IRDS: Interpretable RLVR Data Selection via Verifier-Coupled Sparse Autoencoder Coverage

Anonymous ACL submission

Abstract

Reinforcement learning with verifiable rewards (RLVR) has become a key technique for enhancing LLM reasoning, yet its data inefficiency remains a major bottleneck. Existing methods address this problem only partially, each missing at least one of subset-level coverage, verifier signal use, or interpretability. To address this gap, we present IRDS (Interpretable RLVR Data Selection), which selects RLVR training instances on a sparse autoencoder (SAE) cluster basis so the selection itself is auditable on recognizable problem motifs. To select instances the model both fails on and can still learn from, we introduce a verifier-coupled coverage objective on the SAE basis and solve it by greedy log-determinant maximization. Experiments on three instruction-tuned models and six math reasoning benchmarks show that IRDS achieves the highest overall accuracy, exceeding the strongest baseline by $+3.9/+4.0$ pp on the two Qwen models and by $+0.5$ pp on Llama-3.1-8B, while running an order of magnitude cheaper than the trajectory-based baseline.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) trains large language models using a rule-based verifier that scores each rollout as correct or incorrect, replacing the human preference labels used in earlier alignment pipelines. The recipe drives most recent progress on mathematical reasoning benchmarks such as MATH and AIME, where DeepSeekMath and DeepSeek-R1 together with their successors converged on it as the default approach to math post-training (Shao et al., 2024; Guo et al., 2025). Under this recipe, every training instance contributes to the final policy through its sampled rollouts and the verifier's binary feedback, so the composition of the training set decides what the policy learns at each step. Three failure modes show up immediately in practice. An instance on which the policy already succeeds leaves nothing to repair; an instance on which it always fails produces no within-group reward variation; in both tails the group-relative gradient vanishes. A subset whose instances target overlapping weaknesses pays for the same correction many times. Selecting a subset that is simultaneously improvable, trainable, and diverse is therefore central to RLVR quality at a fixed data budget.

Several lines of work tackle this. Pointwise scalar methods (Li et al., 2025; Yi et al., 2026; Zeng et al., 2025) rank instances by a single utility such as a difficulty estimate or a learning-impact trajectory and keep the top entries, but a sort on one number cannot tell when two top-ranked instances target the same policy weakness. Diversity-aware methods (Yang et al., 2025; Ma et al., 2025; Tang et al., 2025) embed instances in a learned space and pick a spread-out subset, but the embedding is built without the verifier signal, so the diversity budget is spent on mathematical structure that the policy may already solve. Gradient-based methods (Xia et al., 2024a; Zhu et al., 2025) prefer instances whose gradients align with a target signal, but the gradient lives in a dense parameter space whose directions resist human inspection. None of these lines jointly addresses failure relevance, trainability, and non-redundancy on an auditable basis.

We propose IRDS (Interpretable RLVR Data Selection), an offline data-selection method that builds a verifier-coupled coverage objective on top of a sparse-autoencoder representation of the training set. The sparse-autoencoder basis gives a coordinate that is more semantic than surface features such as length or notation density, yet more inspectable than dense hidden states; each axis carries a recognizable problem motif such as geometry, divisor counting, or multiple-choice templates. On this coordinate, IRDS attaches two verifier-coupled weights per instance that disen-

tangle which weakness still needs repair from how trainable the rollouts are, and combines the two into a verifier-coupled metric whose leading directions concentrate difficulty relative to trainability mass. A greedy log-determinant procedure on this metric returns a non-redundant subset that is simultaneously failure-relevant, trainable, and diverse. Across three instruction-tuned models and six math benchmarks under a shared GRPO recipe, IRDS attains the highest Overall accuracy at an order of magnitude lower cost than the trajectory-based baseline, while keeping the SAE block of the selection auditable.

Our contributions are as follows:

- To our knowledge, IRDS is the first RLVR data-selection method built on a sparse-autoencoder cluster basis, which makes the selection auditable on recognizable problem motifs.
- We design a verifier-coupled coverage objective on this basis that jointly targets failure-relevance, trainability, and non-redundancy, and solve it by greedy log-determinant maximization.
- Experiments on three instruction-tuned models and six math benchmarks show that IRDS achieves the best Overall accuracy at every budget, with margins of $+3.9/+4.0$ pp on the two Qwen models and $+0.5$ pp on Llama-3.1-8B, at order-of-magnitude lower selection cost than the trajectory-based baseline.

2 Related Work

Data Selection for RLVR. Recent RLVR data selection follows three lines, each addressing failure relevance, trainability, or coverage in isolation. The largest line of work attaches a scalar utility to each instance and keeps the top entries; utilities include difficulty estimates (Gao et al., 2025; Zhao et al., 2026), uncertainty scores (Yi et al., 2026; Zeng et al., 2025), learning-impact trajectories (Li et al., 2025), and mid-difficulty rollout-replay (Sun et al., 2025; Fatemi, 2026). A second line organizes instances on a learned embedding and picks a spread-out subset (Tang et al., 2025; Bhattacharyya et al., 2025; Zhou et al., 2026), with the embedding built without the verifier signal. A third line aligns per-instance gradients with a target signal (Xia et al., 2024a; Zhu et al., 2025). IRDS differs from all three by representing instances on a sparse-autoencoder basis and lifting verifier-coupled coverage to the set level.

Sparse Autoencoders for Data Selection. Sparse autoencoders (Bricken et al., 2023; Cunningham et al., 2023; Gao et al., 2024; Bussmann et al., 2024) decompose neural activations into sparse codes over an overcomplete dictionary in which many directions correspond to recognizable concepts, which has made them a useful coordinate for downstream analysis. Two prior works use SAE features for data selection in instruction-tuning settings (Yang et al., 2025; Ma et al., 2025); IRDS differs from both in three concrete ways: (i) a verifier-coupled *failure* weight d_i that drives the budget toward SAE directions where the policy still answers wrong, which neither prior method uses because instruction-tuning has no rollout-level correctness signal; (ii) a verifier-coupled *trainability* weight r_i that downweights the all-fail and all-solve tails where group-relative RLVR advantages vanish, which is specific to RL with verifiable rewards; and (iii) a *set-level* coverage objective (greedy log-determinant on a whitened covariance) rather than the pointwise SAE-utility ranking used by both prior works, so redundancy between two equally hard-and-trainable instances is paid only once.

3 Preliminaries

Verifiable-reward reinforcement learning. RLVR replaces the learned reward model of preference-based alignment with a programmatic verifier that returns a binary correctness label per rollout. Concretely, given a training set $\mathcal{D} = \{x_i\}_{i=1}^{N}$ and a model π_θ, each training step draws an instance x_i, samples G_i rollouts from $\pi_\theta(\cdot \mid x_i)$, and the verifier returns one bit $r_{ij} \in \{0, 1\}$ per rollout. GRPO (Shao et al., 2024), the recipe we adopt throughout, then forms group-relative advantages from the G_i rollouts of each instance and updates π_θ with a PPO-style clipped objective. Under group-relative objectives, instances whose rollouts all receive the same verifier label produce degenerate within-group advantages; for a Bernoulli success model with true success rate P_i, the response-level reward variance contains a $P_i(1-P_i)$ factor that vanishes at the all-fail and all-solve tails. In our experiments we sample a fixed $G_i = G = 8$ rollouts per instance throughout, so the only signal consumed by selection is the per-instance success count

$s_i = \sum_{j=1}^{G} r_{ij} \in \{0, 1, \ldots, G\}$, an integer summary of the verifier's behavior on x_i under π_θ.

Sparse Autoencoders. A sparse autoencoder (SAE) (Bricken et al., 2023; Cunningham et al., 2023; Gao et al., 2024) decomposes an internal activation $h \in \mathbb{R}^{d_{\text{model}}}$ into a sparse nonnegative code over an overcomplete dictionary of size $D \gg d_{\text{model}}$, with a linear decoder reconstructing $\hat{h} = W_{\text{dec}}\, a(h) + b_{\text{dec}}$. We adopt the *BatchTopK* variant (Bussmann et al., 2024) to let the per-token sparsity adapt to token-level information content while preserving an average sparsity of k per token. For a batch $\mathcal{B}$ of B tokens, the encoder first produces post-ReLU activations $\tilde{z}_t = \mathrm{ReLU}\big(W_{\text{enc}}(h_t - b_{\text{dec}}) + b_{\text{enc}}\big)$, and BatchTopK then zeros all but the Bk largest entries among $\{\tilde{z}_{t,i} : t \in \mathcal{B},\, i \in [D]\}$:

$$a(h_t)_i \;=\; \tilde{z}_{t,i} \cdot \mathbb{1}\big[(t,i) \in \mathrm{top}_{Bk}(\{\tilde{z}\})\big]. \qquad (1)$$

Training regularization and per-policy widths are in Appendix E.4.

4 Method

Given the pool $\mathcal{D} = \{x_i\}_{i=1}^{N}$, the verifier counts $\{s_i\}$ from §3, and a target budget $K \ll N$, IRDS returns a size-K subset $S^\star \subseteq [N]$ following the pipeline in Figure 1. We represent each instance on an SAE cluster basis (§4.1), construct a verifier-coupled coverage metric on that basis whose leading directions concentrate difficulty relative to trainability mass (§4.2), optionally augment the per-instance design vector with a residualized gradient block (§4.3), and select a non-redundant subset by greedy log-determinant maximization on the resulting design vectors (§4.4). Numerical settings and full pseudocode are in the appendix.

4.1 SAE Cluster Coordinates

We want an instance coordinate that is more semantic than surface features such as length or LaTeX density, yet more inspectable than dense hidden states.Figure 1(panel 1) illustrates the pipeline. The sparse code introduced in Section 3 provides one, since many of the dictionary directions correspond to recognizable concepts.

SAE training. We train one BatchTopK SAE per base model on that model's final-layer activations, harvested from OpenMathInstruct-2 (Toshniwal et al., 2024) as an unsupervised activation source. The SAE optimizes only activation reconstruction and never sees verifier outcomes, evaluation labels, or DEEPSCALER selection statistics, so even if a prompt overlap exists between OpenMathInstruct-2 and the evaluation suite the dictionary cannot leak labels into selection. Throughout the paper we use a shared expansion factor of 32 and average per-token sparsity k=128; per-policy widths, regularization details, and held-out reconstruction quality are in Appendix E.4.

From sparse latents to cluster vectors. Individual SAE latents are too fine-grained for a corpus-level coordinate, since a single 40K-instance pool typically activates several tens of thousands of distinct latents and any individual one fires on only a small fraction of instances. We therefore cluster related latents into F=256 semantic clusters before the selection stage. Each kept latent is embedded by a graph-hybrid representation that concatenates a *presence direction* (from co-activation cosine similarity over instance-presence indicators) with a *residual direction* (from residual mass variation after regressing out the presence signal); we then run spherical K-means on this embedding with F=256 clusters under a fixed seed (F is distinct from both the selection budget K of Eq. (8) and the BatchTopK sparsity k=128). Each instance x_i is summarized by the per-cluster activation mass $m_i \in \mathbb{R}^{F}_{\geq 0}$ obtained as $m_{if} = \sum_{\ell \in \mathcal{C}_f} f_{i\ell}$ where $f_{i\ell} = \mathrm{mean}_t\, a(h_t)_\ell$ averages each latent's activation over the tokens of x_i and $\mathcal{C}_f$ indexes the latents assigned to cluster f. Each cluster admits a short human-readable label, such as circle–coordinate geometry or polynomial-root algebra, which lets a researcher audit on which content directions the selected subset spends its budget. The full activation pipeline and clustering recipe are in Appendix E.4 and E.7.

