tied-best result on four of the five benchmarks. These results suggest that the benefit is not tied to a particular checkpoint or architecture.

6 Analysis and Discussion

Our analysis addresses three questions: Can the score estimator guide selection while reducing scoring cost (§ 6.1)? How does activation consistency relate to prompt difficulty (§ 6.2)? How do the distributions of response representations differ between high- and low-consistency prompts (§ 6.3)?

6.1 Quality and cost of score estimation

We evaluate whether the score estimator in Section 3.3 retains enough information to guide prompt selection. We compare its predictions with directly computed, rollout-based RACS on candidates. Direct scoring of these candidates is used only for validation; the selection pipeline estimates their scores from a 512-prompt reference set. We assess relative ordering and the resulting sampling distribution.

For prompt selection, preserving the relative ordering of scores is important. Figure 3(a) groups candidates into ten deciles by predicted score, from D1 (lowest) to D10 (highest), and reports the mean actual RACS in each group. The means increase overall, with a strictly increasing trend from D1 through D5. This pattern indicates that the estimator distinguishes low-consistency candidates from higher-consistency ones, allowing the selector to assign them different sampling weights.

To examine the actual score distribution of selected prompts, we draw 100 batches of 512 prompts using temperature-softmax weights derived from the estimated scores (equation 7). We then pool the selected prompts. Their actual scores concentrate in the upper range, with substantial mass around 0.88–0.92 and near 1.0 (Figure 3(b)). Thus, the estimated scores can guide selection toward high-consistency prompts without corpus-wide rollouts for scoring.

In terms of computational efficiency, directly computing RACS requires rollout generation and activation extraction for multiple responses per prompt, whereas the estimator predicts RACS scores without additional rollouts. On a single A800 GPU with Qwen2.5-Math-1.5B, scoring all 44K prompts would therefore require approximately 46 GPU-hours per epoch, compared with about 32 minutes for the 512 reference prompts. Relative to scoring the full corpus, scoring only the reference set reduces response-based scoring time by approximately 98.8%.

6.2 Prompt difficulty and RACS distributions

Moderately difficult prompts are often favored in curriculum learning and adaptive data selection (Sun et al., 2025; Bae et al., 2026a). We examine whether RACS-guided data selection maintains such a difficulty profile and whether activation consistency provides a selection signal distinct from prompt difficulty.

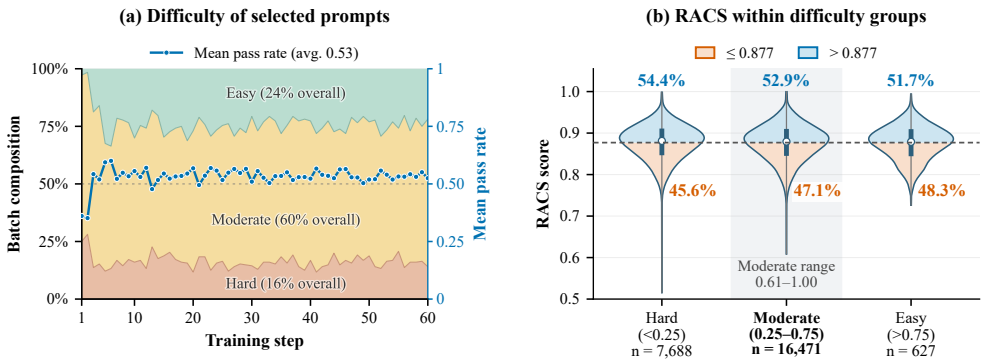


Figure 4: **(a)** Batch proportions of hard (< 0.25), moderate (0.25–0.75), and easy (> 0.75) prompts over 60 steps (shaded areas, left axis), with mean pass rate overlaid (blue line, right axis). **(b)** Corpus score distributions after excluding all-zero and all-one reward groups. Orange and blue indicate scores at or below and above the mean of 0.877, respectively.

The selected training data remains moderately difficult on average as training progresses. Figure 4(a) shows its difficulty composition and mean pass rate over 60 steps, with an overall mean pass rate of 0.53. The mean rises from 0.47 over steps 1–5 to 0.54 over steps 55–60, remaining close to 0.5 as the policy improves. Across all steps, 60% of selected prompts are moderately difficult, compared with 24% easy and 16% hard prompts (Figure 4(a)). Even as the policy improves during training, our online method maintains a majority of moderately difficult prompts in the training batches while retaining easy and hard examples.

