Guiding LLM Post-training Data Engineering with Model Internals from Sparse Autoencoders

Anonymous ACL submission

Abstract

Model internals encode rich information about how a large language model (LLM) processes its training data; however, post-training data engineering largely relies on external signals and ignores rich intrinsic signals lying in model internals. We propose SAERL, a data engineering framework for LLM reinforcement learning (RL). It models three intrinsic data properties: diversity, difficulty, and quality, using model internals extracted with Sparse Autoencoder (SAE), an advanced mechanistic interpretability tool. Each property grounds a concrete data engineering operation: SAE-space clustering with moderate batch mixing for batch diversity control, a difficulty proxy for easy-to-hard curriculum ordering, and a quality probe for data filtering. SAERL improves average accuracy by 3.00% over vanilla GRPO and reaches target accuracy with 20% fewer training steps on `Qwen2.5-Math-1.5B`, with consistent gains across model scales and RL algorithms. Experiments show that SAE transfers effectively across model families and scales, serving as a lightweight and reusable data engineering tool. These results demonstrate that model internals are a powerful and practical source of signals for post-training data engineering.

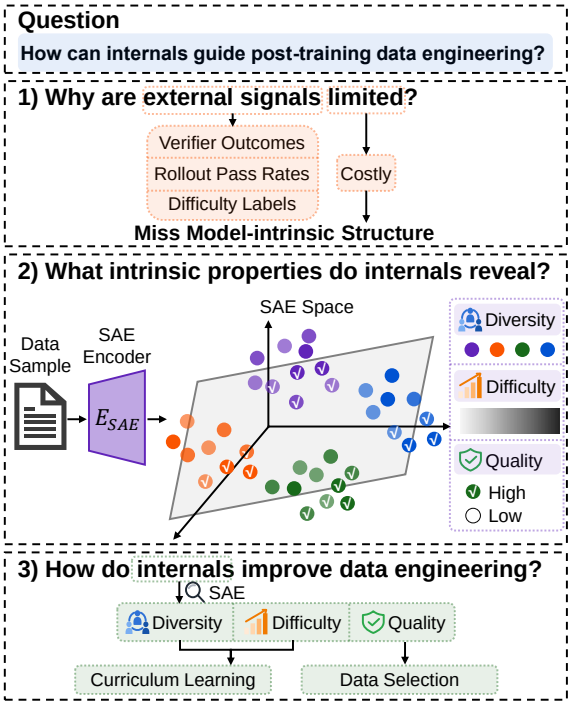

Figure 1: Conceptual overview of SAERL. Sparse Autoencoder (SAE) activations characterize three intrinsic data properties (diversity, difficulty, and quality), enabling SAE-based curriculum learning and data selection for LLM post-training.

1 Introduction

Post-training, especially reinforcement learning, has become central to advancing the capabilities of large language models (OpenAI, 2026; Anthropic, 2026; Zeng et al., 2026; DeepSeek-AI, 2026). Its effectiveness depends heavily on data engineering: which samples are used, how to sort the samples, and batching strategies. These choices shape the training signal at every step, making data engineering an important factor for improving both training efficiency and final performance.

Existing post-training data engineering pipelines typically rely on external feedback signals, including human preferences (Ouyang et al., 2022; Lambert et al., 2024), verifier outcomes (DeepSeek-AI et al., 2025; Shao et al., 2024; Yu et al., 2025), rollout pass rates (Sun et al., 2025; Xu et al., 2025; Zheng et al., 2025), and difficulty signals (Narvekar et al., 2020; Shi et al., 2025; Gao et al., 2025; Zhao et al., 2025). These signals have proven useful for data selection and curriculum learning.

However, external signals are often costly to obtain and to apply throughout training (Casper et al., 2023), leaving the rich data-feedback signals embedded in model internals largely underexplored. Recent work has shown that internal representations can guide data selection in pre-training (Sam et al., 2025; Rathi and Radford, 2026) and supervised fine-tuning (Ivison et al., 2025; Ma et al.,

2025; Chen et al., 2026; Yang et al., 2025b), suggesting that model internals encode structure actionable for training. Whether they can play a similar role in post-training data engineering for reinforcement learning remains an open question.