4.2 Verifier-Coupled SAE Coverage

With the cluster coordinate fixed, we now turn it into a coverage objective for RLVR.As shown in Figure 1(panel2), selection must satisfy two conditions on each picked instance that the cluster coordinate alone does not capture: the instance should be one the policy currently fails on, so training has something to repair, and its rollouts should produce a non-degenerate group-relative gradient, so the optimizer can act on the signal. Both conditions are functions of the per-instance success count s_i rather than of the cluster coordinate m_i, and pull on different parts of the success-rate range, with the

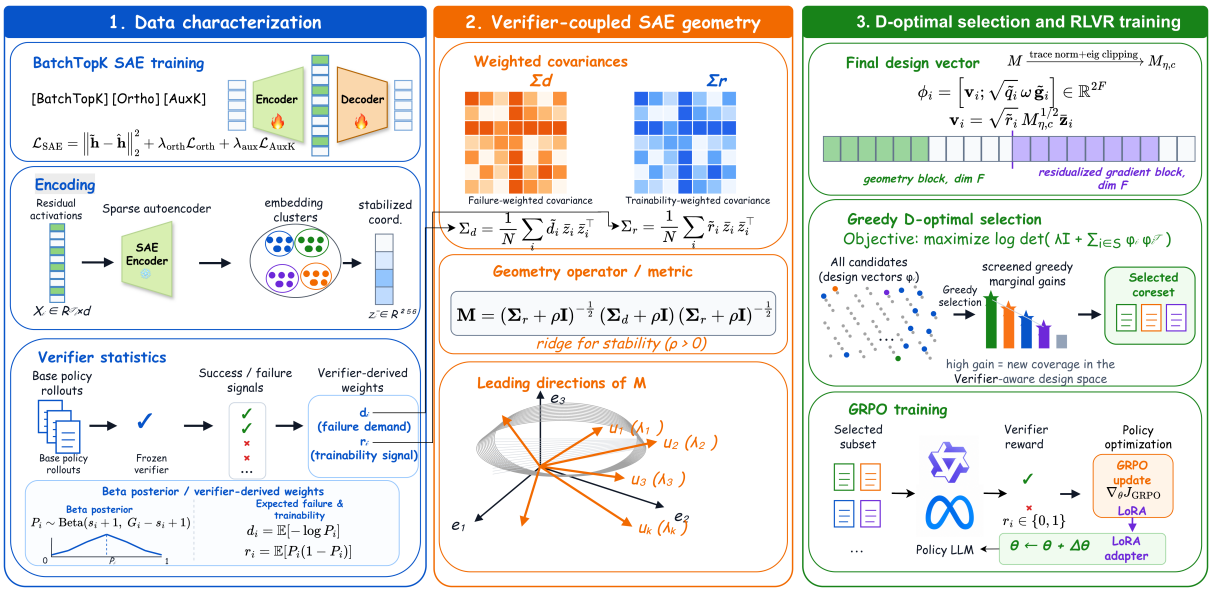

Figure 1: **IRDS overview. 1. Data characterization:** train a BatchTopK SAE on frozen residual activations, encode problems into stabilized SAE-cluster coordinates, and derive verifier weights d_i, r_i from base-policy rollouts. **2. Verifier-coupled SAE geometry:** form the failure-weighted and trainability-weighted covariances Σ_d, Σ_r, combine them into the ridge-regularized metric M, and use its leading directions as the verifier-aware geometry. **3. D-optimal selection and RLVR training:** apply trace normalization and eigenvalue clipping to obtain $M_{\eta,c}$, stack the geometry block with the residualized gradient block in ϕ_i, greedily maximize the log-determinant objective, and train GRPO on the selected coreset.

first condition pulling toward low s_i and the second toward the mid-band where the gradient is largest. The construction below derives one weight per condition, combines them on the cluster coordinate $\bar{z}_i$, and produces a coverage metric on $\mathbb{R}^F$ whose leading directions concentrate failure beyond what already-trainable instances cover.

Difficulty weight and trainability weight. For each instance x_i the verifier count s_i induces a Beta posterior on its success rate, $P_i \sim \mathrm{Beta}(s_i{+}1, G_i{-}s_i{+}1)$, from which we derive two scalar weights with closed forms in the digamma function ψ and the Beta moments:

$$d_i = \mathbb{E}[-\log P_i] = \psi(G_i{+}2) - \psi(s_i{+}1),$$
$$r_i = \mathbb{E}[P_i(1{-}P_i)] = \frac{(s_i{+}1)(G_i{-}s_i{+}1)}{(G_i{+}2)(G_i{+}3)}. \quad (2)$$

We call d_i the *difficulty weight*: it is monotonically decreasing in s_i, so it is high when the model fails often and low when it succeeds often. We call r_i the *trainability weight*: it is unimodal with peak near $s_i \approx G_i/2$, exactly where group-relative training signal is strongest. Concretely, the response-level advantage variance under group-relative optimization contains a $P_i(1{-}P_i)$ factor (Zeng et al., 2025; Gao et al., 2025), and our empirical per-instance gradient norms peak near intermediate success rates, so weighting by r_i downweights both the all-fail tail (where the gradient vanishes) and the all-solve tail (where there is nothing to learn). We mean-normalize each weight across the corpus to keep them on a common scale and write the resulting weights $\tilde{d}_i, \tilde{r}_i$ below. Because d_i and r_i are independent functions of the same success rate, they let the method treat which content needs coverage separately from how trainable the rollouts are.

Verifier-coupled metric. We seek a metric on the SAE coordinate space that lifts directions where verifier failures concentrate, relative to what high-trainability instances already cover. Let $\bar{z}_i \in \mathbb{R}^F$ denote a regularized version of the centered cluster mass m_i, detailed below. On this space we form the weighted covariances

$$\Sigma_d = \frac{1}{N}\sum_i \tilde{d}_i \bar{z}_i \bar{z}_i^\top, \quad \Sigma_r = \frac{1}{N}\sum_i \tilde{r}_i \bar{z}_i \bar{z}_i^\top, \quad (3)$$

and the whitened-coverage metric

$$M = (\Sigma_r{+}\rho I)^{-1/2}(\Sigma_d{+}\rho I)(\Sigma_r{+}\rho I)^{-1/2}. \quad (4)$$

Σ_d and Σ_r are SAE covariances reweighted by the difficulty and trainability weights respectively, so

whitening the former by the latter isolates SAE directions where difficulty exceeds trainability rather than directions that are simply hard or frequent. The leading directions of M maximize the regularized generalized Rayleigh quotient $v^\top(\Sigma_d + \rho I)v / v^\top(\Sigma_r + \rho I)v$, which is large on directions where difficulty mass exceeds trainability mass under the same ridge regularization. We then turn this metric into a per-instance design vector. In practice we apply finite-sample regularization to M (trace normalization and bounded eigenvalue clipping with parameters η, c, denoted $M_{\eta,c}$ in Appendix E.1); we also center $\bar{z}_i$ as described in the appendix. For notational simplicity, we write M for this regularized version in the main text and define the design vector with the trainability weight:

$$v_i = \sqrt{\tilde{r}_i}\, M^{1/2} \bar{z}_i, \tag{5}$$

so that instances contribute most to the $\log\det$ objective when they carry trainability mass and load on directions where difficulty most exceeds trainability. Regularization and ridge constants are listed in Appendix H and Table 9.

4.3 Gradient-Focused Block

The SAE coordinate captures content but is blind to update-direction information that the policy gradient would expose. To complement it, we add an optional auxiliary block whose contribution is concatenated to v_i and turned off by setting its scalar mixing weight ω (Eq. (7)) to zero.

For each instance x_i we compute the per-instance negative log-likelihood (NLL) of the frozen base model on its own greedy continuation y_i^{ref} ($\mathcal{L}_i = -\sum_t \log \pi_\theta(y_{i,t}^{\text{ref}} \mid x_i, y_{i,<t}^{\text{ref}})$), backpropagate to the full transformer-block parameters, and form a per-instance gradient signature $g_i \in \mathbb{R}^{p_g}$ via TRAK-style per-block Johnson–Lindenstrauss random projection (Park et al., 2023) ($p_g = L \cdot 64$ where L is the transformer-block count, so $p_g \in \{1792, 2304, 2048\}$ for the three models; see Appendix E.3). The gradient block then applies two operations to this vector. First, we ridge-residualize against the SAE coordinate so the block contributes only directions not already spanned by $\bar{z}_i$:

$$\mathbf{G}^\perp = \mathbf{G} - \bar{Z}\big(\bar{Z}^\top \bar{Z} + \rho_g I\big)^{-1} \bar{Z}^\top \mathbf{G}, \tag{6}$$

where $\mathbf{G} \in \mathbb{R}^{N \times p_g}$ and $\bar{Z} \in \mathbb{R}^{N \times F}$ stack the gradients and SAE coordinates row-wise (we use bold $\mathbf{G}$ for the gradient matrix to distinguish it from the rollout count G). After outlier clipping, PCA-whitening, and trace normalization to match the SAE block, we obtain $\tilde{g}_i \in \mathbb{R}^{p_g}$. Second, we attach a per-instance weight $\tilde{q}_i \propto \|g_i^\perp\|^{-\alpha}$. Empirically on DEEPSCALER, large residual-gradient norms are dominated by short instances or saturated ones ($s_i = G_i$, all rollouts succeed), whereas focused reasoning instances occupy a lower-norm band; the inverse-norm factor therefore downweights this high-norm tail and redirects selection toward the lower-norm band. The specific per-instance norm ranges are reported in Appendix E.3. Combining the two, the final design vector stacks the SAE block and the weighted gradient block:

$$\phi_i = \begin{bmatrix} \sqrt{\tilde{r}_i}\, M^{1/2} \bar{z}_i \\ \sqrt{\tilde{q}_i}\, \omega\, \tilde{g}_i \end{bmatrix} \in \mathbb{R}^{F + p_g}, \tag{7}$$

in which the scalar ω trades off SAE-block coverage against gradient-block coverage and $\omega{=}0$ recovers an SAE-only design. The full (ω, α) sweep on selection-time coverage diagnostics is in Appendix H.1.

4.4 Subset Selection

As shown in Figure 1(panel 3), Given the design vector ϕ_i, we want a subset whose rows span as many independent directions as possible, so that the budget is not wasted on instances that are mutually redundant. D-optimal design (Kiefer and Wolfowitz, 1960; Pukelsheim, 2006) captures this preference through the log-determinant objective

$$S^\star = \arg\max_{|S|=K} \log\det\Big(\lambda I + \sum_{i \in S} \phi_i \phi_i^\top\Big), \tag{8}$$

which under a linear-Gaussian model is proportional to the expected information gain about the coverage direction (Chaloner and Verdinelli, 1995). Let $A_S = \lambda I + \sum_{i \in S} \phi_i \phi_i^\top$ denote the running Gram matrix. Since the objective is monotone submodular, a greedy procedure that at each step adds the instance with the largest marginal gain $\Delta_i(S) = \log(1 + \phi_i^\top A_S^{-1} \phi_i)$ attains the standard $(1{-}1/e)$ guarantee (Nemhauser et al., 1978; Krause et al., 2008), and Sherman–Morrison rank-one updates on A_S^{-1} keep the per-step cost at $O(p^2)$ rather than $O(p^3)$. Algorithm 1 summarizes the full pipeline; Appendix I.1 compares against four classical alternatives on the same SAE coordinate (k-means clustering, facility-location coverage, statistical leverage scores, and a LESS-style (Xia et al., 2024a) influence proxy).

Algorithm 1 IRDS: verifier-coupled SAE subset selection. Full pseudocode in Appendix E.1.

Require: Instances $\{x_i\}$; verifier counts (s_i, G_i); cluster mass $\{m_i\}$; gradients $\{g_i\}$; budget K.

Posterior weights $\tilde{d}_i, \tilde{r}_i$ from (s_i, G_i).
Regularize $m_i \to \tilde{z}_i$; form Σ_d, Σ_r; whiten to M; set $v_i = \sqrt{\tilde{r}_i}\, M^{1/2} \tilde{z}_i$.
Residualize $g_i^{\perp} = g_i - \bar{Z}(\bar{Z}^\top \bar{Z} + \rho_g I)^{-1} \bar{Z}^\top g_i$; trace-normalize to $\widetilde{g}_i$; concatenate $\phi_i = [v_i;\ \sqrt{\widetilde{q}_i\, \omega}\, \widetilde{g}_i]$.
Greedily build $S^\star$ by the largest log-det marginal gains, using Sherman–Morrison rank-one updates.
return $S^\star$.

5 Experiments

Our experiments answer five questions about IRDS under a shared GRPO recipe: (i) do its selected subsets translate into final-policy accuracy gains over scalar, diversity, and trajectory baselines (§5.2); (ii) which design components in Section 4 are responsible for those gains (§5.3); (iii) do the gains come at an out-of-domain cost (§5.4); (iv) does the selected subset reflect content directions rather than surface artifacts (§5.5); and (v) is the picture stable across data-budget choices (§5.6).

5.1 Experimental Setup

Models and training. We post-train three instruction-tuned models, QWEN3-4B, QWEN3-1.7B, and LLAMA-3.1-8B-INSTRUCT, on the DEEPSCALER math-reasoning corpus (Luo et al., 2025) under a shared GRPO (Shao et al., 2024) recipe with LoRA (Hu et al., 2022) adapters and AdamW (Loshchilov and Hutter, 2019) optimization. Each baseline picks its own 20% training subset under this recipe so that only the data composition varies, and each model uses its own SAE and verifier-scored rollouts to instantiate the per-instance weights. Full hyperparameters and the SAE training recipe are in Appendix E.5 and E.4.

Evaluation and baselines. We evaluate each trained policy on a six-benchmark math suite drawn from MATH500, AIME24, AIME25, AMC23, MINERVA MATH, and OLYMPIADBENCH (Appendix E.8.2). For each problem we draw n=16 stochastic samples at temperature 0.6, top-p=0.95, with a 4k response budget, and score each sample as correct/incorrect against the ground-truth answer; mean@16 reports the average per-problem pass rate over the 16 samples, then averaged across the problem set. We compare against RANDOM sampling, perplexity-pruned subsets (PPL-TOP, PPL-MIDDLE, Ankner et al., 2024), instruction-following difficulty (IFD, Li et al., 2024), token-length ranking (TOKEN-LENGTH, Xia et al., 2024b), trajectory-based LIMR (Li et al., 2025), and the offline density/PageRank/determinantal-point-process (DPP) method DEPO (Tang et al., 2025).