At the corpus level, activation consistency varies among prompts of similar difficulty. Figure 4(b) examines scores across corpus prompts after excluding groups with all-zero or all-one rewards. All three difficulty categories contain both low- and high-consistency prompts, indicating substantial variation in activation consistency within each difficulty level. The distributions are similar across categories; approximately 76% of prompts score above 0.85. Within the moderate-difficulty category alone, scores range from 0.61 to 1.00. The dashed line marks the mean score rounded to 0.877. Of the moderately difficult prompts, 47.1% score at or below this reference. Thus, consistency differentiates prompts within the moderate-difficulty range favored by existing methods (Bengio et al., 2009; Bae et al., 2026b).

High-consistency prompts span all three difficulty categories. Among corpus prompts scoring above the mean of 0.877, 65.9% are moderately difficult, 31.6% are hard, and 2.5% are easy. During candidate sampling, easy and hard prompts receive the same weight as moderately difficult prompts with the same estimated consistency score. Unlike difficulty-based methods that favor intermediate pass rates (Sun et al., 2025; Bae et al., 2026b), our method can therefore sample high-consistency prompts across a broader difficulty range. This breadth is also evident in the training batches in Figure 4(a), which include easy and hard prompts alongside a majority of moderately difficult ones.

6.3 Understanding why RACS works

To understand the gains from high-RACS selection, we examine whether correct and incorrect response representations are more separable for high-RACS prompts than for low-RACS prompts. This analysis is motivated by prior work linking representation structure to reinforcement learning (Kumar et al., 2020; Eysenbach et al., 2022). We use the separability of these two groups to investigate how activation consistency may support learning.

We compare high-consistency selection (equation 7) with low-consistency selection on Qwen2.5-Math-1.5B. The latter reverses the sampling preference. The sampling formula and measurement procedure are given in Appendix B. To avoid zero scatter from singleton classes, we analyze prompts from the corpus with at least two responses per class. We compute the Fisher

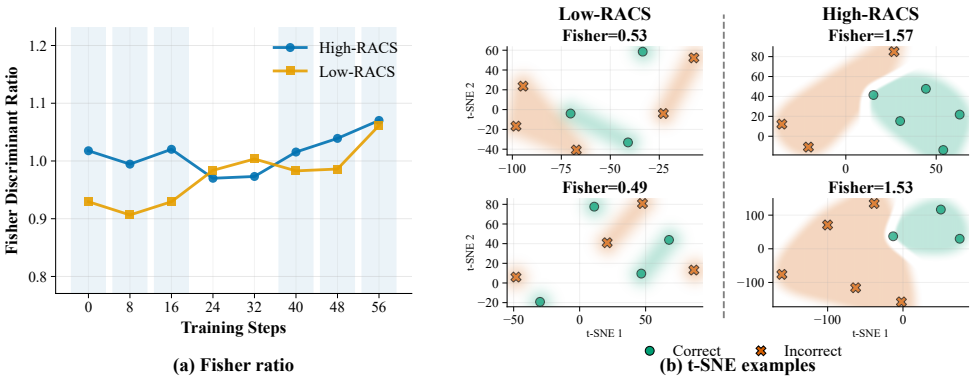


Figure 5: **Representation separability during training.** **(a)** Mean Fisher discriminant ratios for high-consistency and low-consistency prompts at eight measurement points. Blue shading marks the six points where high-consistency prompts have higher separability. **(b)** t-SNE visualizations for four examples at step 0, with low-consistency prompts in the left column and high-consistency prompts in the right column.

discriminant ratio in the original representation space:

$$F = \frac{\|\mu_{\text{correct}} - \mu_{\text{incorrect}}\|_2}{\frac{1}{2}(\sigma_{\text{correct}} + \sigma_{\text{incorrect}})}, \tag{8}$$

where μ and σ denote the centroid and mean scatter within each class. A higher ratio indicates greater separation between correct and incorrect responses relative to their within-group dispersion.