Mechanistic interpretability research (Meng et al., 2022; Wang et al., 2022; Somvanshi et al., 2026) continuously explores how to obtain and understand model internals. As a recent advance, Sparse Autoencoders (SAEs) decompose LLM hidden representations into sparse, fine-grained feature activations (Bricken et al., 2023; Gao et al., 2024; Templeton et al., 2024), providing fine-grained and disentangled perspectives of LLM internals. While recent pioneering work (Wang et al., 2025a) adopts LLM hidden representations in RL data selection, exploring the fine-grained feature space offered by SAE may lead to more holistic and precise modeling of data properties with model internals.

Therefore, we study the method using SAE activations to capture three intrinsic properties of post-training data: (1) *Diversity*: distances and clusters in the internal space can measure how broadly a batch covers distinct feature regions and reasoning patterns. (2) *Difficulty*: sparse activation patterns can reflect the actual demands that a problem imposes on the model, going beyond shallow features such as length or topic. (3) *Quality*: internal activations can help distinguish samples from the target distribution from noisy or off-distribution raw data. These three properties correspond to concrete data engineering operations: batching strategy, curriculum ordering, and data filtering.

Based on these findings, we propose SAERL, an intrinsic framework for RL post-training data engineering based on SAE activations. SAERL uses SAE to model three data properties: quality, difficulty, and diversity. SAERL then proceeds in three steps: (1) an SAE-based quality probe filters the data pool toward target-distribution samples; (2) samples are clustered in SAE space and sorted by calibrated difficulty within each cluster, forming local easy-to-hard trajectories; (3) batches are interleaved across clusters and moderately mixed by swapping a small tail portion between nearby batches, improving coverage while preserving within-batch coherence.

Experiments on mathematical reasoning show that SAERL improves performance and efficiency across model scales and RL algorithms. Ablation studies show that batching strategy, curriculum ordering, and data filtering each contribute to the final results. These results suggest that SAERL improves post-training data engineering by jointly modeling data diversity, sample difficulty, and data quality with SAEs.

Our contributions are twofold: (1) We frame model internals as actionable signals for post-training data engineering. (2) We propose SAERL, which grounds SAE-based quality, difficulty, and diversity signals in concrete data engineering operations for efficient LLM post-training. We hope that this work can facilitate future research on intrinsic data engineering and actionable mechanistic interpretability (Orgad et al., 2026).

2 Motivating Finding

We conduct a preliminary study to examine whether SAE activations encode actionable signals for post-training data engineering. We find that they capture three intrinsic data properties—diversity, difficulty, and quality—motivating the design of SAERL.

2.1 SAE Can Predict Data Diversity

SAE representations encode diversity-relevant semantic information. Since data diversity corresponds to coverage over distinct topics and skills, we examine whether SAE activations capture such semantic variation by testing their ability to predict external topic labels.

We use DEEPMATH (He et al., 2025), a large-scale mathematical reasoning dataset with annotated topic labels, in our pilot study. Given an SAE representation z_i for a data sample, we train a linear probe to predict topic labels at three levels of granularity:

$$\hat{t}_i = f_T(z_i). \tag{1}$$

As shown in Table 1, SAE features substantially outperform the majority-class baseline across all granularities, including 82 leaf topics. This indicates that SAE activations encode topic-level semantic structure, making SAE space a reliable basis for measuring data coverage and diversity in post-training data engineering.

2.2 SAE Can Predict Data Difficulty

SAE representations encode difficulty-relevant information. Data difficulty is reflected in internal activation patterns—problem meanings, symbolic structure, and required skills—making SAE activations a natural interface for extracting difficulty signals. Given the SAE representation z_i, we train