5.2 Main Results

IRDS reaches the highest Overall accuracy on each of the three models (Table 1), scoring 67.0 on QWEN3-4B, 54.2 on QWEN3-1.7B, and 22.6 on LLAMA-3.1-8B-INSTRUCT, with Overall improvements of +3.9, +4.0, and +0.5 pp over the strongest baseline. On the larger MATH500 subset (500 problems), gains are +2.2 pp on QWEN3-4B (91.3 vs. LIMR 89.1) and +3.5 pp on QWEN3-1.7B. The largest per-benchmark gains occur on the competition subsets, which we note are also the smallest test sets (AIME24: 60, AIME25: 30, AMC23: 46 problems): on QWEN3-4B, IRDS gains +11.4 pp on AIME24 (42.0 vs. LIMR 30.6), +7.1 pp on AMC23 (79.5 vs. PPL-TOP 72.4), and +6.6 pp on AIME25 (35.6 vs. DEPO 29.0); QWEN3-1.7B matches this pattern on the same three subsets (+6.1/+6.9/+7.1 pp). On LLAMA-3.1-8B-INSTRUCT the Overall margin is smaller (+0.5 pp) and individual baselines lead on the smaller competition subsets.

5.3 Ablations

Components contribute unequally but each is non-redundant. To attribute the headline gain to specific design choices, we train six component-removal variants under the shared recipe (Table 2; per-subset breakdown in Appendix A). The ablation reveals an asymmetric structure across the three models. The trainability weight r_i is the *structurally critical* component: replacing the pair by d_i alone (i.e. removing r_i) collapses the method by 4.1 to 19.1 pp because pure difficulty-mass selection concentrates the budget on all-fail instances where group-relative advantage variance is near zero. The difficulty weight d_i is a *smaller but non-trivial refinement*: replacing the pair by r_i alone (i.e. removing d_i) costs 0.7 to 3.2 pp because r_i already selects the mid-success band that gradient-based training favors, but lacks d_i's drive toward the SAE directions where the policy still fails most. The whitened metric M and the set-level greedy procedure each carry the next-largest fraction of the lift: setting M=I drops Overall by up to −9.2 pp, and substituting the pointwise product $d_i r_i$ for the

Model	Method	Overall	Olympiad	MATH500	Minerva	AIME24	AMC23	AIME25
QWEN3-4B	RANDOM	56.2	53.3	84.5	39.4	22.7	62.4	21.9
	TOKEN-LENGTH	53.9	50.2	85.3	38.9	18.4	55.2	20.2
	IFD	60.4	58.3	86.2	40.3	28.4	68.9	28.1
	PPL-TOP	60.4	57.9	87.3	40.8	28.6	72.4	27.9
	PPL-MIDDLE	62.9	60.9	88.9	42.9	27.9	67.7	25.6
	DEPO	62.8	60.9	88.8	41.8	28.5	69.0	29.0
	LIMR	63.1	61.1	89.1	42.7	30.6	68.6	27.7
	IRDS	**67.0**	**65.6**	**91.3**	**43.8**	**42.0**	**79.5**	**35.6**
QWEN3-1.7B	RANDOM	46.5	44.0	72.6	30.0	13.9	48.9	14.6
	TOKEN-LENGTH	43.1	39.5	74.8	27.2	10.7	37.5	12.3
	IFD	44.9	41.9	74.1	28.3	11.7	43.2	10.8
	PPL-TOP	45.1	42.0	73.9	29.8	12.2	45.0	9.2
	PPL-MIDDLE	46.7	43.9	75.9	29.5	12.3	42.4	12.7
	DEPO	49.4	46.9	77.1	31.1	14.5	47.6	15.4
	LIMR	50.2	47.6	79.0	31.7	17.2	45.1	13.5
	IRDS	**54.2**	**52.2**	**82.5**	**33.6**	**23.3**	**55.8**	**22.5**
LLAMA-3.1-8B-INSTRUCT	RANDOM	22.1	15.5	54.6	25.3	3.0	23.0	**1.7**
	TOKEN-LENGTH	21.7	15.2	53.3	25.2	3.1	**23.1**	0.6
	IFD	21.2	14.6	53.0	25.0	3.2	19.8	**1.7**
	PPL-TOP	20.8	14.2	53.1	24.3	2.7	20.7	0.2
	PPL-MIDDLE	21.4	15.1	52.0	25.9	3.1	20.4	0.6
	DEPO	22.1	15.4	54.0	27.4	**4.1**	22.8	1.0
	LIMR	21.5	15.0	52.6	25.7	2.4	22.3	0.8
	IRDS	**22.6**	**15.7**	**55.0**	**28.3**	3.4	21.7	0.8

Table 1: Main results: mean@16 (%) on three models and six math benchmarks at the 20% data budget.

set-level objective costs up to -6.6 pp, so both whitening and the redundancy penalty matter. The SAE cluster basis is itself load-bearing on QWEN3-4B, where replacing it with a PCA-256 projection of dense gradients at the same dimension loses -4.5 pp Overall and -10.0 pp on AIME24; on the other two models the same substitution costs 1.1 to 2.1 pp, indicating that SAE content structure is most discriminative on the strongest base policy. The gradient block contributes the smallest margin (0.7 to 2.8 pp); the SAE-only variant (ω=0) already attains 64.2/53.0/21.9 Overall on the three models, so the SAE-coupled coverage objective carries the bulk of the lift and the gradient block is an optional refinement. Across the three budgets reported in §5.6, the structural rank order (r_i removal $\gg M{=}I \gg d_i r_i$ pointwise $\gg d_i$/gradient/PCA-256) holds at every budget on every model, so these deltas reflect stable component contributions rather than budget-specific noise. Per-subset breakdowns are in Appendix A; the (ω, α) sweep, reported on selection-time coverage diagnostics rather than downstream accuracy, is in Appendix H.1.

5.4 Out-of-Domain Transfer

We evaluate every trained checkpoint on HumanEval (Chen et al., 2021), LiveCodeBench (Jain et al., 2025), and ReClor (Yu et al., 2020) to span code, contest coding, and reading-comprehension reasoning (Table 3; per-task in Appendix C). IRDS attains the best three-task average on LLAMA-3.1-8B-INSTRUCT (48.2 vs. 47.9) and QWEN3-1.7B (49.9 vs. 49.5), and is 1.0 pp below RANDOM on QWEN3-4B (64.9 vs. 65.9, concentrated on LIVECODEBENCH), about a fourth of the corresponding +3.9 pp in-domain lift.

Variant	QWEN3-4B	QWEN3-1.7B	LLAMA
IRDS (full)	**67.0**	**54.2**	**22.6**
ω=0 ($-$gradient)	64.2	53.0	21.9
$M{=}I$ ($-$whitening)	57.8	50.3	20.8
r only ($-d_i$)	63.8	52.9	21.9
d only ($-r_i$)	47.9	44.7	18.5
$d_i r_i$ pointwise	60.4	51.2	21.2
dense ϕ_i ($-$SAE basis)	62.5	52.1	21.5

Table 2: Component-removal ablation: Overall mean@16 (%) at 20% budget. Each indented row removes or substitutes one component of IRDS. Columns: QWEN3-4B, QWEN3-1.7B, LLAMA-3.1-8B-INSTRUCT.

5.5 Auditing the Selected Subset

We check that the selected subset reflects content directions rather than surface features via four diagnostics that all agree across the three models: high-difficulty localization, cross-policy consis-

Method	LLAMA-3.1 -8B-INST	QWEN3 -1.7B	QWEN3 -4B
RANDOM	45.5	48.3	**65.9**
TOKEN-LENGTH	47.6	48.5	63.1
IFD	46.4	49.5	65.0
PPL-TOP	47.9	48.6	63.2
PPL-MIDDLE	46.8	49.5	64.7
DEPO	46.9	48.6	63.1
LIMR	46.5	49.0	64.5
IRDS	**48.2**	**49.9**	64.9

Table 3: Out-of-domain transfer (%): three-task average over HumanEval, LiveCodeBench, and ReClor. Llama-3.1-8B-Inst denotes LLAMA-3.1-8B-INSTRUCT.

tency, surface-artifact non-explanation, and label-shuffle falsification.

High-difficulty localization. The leading eigenvectors of M load on instances with unusually low verifier success: on each of the three models, the median success count of the top-25 highest-projection instances is well below the corpus median, so the directions on which the method spends its budget coincide with the high-difficulty regions the metric is meant to lift.

Cross-policy consistency. On each of the three policies' independently trained SAE bases, the leading axes of M label the same set of recognizable problem motifs (geometry, divisor counting, polynomial roots, multiple-choice stems), so the method reaches comparable content directions across architectures.

Surface-artifact non-explanation. A linear regression of the selection indicator on six surface features (length, bracket count, digit density, multiple-choice stem indicator, and two notation-density statistics) explains a negligible fraction of the indicator variance on every model. Conversely, ablating the verifier-coupled SAE block (i.e. replacing $M^{1/2}\bar{z}_i$ with $\bar{z}_i$) raises the per-instance marginal-gain variance several-fold above a mass-matched random control. Together these two checks are inconsistent with a purely surface-driven explanation of the lift.

Label-shuffle falsification. A global verifier-label shuffle collapses the top eigenvalue of M and drives the leading-subspace overlap with the true-label M to near zero, which is the falsification check we expect to pass and which the method indeed passes.

Concrete examples in Appendix J illustrate the method-level mechanism: IRDS picks a high-difficulty multiple-choice question (MCQ) stem exemplar but rejects a near-duplicate from the same template, and routes a contrastive number-theory cluster to a mid-success instance with non-zero trainability weight. A full-spectrum feature-amplification scan (residual-stream injection along each of the 256 M-eigenvectors at varying scales) and a base-vs-trained spectrum-invariance result are in Appendix K.

5.6 Budget Sensitivity

IRDS's lift is robust across data budgets. Training every (method, model) pair at 10% and 30% of DEEPSCALER under the same recipe (Table 4), IRDS retains the highest Overall accuracy at every budget on the three models. Per-benchmark numbers at 10% and 30% are in Appendix D.

	QWEN3-4B			QWEN3-1.7B			LLAMA-INST		
Method	10%	20%	30%	10%	20%	30%	10%	20%	30%
RANDOM	54.7	56.2	58.0	45.4	46.5	48.0	21.1	22.1	22.4
TOKEN-LENGTH	51.5	53.9	56.0	41.3	43.1	44.9	19.9	21.7	22.6
IFD	59.7	60.4	61.9	44.2	44.9	45.8	21.0	21.2	21.4
PPL-TOP	59.3	60.4	63.1	44.3	45.1	47.7	19.2	20.8	22.1
PPL-MIDDLE	62.3	62.9	65.2	45.7	46.7	48.8	21.2	21.4	22.0
DEPO	61.8	62.8	65.2	48.6	49.4	51.3	20.5	22.1	22.3
LIMR	62.3	63.1	65.5	50.0	50.2	52.1	21.1	21.5	23.0
IRDS	**63.8**	**67.0**	**68.8**	**51.4**	**54.2**	**55.6**	**21.9**	**22.6**	**23.5**

Table 4: Data-budget sweep: Overall mean@16 (%) at 10/20/30% of DEEPSCALER. Llama-Inst denotes LLAMA-3.1-8B-INSTRUCT.

6 Conclusion

We presented IRDS, an offline RLVR data-selection method that picks a non-redundant subset on a sparse-autoencoder cluster basis through a verifier-coupled coverage objective. The metric M disentangles where the model still fails (d_i) from where it can still learn (r_i), and greedy log-determinant maximization turns these per-instance signals into a set-level selection that pays for each weakness only once. Across three instruction-tuned models and six math benchmarks, IRDS attains the highest Overall accuracy at an order-of-magnitude lower selection cost than the trajectory-based baseline, with a small in-domain margin on LLAMA-3.1-8B-INSTRUCT and a small out-of-domain regression on QWEN3-4B. By selecting on a human-readable SAE basis, IRDS shows that the verifier-coupled component of RLVR data selection can be made auditable without losing in-domain accuracy.

Limitations

IRDS estimates the verifier-coupled weights once under the frozen model, so the failure profile that the method targets can drift away from the model's actual profile after many updates; a semi-online refresh of these weights on top of the fixed SAE cluster basis is a natural extension that we leave to future work. The method also requires a model-specific SAE trained on the model's final-layer activations, which restricts its application to settings where such activations are accessible. We validate the method only on math RLVR under GRPO, so transfer to non-math RLVR domains and to non-GRPO algorithms remains to be demonstrated. Finally, all reported numbers come from a single training seed per cell, so seed-level RL variance is not captured.

Ethics Statement

IRDS selects existing instances from publicly released math datasets (DEEPSCALER (Luo et al., 2025), OpenMathInstruct-2 (Toshniwal et al., 2024)) and does not generate, label, or release new problems. All models and benchmarks listed in §5.1 are public and used under their original licenses; no human subjects or personally identifying data are involved. The method amplifies instances on which a model already fails, so its intended use is improving the math-reasoning accuracy of post-trained models; IRDS does not change the verifier, reward signal, or policy architecture, and we do not foresee dual-use risks beyond those general to RLVR post-training.