High-consistency prompts have higher Fisher ratios at six of the eight measurement points (Figure 5(a)). The relative advantage is approximately 10% at steps 0, 8, and 16. Overall, the separability advantage is strongest early in training and remains visible at several later stages. To observe the response distributions more directly, we visualize examples with high and low separability. The t-SNE visualizations in Figure 5(b) illustrate these distributional differences at step 0. Low-consistency examples show more intermingled or dispersed correct and incorrect responses, whereas high-consistency examples show clearer separation between the two groups. This separability advantage may help GRPO reinforce correct responses and suppress incorrect ones when training on the high-consistency prompts selected by RACS.

We believe that shared activation directions reflect a common solution strategy. High consistency can reflect a shared strategy across responses, with incorrect responses arising from differences in execution. Low consistency prompts may vary in solution strategies with variation in correctness, leading to more dispersed or overlapping groups. Under this interpretation, high-consistency selection favors prompts for which the model already has a basic solution strategy, rather than prompts that elicit inconsistent solutions. The downstream comparison is consistent with this interpretation. high-consistency selection outperforms low-consistency selection, reaching 75.34% aggregate accuracy versus 70.34%, a gain of five percentage points (Table 2 in Appendix B). Together, the geometric analysis and downstream results suggest that our method selects prompts with response structure that is useful for learning.

7 Conclusion

We presented RACS, a criterion for selecting training prompts in RLVR. It measures the directional consistency of activations across responses to the same prompt. To make this signal practical at corpus scale, we introduced a reference-set estimator that predicts scores from frozen prompt activation signatures, reducing rollout-based scoring to a small fraction of the corpus while preserving the ranking needed for selection. Under a fixed GRPO update budget, RACS consistently outperformed random selection and the strongest baseline in aggregate accuracy across the reported model settings. Our analyses show that the estimator can guide selection efficiently, and that high-consistency prompts tend to have more separable correct and incorrect response representations. These findings support response activation consistency as a useful criterion for online prompt selection in RLVR.

LIMITATIONS

Although our RACS-guided data selection yields consistent gains across multiple model scales and mathematical reasoning benchmarks, our work has several limitations. First, the global consistency score can also overlook useful structure within groups of responses. A low mean resultant length can reflect either widely scattered response directions or several distinct groups that are each internally consistent. If these groups correspond to different solution strategies, favoring high consistency may give too little weight to prompts with useful reasoning diversity. An extension would be to measure the consistency within each strategy separately from the differences between strategies. The challenge is to identify these groups from a small number of sampled responses while distinguishing differences in reasoning from other sources of variation in the activations. Second, the link between activation consistency and policy improvement remains only partly understood. Our analysis in Section 6.3 shows that correct and incorrect response representations tend to be more separable for high-RACS prompts. However, this finding does not establish how greater separation leads to better GRPO updates. These updates depend on how advantage-weighted gradients from different responses reinforce or cancel one another, which activation consistency alone does not determine. Connecting response geometry to these gradients would help explain when consistency predicts learning gains and why prompts with similar scores may differ in their training value.

AI USE STATEMENT

We used generative AI assistants to support manuscript drafting and language polishing, and to assist with literature retrieval and the identification of potentially relevant prior work. Our use of AI assistants was limited to these tasks. The authors reviewed and revised the AI-assisted text and checked the cited sources to verify their accuracy, relevance, and support for the associated claims. Separately, the language models studied in this paper generate responses as part of the RACS scoring procedure, GRPO training, and benchmark evaluation, as described in the methodology and experimental setup. The authors take full responsibility for the final manuscript, including its originality, citations, methods, results, and conclusions.

ETHICS STATEMENT

This work studies data selection for reinforcement learning with verifiable rewards using existing mathematical reasoning datasets and pretrained language models. It does not involve human-subject experiments or the collection of new personal data. Activation consistency does not guarantee correctness or safety, and our findings are limited to mathematical reasoning. Applications beyond this setting require further evaluation of reliability, fairness, and potential misuse.

REFERENCES

Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L Leavitt, and Mansheej Paul. Perplexed by perplexity: Perplexity-based pruning with small reference models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024. URL https://openreview.net/forum?id=0r0Bg1NY1X.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, Jeongyeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 700–719, Rabat, Morocco, March 2026a. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.30. URL https://aclanthology.org/2026.eacl-long.30/.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 700–719, 2026b.

Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *J. Mach. Learn. Res.*, 6:1345–1382, December 2005. ISSN 1532-4435.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

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.