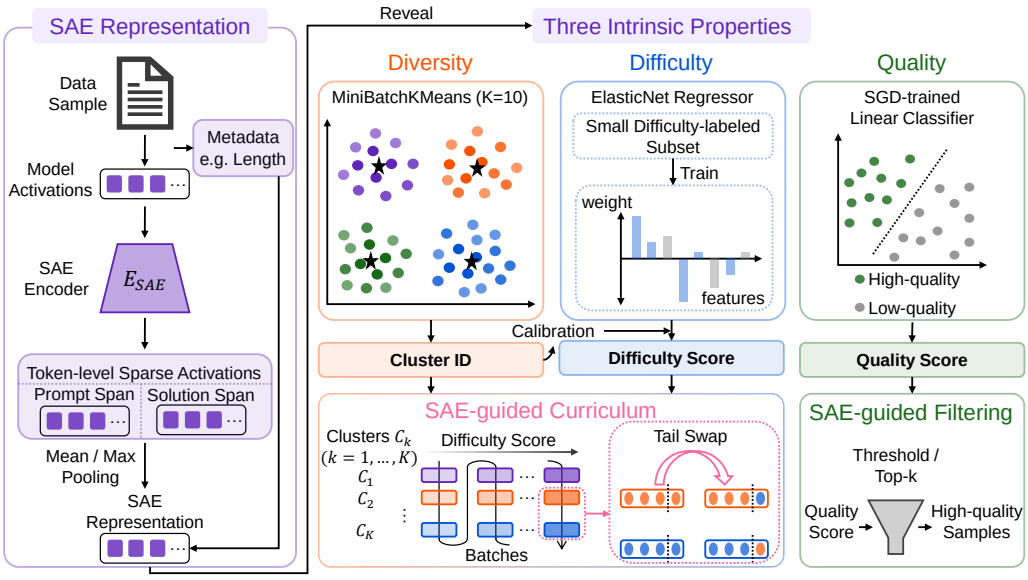

Figure 2: Overview of SAERL. Token-level SAE activations are pooled into a shared representation encoding diversity, difficulty, and quality. These three properties ground two data engineering operations: curriculum construction and data selection.

Target	Labels	Majority	SAE
L2 topic	9	31.8	54.6
L3 topic	36	17.2	37.7
Leaf topic	82	7.5	26.6

Table 1: Linear probe accuracy (%) predicting DEEPMATH topic labels from prompt-side SAE activations. Targets are external dataset metadata, providing a non-circular test of whether SAE representations encode semantic axes relevant to sample diversity.

Regime	Train	Test	ρ
In-domain	DM_{3k}	DM_{rem}	0.749
OOD	DM_{3k}	DSR_{10k}	0.135
Adapted OOD	$\text{DM}_{3k} + \text{DSR}_{800}$	DSR_{10k}	0.286

Table 2: Difficulty prediction from SAE activations using ElasticNet. ρ denotes Spearman correlation; DM and DSR denote DeepMath and DeepScaleR.

an ElasticNet (Zou and Hastie, 2005) regressor to predict a continuous difficulty score:

$$\hat{d}_i = f_D(z_i). \tag{2}$$

As shown in Table 2, SAE features strongly predict in-distribution difficulty and retain a positive signal under distribution shift, indicating that SAE activations capture difficulty-relevant structure beyond shallow cues such as length or topic. This makes them a reliable basis for difficulty-aware curriculum construction.

2.3 SAE Can Predict Data Quality

SAE representations encode quality-relevant information. Data quality reflects whether a training example is reliable, well-formed, and aligned with the target reasoning distribution. These properties are only partially captured by surface statistics such as length, step count, or TeX ratio.

We use PRM800K (Lightman et al., 2024) as the validation setting, as its step-level process labels provide a reliable proxy for solution quality. We convert these labels into numeric scores ($+1 \to 1$, $0 \to 0.5$, $-1 \to 0$) and average them within each example to obtain a continuous sample-level quality score. Given the SAE representation z_i, we train a ridge regressor to predict this score:

$$\hat{q}_i = f_Q(z_i). \tag{3}$$

As shown in Table 3, SAE features outperform both the mean baseline and a metadata-only baseline, improving test Pearson correlation from 0.2100 to 0.3715 over metadata features. This suggests that SAE activations capture quality-relevant structure beyond shallow cues, supporting their use for quality-aware data filtering.

Feature	RMSE ↓	MAE ↓	Pearson ↑
Mean	0.2161	0.1772	–
Metadata	0.2113	0.1718	0.2100
SAE	**0.2007**	**0.1608**	**0.3715**

Table 3: Quality prediction on PRM800K. SAE features outperform metadata features, indicating quality-relevant signal beyond surface statistics.

3 Methodology

Based on the motivating findings above, we propose SAERL, an offline data engineering framework for reinforcement learning post-training that uses SAE to model three intrinsic data properties—diversity, difficulty, and quality—and maps them to concrete operations: batching strategy, curriculum ordering, and data filtering.

3.1 SAE Representation

SAEs decompose dense model activations into sparse, interpretable feature activations (Gao et al., 2024), providing a structured interface for extracting content-level signals from model internals. Given a sample x_i, we extract token-level SAE activations separately from its prompt and solution spans, aggregating each via mean and max pooling to capture both sustained and localized activation patterns. The unified representation is

$$\phi(x_i) = \big[z_i,\, m_i\big], \tag{4}$$

where z_i concatenates the pooled SAE activations over both spans, and m_i is a small set of shallow metadata features (e.g., length statistics, TeX ratio, digit ratio); the SAE part contains 960 features and m_i contains 26.