References

Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L. Leavitt, and Mansheej Paul. 2024. Perplexed by perplexity: Perplexity-based data pruning with small reference models. *arXiv preprint arXiv:2405.20541*.

Chaitali Bhattacharyya, Hyunsei Lee, Junyoung Lee, Shinhyoung Jang, Il Hong Suh, and Yeseong Kim. 2025. FineScope: SAE-guided data selection enables domain-specific LLM pruning and fine-tuning. *Preprint*, arXiv:2505.00624.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, and 1 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Bart Bussmann, Patrick Leask, and Neel Nanda. 2024. BatchTopK sparse autoencoders. *arXiv preprint arXiv:2412.06410*.

Kathryn Chaloner and Isabella Verdinelli. 1995. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Mehdi Fatemi. 2026. Prioritized replay for RL post-training. *Preprint*, arXiv:2601.02648.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*.

Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang, and Liang Tan. 2025. Prompt curriculum learning for efficient LLM post-training. *Preprint*, arXiv:2510.01135.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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

Jack Kiefer and Jacob Wolfowitz. 1960. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366.

Andreas Krause, Ajit Singh, and Carlos Guestrin. 2008. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 7595–7628.

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

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li Li, Raluca Ada Popa, and Ion Stoica. 2025. DeepScaleR: Surpassing O1-Preview with a 1.5B model by scaling RL. Notion Blog and HuggingFace Dataset `agentica-org/DeepScaleR-Preview-Dataset`.

Da Ma, Gonghu Shang, Zhi Chen, Libo Qin, Yijie Luo, Lei Pan, Shuai Fan, Lu Chen, and Kai Yu. 2025. Task-specific data selection for instruction tuning via monosemantic neuronal activations. *Preprint*, arXiv:2503.15573.

George Ma, Zhongyuan Liang, Irene Y. Chen, and Somayeh Sojoudi. 2026. Do sparse autoencoders identify reasoning features in language models? In *International Conference on Machine Learning*.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. 2023. TRAK: Attributing model behavior at scale. In *International Conference on Machine Learning*.

Friedrich Pukelsheim. 2006. *Optimal Design of Experiments*. SIAM.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Penghui Li, Kai Li, and Chuan F. Wu. 2025. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*, pages 1279–1297. ACM.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. 2025. Improving data efficiency for LLM reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *Preprint*, arXiv:2506.05316.

Xinyu Tang, Zhenduo Zhang, Yurou Liu, Wayne Xin Zhao, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. 2025. Towards high data efficiency in reinforcement learning with verifiable reward. *Preprint*, arXiv:2509.01321.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. 2024. OpenMathInstruct-2: Accelerating AI for math with massive open-source instruction data. *Preprint*, arXiv:2410.01560.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024a. Less: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning*.

Tingyu Xia, Bowen Yu, Kai Dang, An Yang, Yuan Wu, Yuan Tian, Yi Chang, and Fei Huang. 2024b. Rethinking data selection at scale: Random selection is almost all you need. *arXiv preprint arXiv:2410.09335*.

Xianjun Yang, Shaoliang Nie, Lijuan Liu, Suchin Gururangan, Ujjwal Karn, Rui Hou, Madian Khabsa, and Yuning Mao. 2025. Diversity-driven data selection for language model tuning through sparse autoencoder. *arXiv preprint arXiv:2502.14050*.

Hao Yi, Yulan Hu, Xin Li, Sheng Ouyang, Lizhong Ding, and Yong Liu. 2026. Learn more with less: Uncertainty consistency guided query selection for RLVR. In *International Conference on Learning Representations*.

Qiying Yu, Zheng Zhang, Rui Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Dai, Gaohong Fan, Lingjun Liu, Xin Liu, and 1 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. ReClor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations*.

Yongcheng Zeng, Zexu Sun, Bokai Ji, Erxue Min, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Haifeng Zhang, Xu Chen, and Jun Wang. 2025. CurES: From gradient analysis to efficient curriculum learning for reasoning LLMs. *Preprint*, arXiv:2510.01037.

Yu Zhao, Fan Jiang, Tianle Liu, Bo Zeng, Yu Liu, Longyue Wang, and Weihua Luo. 2026. Difficulty-estimated policy optimization. *Preprint*, arXiv:2602.06375.

Yuzhe Zhou, Zhenglin Hua, and Haiyun Guo. 2026. Rethinking representativeness and diversity in dynamic data selection. *Preprint*, arXiv:2603.04981.

Erle Zhu, Dazhi Jiang, Yuan Wang, Xujun Li, Jiale Cheng, Yuxian Gu, Yilin Niu, Aohan Zeng, Jie Tang, Minlie Huang, and Hongning Wang. 2025. Data-efficient RLVR via off-policy influence guidance. *Preprint*, arXiv:2510.26491.

A Component-removal ablation: per-subset detail

Table 5 expands the body Table 2 with the per-subset mean@16 on all six benchmarks for every component-removal variant and every model at the shared 20% budget.

B Metric rationale and greedy selection guarantee

This appendix consolidates the design rationale for Section 4: the spectral interpretation of the verifier-coupled metric M together with the per-instance design vector v_i, and the $(1-1/e)$ submodular guarantee for the greedy procedure.

B.1 Spectral decomposition of M and v_i

Lemma 1 (Generalized Rayleigh form of M)**.** *Let $\widetilde{\Sigma}_r = \Sigma_r + \rho I$ and $\widetilde{\Sigma}_d = \Sigma_d + \rho I$ be the ridge-regularized covariances from Section 4.2, and let $M = \widetilde{\Sigma}_r^{-1/2}\widetilde{\Sigma}_d\widetilde{\Sigma}_r^{-1/2}$. The eigenpairs (λ_k, u_k) of M are in one-to-one correspondence with the solutions of the generalized eigenvalue problem $\widetilde{\Sigma}_d w_k = \lambda_k \widetilde{\Sigma}_r w_k$ via $w_k = \widetilde{\Sigma}_r^{-1/2} u_k$, and the leading direction maximizes the regularized Rayleigh quotient*

$$\rho(w) = \frac{w^\top \widetilde{\Sigma}_d w}{w^\top \widetilde{\Sigma}_r w}. \tag{9}$$

The lemma identifies M's top eigenvectors as the SAE coordinate directions where difficulty mass most exceeds trainability mass: directions with comparable difficulty-weighted and trainability-weighted covariance energy receive quotient near one, while directions with excess difficulty relative to trainability mass receive larger eigenvalues. Writing $M = \sum_k \lambda_k u_k u_k^\top$, the per-instance design vector $v_i = \sqrt{\tilde{r}_i}\, M^{1/2} \bar{z}_i$ from Eq. (5) then satisfies

$$\|v_i\|^2 = \tilde{r}_i \sum_k \lambda_k \, (\bar{z}_i^\top u_k)^2, \tag{10}$$

so an instance contributes to the objective in proportion to (i) its trainability weight $\tilde{r}_i$ and (ii) the energy it places on the leading u_k weighted by λ_k. Instances that are either untrainable ($\tilde{r}_i \to 0$) or that load primarily on low-λ_k axes have limited design energy regardless of their total cluster mass.

B.2 Greedy D-optimal guarantee

Let $d_\phi = \dim(\phi_i)$ ($d_\phi = F$ in the SAE-only regime and $d_\phi = F + p_g$ when the gradient block is active), and define the set function

$$J(S) = \log\det\Big(\lambda I_{d_\phi} + \sum_{i \in S} \phi_i \phi_i^\top\Big). \tag{11}$$

Since $\lambda I_{d_\phi} \succ 0$, the normalized objective $\bar{J}(S) = J(S) - J(\emptyset)$ with $J(\emptyset) = d_\phi \log \lambda$ is monotone submodular (Nemhauser et al., 1978; Krause et al., 2008), so the greedy procedure of Section 4.4 that adds at each step the instance with the largest marginal gain $\Delta_i(S) = \log\big(1 + \phi_i^\top A_S^{-1} \phi_i\big)$ attains

$$J(S_{\text{greedy}}) - J(\emptyset) \geq (1 - 1/e)\big(J(S^\star) - J(\emptyset)\big) \tag{12}$$

relative to the optimal size-K subset $S^\star$. The implementation in Algorithm 2 uses a screened candidate queue of size Q=1024 refreshed every R=256 additions, which matches the exact greedy objective within 0.1% on held-out checks.

C Out-of-domain transfer: per-task detail

Table 6 expands the body Table 3 with per-task results on HumanEval (HE) (Chen et al., 2021) for Python code generation, LiveCodeBench (LCB) (Jain et al., 2025) for contest-level coding, and ReClor (RC) (Yu et al., 2020) for reading-comprehension multiple choice. Decoding settings match the in-domain protocol (mean@16 for HE/RC, pass@1 for LCB) under each benchmark's standard prompt format; no system prompt is injected.

D Budget-sweep per-benchmark detail

The body Table 4 reports Overall mean@16 at the three measured budgets (10/20/30%). Tables 7 and 8 below give the full per-benchmark mean@16 breakdown at 10% and 30% budgets, with all seven baseline methods and IRDS on all three models and all six benchmarks.

E Implementation Details

E.1 Algorithm implementation

Algorithm 2 gives the full pseudocode behind the body skeleton in Algorithm 1, taking the encoded cluster masses $\{m_i\}$ (Appendix E.4) and per-instance gradients $\{g_i\}$ (Appendix E.3) as inputs. The greedy D-opt solver uses λ=1, a screened

Model	Variant	Overall	Olympiad	MATH500	Minerva	AIME24	AMC23	AIME25
QWEN3-4B	**IRDS (full)**	**67.0**	**65.6**	**91.3**	**43.8**	**42.0**	**79.5**	**35.6**
	ω=0 ($-$gradient block)	64.2	62.9	88.5	41.4	33.7	78.5	23.4
	M=I ($-$whitening)	57.8	55.6	84.0	36.8	24.5	74.9	27.3
	r-only ($-d_i$)	63.8	62.8	88.8	36.1	31.5	79.3	23.5
	d-only ($-r_i$)	47.9	44.8	76.8	31.3	11.6	54.0	11.5
	$d_i r_i$ pointwise	60.4	58.9	85.3	37.0	30.8	67.9	23.6
	dense ϕ_i ($-$SAE basis)	62.5	60.8	88.0	39.7	32.0	79.0	23.9
QWEN3-1.7B	**IRDS (full)**	**54.2**	**52.2**	**82.5**	**33.6**	**23.3**	**55.8**	**22.5**
	ω=0 ($-$gradient block)	53.0	51.1	81.1	32.7	19.9	55.8	16.6
	M=I ($-$whitening)	50.3	48.0	79.2	30.9	17.9	54.3	19.8
	r-only ($-d_i$)	52.9	51.2	81.5	29.9	19.5	58.4	17.5
	d-only ($-r_i$)	44.7	42.0	74.6	27.9	11.3	44.7	11.2
	$d_i r_i$ pointwise	51.2	49.2	79.5	30.6	19.5	50.8	17.9
	dense ϕ_i ($-$SAE basis)	52.1	50.0	81.0	31.8	19.6	56.1	17.5
LLAMA-3.1-8B-INSTRUCT	**IRDS (full)**	**22.6**	**15.7**	**55.0**	**28.3**	3.4	**21.7**	0.8
	ω=0 ($-$gradient block)	21.9	15.4	54.1	25.2	2.9	21.5	0.6
	M=I ($-$whitening)	20.8	14.4	52.8	23.8	2.6	21.1	0.6
	r-only ($-d_i$)	21.9	15.4	54.3	23.0	2.8	21.5	0.6
	d-only ($-r_i$)	18.5	12.6	49.7	21.5	1.6	17.4	0.4
	$d_i r_i$ pointwise	21.2	14.8	53.0	23.6	2.8	19.8	0.6
	dense ϕ_i ($-$SAE basis)	21.5	15.0	54.0	24.5	2.9	21.5	0.6

Table 5: Per-subset component-removal ablation: mean@16 (%) at 20% budget. Each indented row removes or substitutes a single component of IRDS.