Zhichen Dong, Zhanhui Zhou, Zhixuan Liu, Chao Yang, and Chaochao Lu. Emergent response planning in LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Ce79P8ULPY.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in neural information processing systems*, volume 35, pp. 35603–35620, 2022.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Aviral Kumar, Abhishek Gupta, and Sergey Levine. Discor: Corrective feedback in reinforcement learning via distribution correction. In *Advances in neural information processing systems*, volume 33, pp. 18560–18572, 2020.

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. In *Advances in neural information processing systems*, volume 35, pp. 3843–3857, 2022.

Shipeng Li, Zhiqin Yang, Shikun Li, Xiaobo Xia, Hengyu Liu, Xinghua Zhang, Gaode Chen, Dong Fang, Ying Tai, and Zhe Peng. Learnalign: Data selection for llm reinforcement learning with improved gradient alignment. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40417–40434, 2026.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling. *arXiv preprint arXiv:2502.11886*, 2025.

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.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 3(5), 2025.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Causal Representation Learning Workshop at NeurIPS 2023*, 2023. URL https://openreview.net/forum?id=T0PoOJg8cK.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for LLM reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=uwUkETPIJN.

Xinyu Tang, Zhenduo Zhang, Yurou Liu, Xin Zhao, Zhiqiang Zhang, JUN ZHOU, et al. Towards high data efficiency in reinforcement learning with verifiable reward. In *International Conference on Learning Representations*, volume 2026, pp. 32611–32635, 2026.

Roman Vashurin, Maiya Goloburda, Albina Ilina, Aleksandr Rubashevskii, Preslav Nakov, Artem Shelmanov, and Maxim Panov. Uncertainty quantification for llms through minimum bayes risk: Bridging confidence and consistency. *arXiv preprint arXiv:2502.04964*, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jianghao Wu, Jianfei Cai, Weiqiang Wang, Jin Ye, Daniel F. Schmidt, and Yasmeen George. Single-rollout hidden-state dynamics for training-free RLVR data selection. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=pvq2AhjOy1.

Quan Xiao, Debarun Bhattacharjya, Balaji Ganesan, Radu Marinescu, Katsiaryna Mirylenka, Nhan H. Pham, Michael Glass, and Junkyu Lee. The consistency hypothesis in uncertainty quantification for large language models. In *Proceedings of the Forty-First Conference on Uncertainty in Artificial Intelligence*, UAI '25. JMLR.org, 2025.

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.

Ningyuan Yang, Weihua Du, Weiwei Sun, Sean Welleck, and Yiming Yang. Gradalign: Gradient-aligned data selection for LLM reinforcement learning. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=bGUY6CJSbu.

Xinyu Zhou, Boyu Zhu, Haotian Zhang, Huiming Wang, and Zhijiang Guo. Efficient rlvr training via weighted mutual information data selection. *arXiv preprint arXiv:2603.01907*, 2026.

A IMPLEMENTATION DETAILS

Training protocol. We train all policies with GRPO (Shao et al., 2024) in the `verl` framework, using binary rewards for final-answer correctness. Each prompt has $K = 8$ sampled responses, generated at temperature 1.0. The training batch size is 256 prompts, and the learning rate is 5×10^{-6}. Each epoch consists of two training steps. We select $B = 512$ prompts per epoch, split into two batches of 256 prompts. We set the maximum prompt and response lengths to 1,024 and 2,048 tokens, respectively, and disable the KL penalty. All methods use the same training pool and hyperparameters and run for 60 update steps. These comparisons match the training-update budget, not the total rollout or wall-clock budget.

RACS configuration. At the start of each epoch, we uniformly sample $M = 512$ reference prompts and compute their scores under the current policy. We extract response activations at the last valid response token from three evenly spaced transformer blocks, including the final block, and average the layer-wise scores with equal weights. The score estimator uses $n = 8$ neighbors with cosine similarity and softmax weighting, as defined in equation 6. Prompt features are obtained from a frozen copy of the base model, with encoder inputs truncated to 1024 tokens. We sample training candidates using equation 7, with a default selection temperature of $\tau = 10^{-2}$. Only prompts whose response groups contain both correct and incorrect answers are retained for training. Retained rollouts are cached and reused for the subsequent GRPO updates, avoiding duplicate generation for accepted prompts. Additional implementation details, including automatic relaxation of the selection temperature, are given below. Our code is available at https://anonymous.4open.science/r/RACS-15D0.