	LLAMA-3.1-8B-INSTRUCT				QWEN3-1.7B				QWEN3-4B			
Method	HE	LCB	RC	Avg	HE	LCB	RC	Avg	HE	LCB	RC	Avg
RANDOM	64.6	10.0	62.0	45.5	65.2	20.0	59.7	48.3	82.3	**35.0**	**80.3**	**65.9**
TOKEN-LENGTH	67.1	**13.0**	62.7	47.6	64.0	20.0	61.3	48.5	80.5	31.0	77.7	63.1
IFD	64.1	**13.0**	62.0	46.4	65.2	22.0	61.3	49.5	**83.5**	33.0	78.3	65.0
PPL-TOP	**68.9**	**13.0**	61.7	47.9	64.0	19.0	62.7	48.6	79.3	32.0	78.3	63.2
PPL-MIDDLE	63.4	12.0	**65.0**	46.8	64.6	**23.0**	61.0	49.5	81.7	**35.0**	77.3	64.7
DEPO	68.3	10.0	62.3	46.9	62.8	22.0	61.0	48.6	78.1	33.0	78.3	63.1
LIMR	67.1	10.0	62.3	46.5	62.2	21.0	**63.7**	49.0	81.7	32.0	79.7	64.5
IRDS	**68.9**	**13.0**	62.7	**48.2**	**66.5**	**23.0**	60.3	**49.9**	82.3	32.0	**80.3**	64.9

Table 6: Out-of-domain transfer of math-RLVR-trained models on HumanEval (HE), LiveCodeBench (LCB) and ReClor (RC); mean@16 for HE/RC, pass@1 for LCB. **Bold** marks the per-column maximum.

candidate queue of size Q=1024 (refreshed every R=256 additions), and rank-one Sherman–Morrison updates of A^{-1} to avoid recomputing the full inverse at each iteration. The full selection completes in under 60 seconds on a single GPU.

E.2 Method hyperparameters

Table 9 compiles the numerical settings of every method stage and the shared GRPO recipe; they are referenced individually in the corresponding subsections of Section 4 and Appendix E.5.

E.3 Per-instance gradient extraction

The gradient block in Eq. (7) uses a single per-instance residualized gradient $g_i^{\perp}$ produced by the following one-pass pipeline against the frozen model, applied identically to QWEN3-4B, QWEN3-1.7B, and LLAMA-3.1-8B-INSTRUCT.

Per-instance NLL gradient (TRAK signature). For each instance x_i, we sample one greedy continuation y_i^{ref} from the same frozen base model used by the verifier scoring step, and compute the per-token negative log-likelihood loss $\mathcal{L}_i = -\sum_t \log \pi_\theta(y_{i,t}^{\text{ref}} \mid x_i, y_{i,<t}^{\text{ref}})$. We backpropagate $\mathcal{L}_i$ once and read $\nabla_\theta \mathcal{L}_i$ off every transformer-block parameter (token embeddings, `lm_head`, and the final norm are excluded as they are dominated by token-identity effects and uninformative for problem-level coverage). For each transformer block $\ell \in \{1, \ldots, L\}$ we concatenate the block's

Model	Method	Overall	Olympiad	MATH500	Minerva	AIME24	AMC23	AIME25
QWEN3-4B	RANDOM	54.7	52.1	82.4	37.7	18.1	61.1	16.7
	TOKEN-LENGTH	51.5	48.4	82.3	37.4	14.8	52.7	14.8
	IFD	59.7	58.1	85.5	39.5	22.5	68.0	21.5
	PPL-TOP	59.3	56.2	86.1	40.2	23.9	69.9	21.2
	PPL-MIDDLE	62.3	60.0	87.6	41.5	22.3	67.5	19.7
	DEPO	61.8	60.4	87.5	40.8	22.9	68.2	22.5
	LIMR	62.3	**60.5**	88.6	**42.5**	24.6	68.0	22.2
	IRDS	**63.8**	58.4	**89.0**	40.3	**32.6**	**76.8**	**26.0**
QWEN3-1.7B	RANDOM	45.4	43.3	70.8	29.3	11.6	48.0	10.6
	TOKEN-LENGTH	41.3	38.0	71.8	26.2	9.2	36.9	9.5
	IFD	44.2	41.4	72.8	28.1	10.0	41.8	8.4
	PPL-TOP	44.3	41.8	73.5	29.6	10.0	44.8	6.8
	PPL-MIDDLE	45.7	43.7	74.5	29.3	9.1	40.8	9.4
	DEPO	48.6	45.0	76.0	30.5	11.5	47.4	11.9
	LIMR	50.0	47.2	78.2	**30.9**	14.2	44.9	10.5
	IRDS	**51.4**	**49.1**	**80.5**	29.7	**18.3**	**52.9**	**16.4**
LLAMA-3.1-8B-INSTRUCT	RANDOM	21.1	14.6	49.9	24.3	2.5	21.8	**1.5**
	TOKEN-LENGTH	19.9	14.2	49.3	23.4	2.8	21.1	0.3
	IFD	21.0	14.4	50.5	24.5	2.5	19.6	1.2
	PPL-TOP	19.2	13.6	49.4	22.8	1.9	18.2	0.1
	PPL-MIDDLE	21.2	14.9	49.7	25.7	2.8	20.1	0.4
	DEPO	20.5	14.6	50.5	26.2	**2.9**	**22.3**	0.9
	LIMR	21.1	14.8	52.2	25.2	1.7	22.1	0.7
	IRDS	**21.9**	**15.3**	**54.5**	**27.1**	2.7	21.0	0.4

Table 7: Main results at 10% data budget: mean@16 (%) by model and benchmark.

Model	Method	Overall	Olympiad	MATH500	Minerva	AIME24	AMC23	AIME25
QWEN3-4B	RANDOM	58.0	54.8	87.3	41.0	23.2	64.9	23.6
	TOKEN-LENGTH	56.0	52.2	88.8	40.1	18.6	57.4	20.4
	IFD	61.9	59.0	88.0	40.8	28.6	70.2	29.6
	PPL-TOP	63.1	61.4	91.7	43.2	30.3	74.6	29.0
	PPL-MIDDLE	65.2	62.9	92.5	44.3	28.9	70.6	26.9
	DEPO	65.2	63.4	92.5	43.8	30.5	71.4	29.8
	LIMR	65.5	63.0	92.8	44.5	32.2	71.4	29.4
	IRDS	**68.8**	**67.2**	**93.7**	**45.3**	**43.2**	**82.4**	**37.4**
QWEN3-1.7B	RANDOM	48.0	46.0	75.4	31.7	14.1	51.2	15.2
	TOKEN-LENGTH	44.9	40.9	77.9	28.4	10.9	39.2	12.6
	IFD	45.8	42.8	78.7	29.0	11.9	43.4	11.4
	PPL-TOP	47.7	45.0	79.2	31.6	12.7	46.9	10.2
	PPL-MIDDLE	48.8	46.1	79.3	30.9	12.5	44.0	13.2
	DEPO	51.3	49.1	80.1	32.6	14.7	50.2	16.1
	LIMR	52.1	49.7	82.1	33.5	18.6	46.4	14.1
	IRDS	**55.6**	**53.4**	**85.1**	**34.9**	**23.9**	**57.5**	**23.4**
LLAMA-3.1-8B-INSTRUCT	RANDOM	22.4	16.1	55.5	25.9	3.2	23.2	**2.0**
	TOKEN-LENGTH	22.6	15.4	55.6	25.6	3.3	**23.8**	0.7
	IFD	21.4	14.8	53.5	25.7	3.5	20.2	1.8
	PPL-TOP	22.1	15.2	57.2	25.9	2.9	22.9	0.3
	PPL-MIDDLE	22.0	15.3	53.5	26.6	3.3	20.6	0.9
	DEPO	22.3	15.6	54.7	28.7	**4.8**	23.0	1.6
	LIMR	23.0	16.1	56.0	26.8	2.9	**23.8**	0.9
	IRDS	**23.5**	**16.8**	**57.9**	**29.7**	3.6	22.3	0.9

Table 8: Main results at 30% data budget: mean@16 (%) by model and benchmark.

parameter gradients in a fixed name order into a single vector and Johnson–Lindenstrauss–project it to p_ℓ=64 dimensions using a Gaussian random matrix with a layer-deterministic seed; the matrix is regenerated on the fly with bounded memory and shared across all instances, so projected signatures are directly comparable across x_i. This is the per-block TRAK projection of Park et al. (2023).

Algorithm 2 IRDS: full pseudocode.

Require: Instances $\{x_i\}_{i=1}^N$ with verifier counts (s_i, G_i), raw cluster masses $\{m_i\} \in \mathbb{R}^F$, per-instance TRAK NLL-gradient signatures $\{g_i\} \in \mathbb{R}^{p_g}$ (Appx. E.3), budget K.
Require: Defaults $\rho{=}10^{-1}$, $\eta{=}0.5$, $c{=}2$, $\rho_g{=}10^{-3}$, $\omega{=}1$, $\alpha{=}1$, $\lambda{=}1$, $\varepsilon{=}10^{-3}$, screen size $Q{=}1024$, refresh $R{=}256$.
Ensure: Subset $S^\star \subseteq [N]$, $|S^\star| = K$.

Per-instance posterior weights.
$d_i \leftarrow \psi(G_i + 2) - \psi(s_i + 1)$
$r_i \leftarrow (s_i + 1)(G_i - s_i + 1)/[(G_i + 2)(G_i + 3)]$
$\tilde{d}_i, \tilde{r}_i \leftarrow$ mean-one rescaling
SAE metric block.
Regularize m_i to $\bar{z}_i$ (success-axis removal, q_{99} row clip, bucket-mean subtraction)
$\Sigma_d \leftarrow \frac{1}{N}\sum_i \tilde{d}_i\, \bar{z}_i \bar{z}_i^\top, \quad \Sigma_r \leftarrow \frac{1}{N}\sum_i \tilde{r}_i\, \bar{z}_i \bar{z}_i^\top$
$M \leftarrow (\Sigma_r + \rho I)^{-1/2}(\Sigma_d + \rho I)(\Sigma_r + \rho I)^{-1/2}$
$M_{\eta,c} \leftarrow U\ \mathrm{diag}(\mathrm{clip}(\Lambda^\eta, 1/c, c))\, U^\top$ with $M = U\Lambda U^\top$, trace-normalized
$v_i \leftarrow \sqrt{\tilde{r}_i}\, M_{\eta,c}^{1/2}\, \bar{z}_i$
Optional gradient block.
if $\omega > 0$ **then**
  $g_i^\perp \leftarrow g_i - \bar{Z}(\bar{Z}^\top \bar{Z} + \rho_g I)^{-1} \bar{Z}^\top g_i$
  $\widetilde{g}_i \leftarrow$ row-clip at q_{99}, unit-normalize, whitened PCA (ridge 10^{-4}, drop ridge-floor eigendirections), trace-match to SAE block
  $\bar{g} \leftarrow \frac{1}{N}\sum_j \|g_j^\perp\|, \quad \tilde{q}_i \leftarrow (\bar{g}/\max(\|g_i^\perp\|, \varepsilon\bar{g}))^\alpha$
  $\phi_i \leftarrow [\, v_i;\ \sqrt{\tilde{q}_i\, \omega}\, \widetilde{g}_i \,] \in \mathbb{R}^{F+p_g}$
else
  $\phi_i \leftarrow v_i \in \mathbb{R}^F$
end if
Greedy D-optimal selection.
$S \leftarrow \emptyset, \quad A^{-1} \leftarrow \lambda^{-1} I$
for $k = 1, \ldots, K$ **do**
  $\mathcal{Q}_k \leftarrow$ top-Q candidates by $\phi_i^\top A^{-1} \phi_i$ (refresh every R steps)
  $i^\star \leftarrow \arg\max_{i \in \mathcal{Q}_k \setminus S} \log(1 + \phi_i^\top A^{-1} \phi_i)$
  $S \leftarrow S \cup \{i^\star\}$
  $A^{-1} \leftarrow A^{-1} - \dfrac{(A^{-1}\phi_{i^\star})(A^{-1}\phi_{i^\star})^\top}{1 + \phi_{i^\star}^\top A^{-1} \phi_{i^\star}}$ (Sherman–Morrison)
end for
return $S^\star \leftarrow S$

Stacking across blocks gives the per-instance gradient signature $g_i \in \mathbb{R}^{p_g}$ with $p_g{=}L{\cdot}p_\ell$: $p_g{=}1792$ on QWEN3-1.7B ($L{=}28$), $p_g{=}2304$ on QWEN3-4B ($L{=}36$), and $p_g{=}2048$ on LLAMA-3.1-8B-INSTRUCT ($L{=}32$). SAE coupling enters in the next step (residualization), not in the gradient signature itself.

Residualization, row-clip, and whitening. We then apply the four-step gradient block construction of Algorithm 1: ridge residualization $g_i^\perp = g_i - \bar{Z}(\bar{Z}^\top \bar{Z} + \rho_g I)^{-1} \bar{Z}^\top g_i$ with $\rho_g{=}10^{-3}$, which removes from g_i any direction linearly explained by the SAE coordinate $\bar{z}_i$; row-clip at the 99th percentile of $\|g_j^\perp\|$ followed by row-renormalization to unit norm; PCA-whitening of the residual matrix (ridge 10^{-4}, dropping eigendirections below the ridge floor); and trace-normalization so that $\frac{1}{N}\sum_i \|\widetilde{g}_i\|^2$ matches the SAE block at $\omega{=}1$. The mean-normalized inverse-norm weight $\tilde{q}_i = (\bar{g}/\max(\|g_i^\perp\|, \varepsilon\bar{g}))^\alpha$ with $\bar{g} = \frac{1}{N}\sum_j \|g_j^\perp\|$ and $\varepsilon{=}10^{-3}$ (and $\alpha{=}1$ in the headline runs), used in Eq. (7), is then applied as a scalar row weight on the rescaled $\widetilde{g}_i$. As a sanity check on the residualization, regressing g_i on $\bar{z}_i$ across the 40K-instance pool yields $R^2{=}0.024$ globally on QWEN3-1.7B, with mean residual-norm reduction below 1% across success-count buckets, so the gradient block enters ϕ_i as substantially independent information rather than a near-duplicate of the SAE coordinate.

High-gradient-norm tail. The residual-gradient norm $\|g_i^\perp\|$ on DEEPSCALER grows monotonically with the verifier success count s_i: on QWEN3-1.7B the per-bucket mean increases from $\|g_i^\perp\|{\approx}7.4$ at the all-fail tail ($s_i{=}0$) to $\|g_i^\perp\|{\approx}11.1$ at the saturated tail ($s_i{=}G$), with the saturated-tail 95th percentile reaching ≈ 21 versus ≈ 14 at the mid-success bucket. The same monotone pattern holds on the other two models. The inverse-norm row weight $\tilde{q}_i$ in Section 4.3 downweights this saturated tail—short, easy, and all-success instances dominate the high-norm region, so suppressing it routes the budget toward focused-reasoning instances at intermediate success rates.

Component	Setting
Verifier probe	G=8 rollouts per prompt; Beta$(1,1)$ prior
Selection budget	K=8062 (20% of 40,309)
SAE coordinate	BatchTopK-Ortho, $\times$32 expansion, k=128; F=256 graph clusters
Stabilization	success-axis removal, bucket-mean residualization, q_{99} row clip
Target metric M	ridge ρ=0.1, shrinkage η=0.5, eigen-clip c=2
Gradient block	block scale ω=1, inv-norm exponent α=1, residualization ridge ρ_g=10^{-3}, gradient-norm floor ε=10^{-3}
D-opt solver	regularizer λ=1, screened queue 1024, refresh 256
GRPO training	LoRA r=64, α=128; lr 5×10^{-6}; 2 epochs; G=8 rollouts; 8$\times$ GPU

Table 9: IRDS hyperparameters. The (ω, α) sweep is in Appendix H.1.

E.4 SAE training

Each model uses its own SAE trained on that policy's final-layer math activations; no SAE is shared across the three policies. We use shared expansion factor 32 and sparsity k=128 throughout, yielding per-policy SAE widths D=81,920 / 65,536 / 131,072 for QWEN3-4B / QWEN3-1.7B / LLAMA-3.1-8B-INSTRUCT (d_{model}=2,560 / 2,048 / 4,096). The activation corpus is the OpenMathInstruct-2 (Toshniwal et al., 2024) math-instruction collection (300,000 problems spanning grade-school to competition-level math, drawn from public sources curated by NVIDIA for math-reasoning post-training), used here purely as an activation source for unsupervised SAE training. We run each policy on these instances and harvest last-layer residual-stream token activations, sampling at most 512 tokens per instance with seeded sampling for determinism; the resulting per-policy training set contains $\sim$20M tokens, written to disk in eight shards across 8$\times$ GPU. We remove known overlaps with the math evaluation suite when identifiable; the SAE receives no verifier reward, validation outcome, evaluation label, or downstream benchmark label, is fixed before selection, and is not updated during GRPO.

SAE training adds a decoder-orthogonality penalty $\lambda_{\text{ortho}} \sum_{i \neq j} (W_{\text{dec}}[:,i]^\top W_{\text{dec}}[:,j])^2$ to discourage duplicate dictionary directions and an AuxK auxiliary loss (Gao et al., 2024) to recover dead latents. Figure 2 reports the BatchTopK-Ortho sparsity sweep across all three model-specific SAEs at $k \in \{16, 32, 64, 128, 256\}$: held-out reconstruction loss decreases monotonically with k on every model, and the orthogonality penalty keeps the duplicate / dead-feature rate low without harming convergence. We use k=128 as the shared setting. The selected QWEN3-4B SAE checkpoint is evaluated on 200k held-out activation samples: normalized MSE 0.0834, mean L_0=128, active-latent count 80,251 of 81,920, dead-feature rate 2.0%. End-to-end SAE training wall-clock on one 8$\times$ GPU node is 18 min for QWEN3-1.7B, 27 min for QWEN3-4B, and 72 min for LLAMA-3.1-8B-INSTRUCT. We then encode the entire training set in eight shards; each instance has a nonnegative cluster-mass vector, used for selection after centering, residualization, and whitening and for interpretation in raw form.

E.5 RLVR training and evaluation

We adopt GRPO (Shao et al., 2024), implemented in the VeRL framework (Sheng et al., 2025), as the default RL algorithm. IRDS is an instance-selection method that operates upstream of the RL loop and is algorithmically compatible with other on-policy RLVR algorithms such as DAPO (Yu et al., 2025). At each step we sample G=8 rollouts per instance at temperature 1.0 and top-p=1.0, score with a rule-based binary verifier (1 if the response answer matches the ground-truth label, 0 otherwise), and form group-relative advantages as in standard GRPO. Training runs for 2 epochs on the selected 20% subset, with instance batch size 128, maximum input length 1,024 tokens, and maximum response length 4,096 tokens; all updates use LoRA adapters (Hu et al., 2022) (rank 64, scaling α_{LoRA}=128; distinct from the inverse-norm exponent α of Section 4.3) in bf16. Optimization is performed by AdamW (Loshchilov and Hutter, 2019) (β_1=0.9, β_2=0.999, weight decay 0.1) with learning rate 5×10^{-6}, 5 warmup steps, and gradient clip norm 1.0. The PPO clip ratio is 0.2, the KL coefficient against the frozen reference policy is 10^{-3}, and the entropy coefficient is 10^{-3}. All experiments are conducted on 8$\times$ NVIDIA H20 GPUs.

We monitor validation on MATH500 with n=16 samples per problem during training. Final evaluation uses identical decoding across all methods on the 3,230-instance union of math benchmarks (Appendix E.8.2), with n=16 samples per problem at temperature 0.6, top-p=0.95, and a 4k response budget; only the selected instance subset

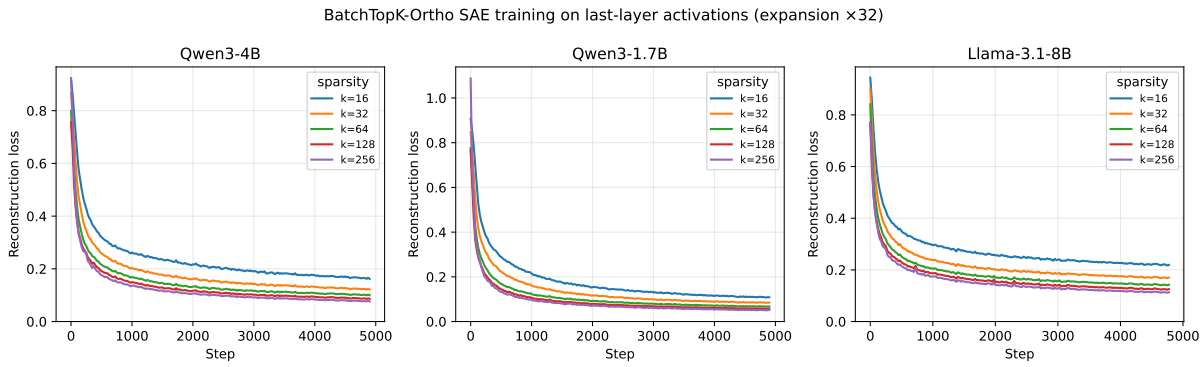

Figure 2: BatchTopK-Ortho SAE training: held-out reconstruction loss vs. step, per model and per sparsity k.

varies across methods.

E.6 Chat template

Each problem is wrapped as a single user turn and tokenized through the model's native chat template (QWEN3-4B, QWEN3-1.7B, and LLAMA-3.1-8B-INSTRUCT each use their own instruction template as released). We prepend a single system prompt that asks the model to produce a chain-of-thought followed by a boxed final answer; the rule-based verifier extracts the boxed expression and checks equivalence with the ground truth.

System prompt (training and evaluation).
Please reason step by step, and put your final answer within `\boxed{}`.

User turn.
{problem statement from DEEPSCALER}

The same template is used for selection-time verifier rollouts (G=8 per instance), for GRPO training rollouts, and for final benchmark evaluation, so the verifier always sees responses in the same format. Out-of-domain evaluation on HumanEval, LiveCodeBench, and ReClor uses each benchmark's standard prompt format (Section 5.4); no system prompt is injected on those tasks.

E.7 Cluster construction recipe

The 256 graph clusters are produced from the trained BatchTopK-Ortho SAE through a fixed construction procedure. We give a self-contained recipe so the same clusters can be reproduced from the SAE checkpoint and the training set without consulting the codebase.

Per-instance feature activations. For each instance i we encode every token through the SAE and average the resulting non-negative latent activations across tokens, giving a single 81,920-dimensional sparse vector $f_i \in \mathbb{R}^{81920}_{\geq 0}$ in which $f_{i\ell}$ is the mean activation of latent ℓ across the instance's tokens. This is the per-instance feature aggregation referenced in Section 4.1.

Latent frequency filter. We keep latents whose activation rate $\Pr_i[f_{i\ell} > 0]$ falls into the band $[\text{MIN_FREQ}, \text{MAX_FREQ}] = [0.01, 0.80]$. The lower threshold removes near-dead latents (rare in BatchTopK-Ortho with AuxK; in our case 2.0%); the upper threshold removes latents that fire on nearly every instance and would dominate any subsequent similarity metric. This leaves 42,473 latents on the QWEN3-4B SAE.

Latent embedding (graph-hybrid). Each kept latent ℓ is represented by the concatenation of two unit-normalized 64-dimensional vectors: **(a) presence direction.** For each of the 32 nearest co-activating latents (cosine similarity over instance-presence indicators), we keep the unit cosine score; we then PCA-project the resulting $42{,}473 \times 32$ presence-similarity matrix down to 64 dimensions and unit-normalize each row. **(b) residual direction.** We regress each latent's per-instance activation mass on the presence direction (ridge 0.001) and PCA-project the $42{,}473 \times 32$ residual matrix down to 64 dimensions, again unit-normalized. The latent embedding is the row-wise concatenation of the two, re-normalized to unit length; it lives in $\mathbb{R}^{128}$ and is what we call the *graph-hybrid* embedding.

Spherical k-means with $k = 256$. Run 20 iterations of mini-batch spherical k-means (BATCH = 8192) on the latent embedding with k=256. The fixed seed 0 produces the cluster labels reported throughout the paper. Cluster sizes range from 65

to 520 (median 143.5); mean within-cluster cosine similarity is 0.428. Latents outside the frequency band carry label -1 and contribute zero cluster mass.

Per-instance cluster activations. The cluster-activation matrix used as m_i in Section 4.1 is $m_{if} = \sum_{\ell:\text{cluster}(\ell)=f} f_{i\ell}$, of shape $N \times 256$, where N is the corpus size. After this the method applies the conditional residualization in Section 4.2, the verifier-coupled metric, and greedy selection.

E.8 Datasets and evaluation

This subsection documents the training set used for selection and the benchmarks used to score the resulting RLVR runs.

E.8.1 Selection training set

The selection corpus is the publicly released DEEPSCALER training set (Luo et al., 2025), normalized to a single ground-truth string per instance. Each entry contains (i) a problem statement `question`, (ii) a verifier-friendly `ground_truth` string, and (iii) per-instance artifact features (character length, LaTeX-density, equation count, answer length) used only for diagnostic regression and never read by the method. We do not subsample by source or by difficulty band; the corpus is the full DEEPSCALER preview release with deduplicated identifiers.

E.8.2 Evaluation benchmarks

Trained policies are evaluated on 3,230 problems under matched decoding settings. Table 10 lists the composition: MATH500, three competition benchmarks (AIME24/25, AMC23), MINERVA MATH, and English/Chinese OLYMPIADBENCH subsets, covering both in-distribution math reasoning and harder competition-style problems.

F Baseline Methods

All main baselines select the same 20% budget from the same DEEPSCALER training set and are trained with the GRPO recipe in Appendix E.5. For response-dependent SFT-style methods, we use the provided answer or a single generated response, while offline RLVR methods use the same verifier-sampling records as IRDS.

Scalar and SFT-style methods. RANDOM samples uniformly without replacement. TOKEN-LENGTH ranks by the total tokenized length of the problem and response, following the length-control use in SFT-scale data selection of Xia et al. (2024b). PPL-TOP and PPL-MIDDLE both follow the small-reference-model perplexity-pruning recipe of Ankner et al. (2024): a fixed scorer model assigns each instance a conditioned perplexity (mean negative log-likelihood of the response tokens given the instruction), and the method takes the top 20% for PPL-TOP or the middle 20% closest to the corpus median for PPL-MIDDLE. IFD uses Instruction-Following Difficulty (Li et al., 2024), computed as the discrepancy between the response likelihood with and without the instruction context. For the DEEPSCALER training set, these scores use the provided problem-answer pairs as the response text.