RACS computation. For each prompt we take the K=8 GRPO rollout responses, run a single forward pass of the policy over the full input (prompt concatenated with response) to obtain internal activations, and extract the response-region representations. We select $|\mathcal{L}|$=3 evenly spaced layers across the network depth (including the final layer). The per-layer score is the mean resultant length of equation 1; the final score averages the three layers with equal weight. For each layer, the representation is the activation at the last valid token position within the response region.

Proxy predictor. The proxy uses a frozen copy of the base model. Each prompt is encoded with left padding and truncation to 1024 tokens; we extract the activation at the last valid token position from each of the same 3 evenly spaced layers and concatenate them into the activation-signature feature $\varphi(x)$. Propagation uses cosine similarity, n=8 neighbours, and a softmax temperature; reference prompts retain their exact score. All activation-signature features are cached within an epoch.

Table 2: **high-consistency versus low-consistency selection.** Accuracy (%) after GRPO training with the two selection strategies. "All" denotes aggregate accuracy over the 2,161 pooled test items. High-consistency selection performs better on four of the five benchmarks and in aggregate accuracy.

Method	GSM8K	MATH500	Minerva	AIME24	AMC23	All
high-consistency	**86.81**	**76.40**	**26.84**	13.33	**60.00**	**75.34**
low-consistency	81.88	70.20	22.43	**20.00**	55.00	70.34

B HIGH- AND LOW-CONSISTENCY SELECTION

We compare high-consistency selection (equation 7) with low-consistency selection on Qwen2.5-Math-1.5B. The latter reverses the sampling preference:

$$p_{\text{low}}(x) \propto \exp\left(-\frac{\widehat{\text{RACS}}(x) - \max_{x' \in \mathcal{D}} \widehat{\text{RACS}}(x')}{\tau_{\text{low}}}\right), \qquad \tau_{\text{low}} > 0, \tag{9}$$

where τ_{low} is the effective sampling temperature. For both strategies, we record selected prompts at step 0 and every 8 steps thereafter through step 56.

For each prompt's K=8 responses, we extract final-layer hidden states at the last valid response token. We compute the Fisher discriminant ratio in the original representation space using equation 8.

Algorithm 1 One epoch of RACS-guided data selection

Require: Corpus $\mathcal{D}$, current policy π_θ, frozen base model, epoch t
Require: Budget B, reference size M, response count K, layer set $\mathcal{L}$, neighbor count n, temperature τ

```
Encode prompts with the frozen base model and cache {φ(x) : x ∈ D}
Sample R_t ⊂ D uniformly, with |R_t| = M
for x ∈ R_t do
    Generate K responses using π_θ(· | x)
    Compute RACS(x) from response activations by equation 4
end for
Predict non-reference scores by equation 6; retain direct reference scores
Form the sampling distribution p by equation 7
Initialize selected prompts S ← ∅ and rollout cache C ← ∅
while |S| < B do
    Sample candidate prompts according to p
    Generate K responses per candidate and evaluate their rewards
    Retain prompts with both correct and incorrect responses, up to the remaining budget
    Add retained prompts to S and their responses and rewards to C
end while
return S and C for GRPO updates
```

Table 2 reports the full benchmark results for the two selection strategies.

C MEDIUM-CONSISTENCY SELECTION

For the medium-consistency group in the pilot study (Figure 1), we favor prompts whose predicted RACS is close to the median of the current corpus scores. Let $m = \mathrm{median}_{x\in\mathcal{D}} \widehat{\mathrm{RACS}}(x)$. The sampling probability is

$$p_{\mathrm{mid}}(x) = \frac{\exp\left(-\frac{1}{\tau}\left|\widehat{\mathrm{RACS}}(x) - m\right|\right)}{\sum_{x'\in\mathcal{D}} \exp\left(-\frac{1}{\tau}\left|\widehat{\mathrm{RACS}}(x') - m\right|\right)}, \qquad x \in \mathcal{D}, \tag{10}$$

We set the sampling temperature to $\tau{=}10^{-2}$. Prompts closer to the median receive higher sampling probabilities.