RLVR and uncertainty methods. LIMR follows the learning-impact trajectory criterion of Li et al. (2025); all instances are scored after the same warmup and selected at the same 20% budget. DEPO (Tang et al., 2025) is the Data-Efficient Policy Optimization pipeline of Tang et al., here labeled DEPO; it differs from the unrelated Difficulty-Estimated Policy Optimization of Zhao et al. (2026) (which uses an online difficulty estimator) and which we discuss separately in Related Work. Tang-DEPO has both offline and online stages; we implement its *offline selection component*: embed each instance with the model's hidden activations, build a k-nearest-neighbor graph in that embedding space, run PageRank on the graph for a centrality weight, prune to a mid-difficulty band using an external grading file, and run a PageRank-weighted DPP on the survivors to obtain a diverse 20% subset. We use it as a strong dense-graph + diversity baseline that combines centrality and redundancy control in a single non-SAE pipeline; the online explorability/replay stage of Tang-DEPO is not used here.

G Method Diagnostics

G.1 Surface-artifact residualization

Table 11 residualizes the selection design against ten surface features (length, brackets, digit density, multiple-choice stem indicator, etc.) before running the same greedy selection in the residualized space. The table displays the six most interpretable features; the fitted control also includes cluster-mass statistics used in the run-level diagnostics. The control isolates whether length and notation statistics alone reconstruct the selected subset.

Benchmark	subset	language	# problems	primary citation
MATH500	default	en	500	Lightman et al. (2024)
AIME24	default	en	60	AMC/AIME competition
AIME25	default	en	30	AMC/AIME competition
AMC23	default	en	46	AMC/AIME competition
MINERVA MATH	default	en	272	Lewkowycz et al. (2022)
OLYMPIADBENCH-EN	OE_TO_maths_en_COMP	en	674	He et al. (2024)
OLYMPIADBENCH-ZH-CEE	OE_TO_maths_zh_CEE	zh	1,240	He et al. (2024)
OLYMPIADBENCH-ZH-COMP	OE_TO_maths_zh_COMP	zh	408	He et al. (2024)
Total			3,230	

Table 10: Evaluation benchmark composition. Per-subset problem counts; the total is the union of all rows. The Overall column of Table 1 is the problem-count weighted average of the per-subset mean@16 values.

	ρ_{sel}			Mean shift Δ		
Feature	QWEN3-1.7B	QWEN3-4B	LLAMA-3.1 -8B-INSTRUCT	QWEN3-1.7B	QWEN3-4B	LLAMA-3.1 -8B-INSTRUCT
Characters	+0.038	+0.138	−0.037	+13.9	+51.2	−14.2
Words	+0.019	+0.134	−0.059	+1.09	+7.78	−3.35
LaTeX density	−0.100	−0.082	−0.000	−0.015	−0.012	−0.000
Digit density	−0.002	−0.057	+0.011	−0.000	−0.005	+0.001
Equation count	−0.006	+0.007	+0.004	−0.025	+0.030	+0.017
Operator density	−0.027	−0.053	+0.011	−0.001	−0.002	+0.000
Joint R^2 (6)	0.017	0.024	0.005		n/a	

Table 11: Surface-feature audit. ρ_{sel}: Pearson with the selection indicator; Δ: per-feature shift from the corpus mean.

G.2 SAE shortcut and reconstruction sanity

A specific concern about SAE-feature labels, articulated by Ma et al. (2026), is that a candidate "task feature" might fire on a small token set rather than on a content-level concept. We test this on QWEN3-4B by encoding 200 IRDS-top and 200 IRDS-bottom instances through the model's SAE, ranking the 81,920 features by the mean per-instance activation difference $\bar{\Delta}_f \equiv \mathbb{E}_{x\in\text{top}}[a_f(x)] - \mathbb{E}_{x\in\text{bot}}[a_f(x)]$, and computing the share of total activation mass concentrated on the top-50 activating tokens for each of the top-20 discriminative features. The ratio is 0.003–0.010 across these 20 features, roughly two orders of magnitude below the per-feature concentration ratios Ma et al. (2026) report for contrastively-selected reasoning features in middle layers; top-feature mass is broadly distributed over ∼thousands of tokens in standard math context, so none of the top-20 discriminative features are flagged as token-shortcut features under this diagnostic.

As an additional sanity check, on the same 200+200 instances the per-token reconstruction MSE in the SAE's normalized hidden space is mean 0.1387 (top-cluster) vs. 0.1394 (bottom-cluster), gap −0.0007 with Cohen's $d = -0.012$, so the IRDS signal cannot be attributed to differential SAE reconstruction quality between the two clusters. Every top-20 discriminative feature also fires at least once in every one of the 400 audit instances, so the IRDS distinction is a magnitude effect on near-universally-active features rather than a presence effect on rarely-active ones.

G.3 Runtime and scaling

Table 12 reports end-to-end wall-clock cost across methods on QWEN3-4B with one 8× GPU node, each method run under its native pipeline. The IRDS total of 7h (56.0 GPU-h) is an end-to-end accounting that includes SAE training (27 min wall-clock, ∼3.6 GPU-h), verifier-rollout scoring on N=40K instances at G=8 (∼ 5.5h, dominant), per-instance NLL-gradient extraction (∼ 25 min), cluster construction (graph-hybrid latent embedding + spherical K-means with F=256 clusters, Appendix E.7) and full-corpus SAE encoding of N=40K instances (∼ 35 min combined), and the greedy log-det selection itself (< 60 s, single GPU). LIMR, which requires an online RL trajectory rather than an offline scorer, costs 567.5 GPU-h, an order of magnitude more than IRDS at equal or better downstream accuracy. The shared downstream

Method	Wall-clock	GPU-h
RANDOM	<1 s	<0.01
TOKEN-LENGTH	5h 47m	46.3
PPL-TOP	5h 57m	47.6
PPL-MIDDLE	5h 57m	47.6
IFD	6h 12m	49.6
DEPO	6h 12m	49.6
LIMR	70h 56m	567.5
IRDS (ours)	**7h 00m**	**56.0**

Table 12: End-to-end selection cost on QWEN3-4B, one 8× GPU node. Downstream GRPO training is shared and excluded.

ω	α	QWEN3-1.7B	QWEN3-4B	LLAMA-3.1 -8B-INSTRUCT
0.0	n/a	957.8	959.6	987.9
0.5	0.0	2120.5	2531.8	2620.9
0.5	0.5	2179.8	2633.6	2717.0
0.5	1.0	2304.2	2826.1	2996.2
0.5	1.5	2442.8	3025.6	3250.0
1.0	0.0	2732.9	3362.2	3410.3
1.0	0.5	2812.6	3492.0	3544.6
1.0	**1.0**	**2976.5**	**3739.3**	**3899.4**
1.0	1.5	3154.7	3990.4	4211.0
2.0	0.0	3515.9	4413.0	4384.6
2.0	0.5	3614.4	4566.5	4554.1
2.0	1.0	3813.6	4860.6	4973.8
2.0	1.5	4026.0	5155.0	5332.5

Table 13: (ω, α) sweep at 20% budget: stacked-design target $\log\det\phi_i$. **Bold**: the trained configuration $(\omega, \alpha)=(1, 1)$.

GRPO consumes 4.3h on QWEN3-1.7B and 8.2h on QWEN3-4B per subset.

H Configuration Ablations

This appendix ablates the method's main configuration axes (cluster aggregation, gradient-block scale (ω, α), and selection budget K), and provides additional evidence that IRDS's gains are robust to these choices.

H.1 Gradient-block scale and inverse-norm exponent

The gradient block contributes to ϕ_i through two scalar knobs: the block-scale ω that balances SAE and gradient mass and the inverse-norm exponent α in the row weight $\sqrt{\tilde{q}_i\,\omega}$ with $\tilde{q}_i \propto \|g_i^{\perp}\|^{-\alpha}$ (see Eq. 7). Table 13 sweeps $\omega \in \{0, 0.5, 1, 2\}$ and $\alpha \in \{0, 0.5, 1, 1.5\}$ on both QWEN3-1.7B and QWEN3-4B, holding everything else at the trained configuration. Four diagnostics characterize the resulting subset: target $\log\det v_i$ on the SAE block (the paper's headline diagnostic), $\log\det\phi_i$ on the stacked design, the number of all-success (s_i=8) picks, and the share of non-degenerate ($s_i \in \{1, \ldots, G-1\}$) picks.

The ω=0 row turns the gradient block off and reproduces the SAE-only method. Raising ω moves selection mass into the gradient block, lifts $\log\det\phi_i$ as expected, and lowers $\log\det v_i$ on the SAE block alone, which is a coverage trade between the two blocks. The α=0 column replaces the inverse-norm row weight with a constant: the gradient block is then dominated by short, high-entropy instances and the s=8 count balloons (from 223 at α=1 to 357 at α=0 for QWEN3-1.7B, and 351 to 507 for QWEN3-4B). The trained configuration $(\omega, \alpha)=(1, 1)$ is a deliberate trade: equal block trace mass and mean-normalized inverse-norm weighting roughly halve the all-success count relative to α=0 while keeping the non-degenerate share at 76–78% and the SAE-block $\log\det v_i$ within 20–24 units of the SAE-only ceiling on both policies.

H.2 Cluster-level breakdown of methods and benchmark relevance

We use the cluster-mass matrix $\{m_i\}_{i=1}^{N}$ from Section 4.2 (each $m_i \in \mathbb{R}^F_{\geq 0}$ is the per-instance mass over the F=256 SAE clusters) to summarize how each method distributes its 20% budget over clusters. Table 14 reports the effective number of clusters covered $\mathbb{E}_{\text{sel}}[\,m_i\,] \mapsto \exp(-\sum_f q_f \log q_f)$ where q_f is the average cluster-mass distribution of the selected set, and the symmetric KL divergence of q_f from the pool distribution $\bar{q}_f$. IRDS attains a higher effective cluster count than every verifier-coupled one-dimensional ranker (Top-r_i, Top-d_i) on all three models, consistent with the D-optimal redundancy penalty actively widening cluster coverage; the random-selection baseline is included as a model-free reference.

The two largest labeled motifs IRDS over-allocates relative to TOP-r_i on QWEN3-4B are F149 (card-suit probability) at +0.41 log-ratio and F161 (drawn geometry) at +0.39 log-ratio; on QWEN3-1.7B the labeled F5 (divisor-count arithmetic) and F221 (binomial coefficients) clusters show similar over-allocation patterns relative to TOP-r_i. The contest-style subsets AIME24/25, AMC23, and OLYMPIAD weight probability, combinatorics, and geometry problems heavily, while MATH500 is more uniform. These cluster-level

Method	QWEN3-1.7B		QWEN3-4B		LLAMA-3.1-8B-INSTRUCT	
	N_{eff}	KL	N_{eff}	KL	N_{eff}	KL
RANDOM	122.1	0.0002	119.9	0.0002	138.8	0.0006
IRDS	**119.5**	0.0207	**128.7**	0.0226	**145.7**	0.0146
Top-r_i	113.8	0.0049	115.5	0.0034	144.6	0.0480
Top-d_i	113.1	0.0069	114.5	0.0072	127.3	0.0355

Table 14: Cluster-level allocation diagnostics at the shared 20% budget on all three models. N_{eff}: effective number of clusters covered; KL: symmetric KL divergence from the pool distribution. **Bold** marks IRDS (our method); IRDS covers more clusters than every verifier-coupled one-dimensional ranker (Top-r_i, Top-d_i) on all three policies.

shifts are consistent with the larger IRDS gains observed on contest-style subsets but do not by themselves establish a causal explanation.

I Coverage and Diversity Baselines

We compare IRDS against generic coverage and diversity baselines on the same SAE coordinate space and budget.

I.1 Coverage and diversity baselines on the same SAE space

Table 15 runs five geometric baselines (spherical k-means in $\bar{z}$ and v_i space, greedy facility location in $\bar{z}$, row-leverage scores) and a LESS-style influence proxy (cosine to the top-1 eigenvector of M) on the same SAE features at the shared 20% budget, scoring each on IRDS's target log-determinant. IRDS attains the highest target $\log\det$ among all geometric baselines.

J Qualitative and Representation Audits

J.1 Instance-level case study

Following the contrastive case-study style used in recent RLVR data-selection work, we choose examples by a predeclared diagnostic rule rather than by manual inspection. All four cases below are taken from the u_1 eigendirection of the QWEN3-4B verifier-coupled metric M. On QWEN3-4B the +end of u_1 is dominated by SAE cluster F78, whose top activating instances share the AMC/AIME multiple-choice stem with five lettered options; the −end places mass on number-theory clusters such as divisor counting and small-set cover.

A1: selected high-difficulty MCQ-stem exemplar. The highest-projection +end instance selected by IRDS is a digit-sum MCQ: "$S(n) = 1274$; which of (A) 1, (B) 3, (C) 12, (D) 1239, (E) 1265 could be $S(n+1)$?". The instance is all-fail under the eight verifier samples ($s_i = 0$), has projection 0.919, and is dominated by SAE cluster F78 (MCQ template with five lettered options) at mass 24. IRDS selects this instance early in the greedy procedure, identifying the failure-direction the metric is designed to lift.

B1: rejected near-template. The next-highest +end all-fail instance is a probability MCQ "six distinct integers from $\{1, \ldots, 10\}$, what is the probability that the second smallest is 3?" with the same five-option lettered stem and the same dominant cluster F78 at mass 24. Its projection is 0.907, nearly identical to A1, but it is rejected by IRDS. This is the intended D-optimal behavior: after A1 covers the MCQ-template direction, B1 contributes mostly redundant information, and its all-fail verifier count yields zero trainability weight.

A2: selected contrast direction. At the −end of u_1, IRDS selects a divisor-counting problem ("smallest natural number with exactly 70 divisors"). This instance has $s_i = 6/8$, projection magnitude 0.161, and dominant cluster F5 (small-integer divisor / factor enumeration). It shows that the same eigendirection is not a one-sided MCQ filter; the method also spends budget on a high-trainability instance for the contrastive number-theory cluster.

B2: IRDS-only high-trainability exemplar. An instance selected uniquely by IRDS is the small-set-cover question "$A_1, \ldots, A_n \subseteq \{1, \ldots, 10\}$ with $A_i \cup A_j \neq A$; find the maximum n". It lies on the +end side of u_1 at projection 0.108, but unlike B1 it has intermediate success ($s_i = 3/8$), so the trainability weight r_i is high. This is the core difficulty/trainability distinction: IRDS does not simply choose the hardest near-template instances; it routes a failure-relevant SAE direction to an instance that diversifies the selected set and still produces reward variation for RLVR.

These cases illustrate the method-level mecha-

Method	target $\log\det v_i$			Jaccard with IRDS		
	QWEN3-1.7B	QWEN3-4B	LLAMA-3.1 -8B-INSTRUCT	QWEN3-1.7B	QWEN3-4B	LLAMA-3.1 -8B-INSTRUCT
IRDS	**900.3**	**895.6**	**932.5**	1.000	1.000	1.000
D-opt on ϕ_i (rebuild)	881.0	872.4	840.7	0.326	0.258	0.111
Leverage on ϕ_i	895.7	883.8	814.1	0.185	0.161	0.078
k-means on ϕ_i	836.4	843.9	863.8	0.104	0.104	0.119
Facility location on ϕ_i	830.8	836.5	852.8	0.107	0.107	0.113
LESS-proxy on ϕ_i	815.4	812.8	820.7	0.101	0.133	0.122
Random	831.5	836.9	851.2	0.109	0.108	0.112

Table 15: Coverage and influence baselines on the same ϕ_i. Higher target $\log\det v_i$ is better.

nism behind the aggregate results. The verifier-coupled SAE metric identifies which content directions need coverage, while the $\sqrt{\tilde{r}_i}$ weighting and D-optimal design decide whether a concrete instance is a useful exemplar or a redundant failure example. Table 17 reports A1/B1/A2/B2 in compact form on QWEN3-4B, and Table 18 summarizes the eight SAE clusters with the largest target-direction score under M, along with their top keywords, maximum surface-feature correlation, and cluster-removal z-score.

K Spectrum-level interpretability of the SAE basis

Appendix J examined IRDS's behavior on the leading eigendirection u_1 of the verifier-coupled metric M via contrastive case studies. This appendix scales the analysis to the full $F = 256$ SAE cluster basis on QWEN3-1.7B: the population-level monosemanticity statistics on all 48,907 active SAE features, and a complete 256-eigenvector feature-amplification scan. All scans are inference-only on the trained QWEN3-1.7B IRDS checkpoint at roughly 30 GPU-min on 8×H20.

K.1 Feature morphology and population-level monosemanticity

Table 16 lists nine representative SAE features grouped into three morphological regimes, with three verbatim top-activating instances each. **(A) Numeric variation** (e.g. feature #27000: "Compute $C(5,1)$", "Compute $C(6,3)$", "Compute $C(9,8)$"): top activators share an identical text template differing only in numerical parameters. **(B) Semantic-coherent, lexically diverse** (e.g. feature #46942 on cars and miles-per-gallon): top activators share a clear concept with noticeably different surface form. **(C) Format / structural** (e.g. feature #7344 on Asymptote diagram blocks, feature #992 on 5-option multiple-choice stems): the feature fires on a recognizable cross-instance surface pattern. IRDS selects on the 256-d cluster-mass basis rather than on single features, so all three regimes contribute to coverage through their cluster memberships.

Population distribution rules out cherry-picking. Computing a per-feature monosemanticity statistic (the top-5 activation Jaccard against the cluster-mean activation pattern) on every active SAE feature on QWEN3-1.7B gives a population distribution with median 1.81, $p_{75} = 8.58$, $p_{95} = 21.20$, well above the 1.0 baseline expected if cluster assignment were random with respect to top-instance selection; 35.8% of features cross the ≥ 5.0 cutoff. The high-monosemanticity regime that Table 16 illustrates is therefore a population-level property, not a hand-picked tail.

K.2 Full 256-eigenvector spectrum scan

We run the same feature-amplification protocol on all 256 M-eigenvectors at $\alpha \in \{0.5, 1, 2, 5\}$ (Figure 3). Logit-shift magnitude is spread across the spectrum, with the strongest direction at rank 71 (eigenvalue λ_{71}=4.73, $|\Delta|$=2.50) and substantial mass beyond the named top-10. This motivates IRDS's design choice in §4.2 of running D-opt on the full 256-dimensional cluster basis rather than on a top-K truncation, so that the selection criterion has access to every direction of M that visibly moves the model output rather than only the eigenvalue-leading subset.

The same scan on the *untrained* base QWEN3-1.7B matches the trained checkpoint at Pearson r=1.00 (max drift 0.031 across all 256 axes), so the M-eigenbasis is intrinsic to the SAE cluster basis.

feat	template (Likert)	three representative top-activating instances
(A) Numeric variation (same template, different numerical parameters)		
#27000	binomial $C(n,k)$ compute	**1)** Compute C 5 1 . **2)** Compute C 6 3 . **3)** Compute C 9 8 .
#16072	$0.\overline{ab} \to$ frac	**1)** Express .<28> as a common fraction. **2)** Write 0.<43> as a simplified fraction. **3)** Express 0.4 5 as a common fraction.
#2301	random arrival-time word problem	**1)** Alice and Bob each arrive at a gathering at a random time ... **2)** Alice and Bob each arrive at a party at a random time bet ... **3)** A train arrives randomly sometime between 3:00 and 4:00 P ...
(B) Semantic-coherent, lexically diverse (same concept, different surface form)		
#46942	cars / miles-per-gallon	**1)** Ray's car averages 40 miles per gallon of gasoline, and T ... **2)** Karl's car uses a gallon of gas every 35 miles, and his g ... **3)** Margie's car can go 32 miles on a gallon of gas, and gas ...
#62446	Pascal's triangle properties	**1)** The pattern of Pascal's triangle is illustrated in the di ... **2)** In Pascal's Triangle, each number is the sum of the numbe ... **3)** Let f(n) be the base-10 logarithm of the sum of the eleme ...
#45192	circles geometry	**1)** Three coplanar circles intersect as shown. **2)** Three circles of radius 1 are externally tangent to each ... **3)** The same amount of steel used to create eight solid steel ...
(C) Format / structural (SAE captures non-mathematical surface regularities)		
#7344	Asymptote `[asy]` block	**1)** Thirteen blue and six green hexagonal tiles were used to ... **2)** In this diagram, both polygons are regular. **3)** Two identical rectangular crates are packed with cylindri ...
#992	5-option $(A)\ldots(E)$ MC	**1)** For a constant c, in spherical coordinates (, ,), find ... **2)** For a positive constant c, in spherical coordinates (, , ... **3)** Find the curve defined by the equation r = 2.
#32405	English-word arithmetic	**1)** Five million times eight million equals **2)** Three tenths plus four thousandths. **3)** Nine hundred forty-three minus eighty-seven equals

Table 16: **Nine SAE features in three monosemanticity regimes.** Each row shows the feature id and the three top-activating pool instances (verbatim duplicates collapsed; CJK runs rendered as [CJK]). **(A) Numeric variation**: top activators share an identical text template differing only in numerical parameters. **(B) Semantic-coherent, lexically diverse**: top activators share a clear concept with noticeably different surface form. **(C) Format/structural**: top activators share a non-mathematical format regularity (Asymptote diagrams, 5-option MC, English-word arithmetic). Across the full 65,536-feature population, 35% pass the monosemantic bar (top-5 Jaccard ≥ 0.30); only 0.44% are noisy, of which only 6 have top-1 activation > 0.4 (high-activation noisy fraction $< 0.01\%$).

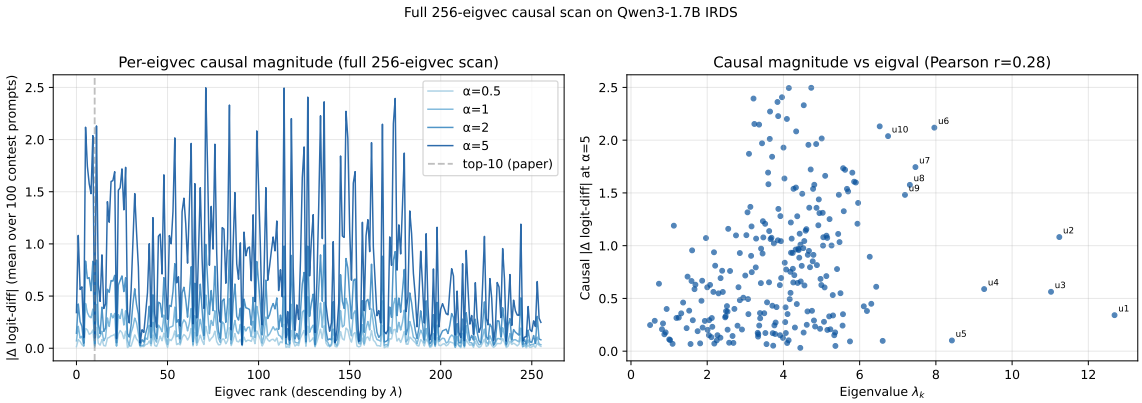

Figure 3: **Full 256-eigenvector feature-amplification scan on** QWEN3-1.7B**.** *Left:* $|\Delta$logit-diff$|$ vs eigenvector rank for $\alpha \in \{0.5, 1, 2, 5\}$. The dashed line at rank 10 marks the named-axis cutoff. *Right:* logit-shift magnitude at α=5 vs eigenvalue λ_k. Sensitivity mass is distributed across the full 256-d basis, with $\overline{|\Delta|}$=0.82 and $\max$ =2.50 at rank 71.

Card	Position on u_1	Instance (truncated)	s_i/G_i	$\lvert\text{proj}\rvert$	Cluster
A1	$+$end u_1, selected	Let $S(n)$ equal the sum of digits of n, $S(1507)$=13. For some n, $S(n)$=1274; which of (A) 1, (B) 3, (C) 12, (D) 1239, (E) 1265 could be $S(n{+}1)$?	$0/8$	0.919	MCQ template (F78)
A2	$-$end u_1, selected	Find the smallest natural number with exactly 70 natural divisors (including 1 and itself).	$6/8$	0.161	Divisor prototype (F5)
B1	$+$end u_1, rejected	Six distinct integers picked from $\{1,\ldots,10\}$. Probability that the second smallest is 3? (A) $\frac{1}{60}$, (B) $\frac{1}{6}$, (C) $\frac{1}{3}$, (D) $\frac{1}{2}$, (E) none.	$0/8$	0.907	Near-template (F78)
B2	$+$end u_1, IRDS-only	Subsets $A_1,\ldots,A_n \subseteq \{1,\ldots,10\}$ with $A_i \cup A_j \neq A$; find the maximum n.	$3/8$	0.108	Set-cover proto (F231)

Table 17: Case cards along the leading axis u_1 of M on QWEN3-4B.

F#	Top keywords	target-lift	max $\lvert\rho_{\text{artif}}\rvert$	removal z
78	θ/frac/length/12	0.274	0.07 (chars)	$+0.8$
5	divisors/exactly/smallest/16	0.234	0.07 (latex)	-2.0
7	regular/hexagon/octagon/area	0.104	0.09 (latex)	$+0.7$
54	$\surd$/prime/divisible/not	0.104	0.18 (chars)	$\mathbf{+2.0}$
147	$\triangle$/area/$\angle$/length	0.103	0.18 (eq. count)	$\mathbf{+2.3}$
33	θ/circle/correct/enter	0.081	0.09 (chars)	$+1.8$
218	prime/relatively/probability	0.077	0.11 (ans. len)	$\mathbf{+3.2}$
237	$\times$/squares/black/unit	0.059	0.30 (latex)	$+1.4$

Table 18: Top-lifted SAE clusters on QWEN3-4B ranked by target-direction score (top eight by target-lift). Bold: $\lvert z\rvert \geq 2$.