3.2 Diversity-driven Batching Strategy

We model batch diversity by clustering samples in SAE space and applying moderate batch mixing. Empirically, we find that batch diversity in SAE space has a concave relationship with downstream performance: moderate cross-cluster mixing improves over pure-cluster batches, while excessive mixing hurts optimization (Section 5.2). Appendix A provides a bias–variance perspective analysis on this finding.

Clustering. We cluster samples using SAE features and metadata via MiniBatchKMeans (Sculley, 2010) at K=10, capturing model-internal structure such as mathematical semantics, problem format, and skill patterns.

Moderate batch mixing. Each batch is paired with a partner batch drawn from a nearby curriculum stage, matched by similar average difficulty and sequence length but required to have a different dominant cluster, with a small tail portion exchanged between the two batches.

3.3 Difficulty-driven Curriculum Ordering

We model sample difficulty from SAE representations and use it to construct a cluster-first easy-to-hard curriculum.

Difficulty proxy and calibration. As described in Section 2.2, we train a lightweight Elastic-Net regressor on a small difficulty-labeled subset ($|L|$=3k) to estimate sample difficulty, producing a raw score $\hat{d}_i = f_D(\phi(x_i))$ for each sample.

Since scores may vary in scale across clusters, we apply cluster-wise calibration using a global mapping with shrinkage-based cluster corrections:

$$r_i = \text{Calibrate}\big(\hat{d}_i,\, c_i\big), \tag{5}$$

where c_i is the cluster assignment of x_i and r_i is the final ranking score used for curriculum ordering.

Cluster-first curriculum. Within each cluster, samples are sorted by r_i into fixed-size batches, forming local easy-to-hard trajectories. The global curriculum then interleaves batches across clusters stage by stage, with moderate batch mixing applied within each stage.

3.4 Quality-driven Data Filtering

We model sample quality from SAE representations to filter noisy data before curriculum ordering. The probe formalizes this as binary classification: given a sample x_i, it outputs the probability of belonging to the target distribution,

$$s_i = p_\psi\big(y_i = 1 \mid \phi(x_i)\big), \tag{6}$$

implemented as a SGD-trained linear classifier (Bottou, 2010) over SAE activations, trained on a subset of source-labeled samples. High-scoring samples are then selected by a fixed threshold $\mathcal{D}_\tau = \{x_i : s_i \geq \tau\}$ or top-k ranking $\mathcal{D}_k = \text{TopK}_{x_i}(s_i)$, filtering the noisy data pool toward the target distribution and providing a higher-quality data source for post-training.

4 Main Experiment

We evaluate SAERL in the mathematical reasoning domain, focusing on downstream performance, training efficiency, and noisy-data selection.

4.1 Experiment Setup

Models and training. We train two model scales, `Qwen2.5-Math-1.5B` and `Qwen2.5-Math-7B` (Yang et al., 2024), on `DeepMath-103K` (He et al., 2025) with a batch size of 128 to test the generality of SAERL. We denote SAERL trained with GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025) as **SAERL^{G}** and **SAERL^{D}**, respectively. We train an SAE on layer-27 activations of `Qwen3-1.7B` (Yang et al., 2025a) as the shared encoder for all data engineering operations, demonstrating that a single SAE trained on one model can effectively guide post-training data engineering for other model families and larger scales. Additional details are provided in Appendix C.2.

Evaluation. We instantiate SAERL in the mathematical reasoning domain and evaluate on six benchmarks spanning a wide difficulty range: `GSM8K` (Cobbe et al., 2021) and `AMC23` (lower), `MATH500` (Lightman et al., 2024) and `MinervaMath` (Lewkowycz et al., 2022) (mid), and `OlympiadBench` (He et al., 2024) and `AIME24` (competition-level), which are referred to as `GSM8K`, `AMC`, `MATH`, `MNV`, `OLPD`, and `AIME`, respectively. We report `Pass@8` for `AIME24` and `Avg@8` for the remaining five benchmarks.

Baselines. We compare SAERL against five baselines. Vanilla **GRPO** (Shao et al., 2024) and **DAPO** (Yu et al., 2025) serve as RL algorithm baselines without curriculum, and we pair SAERL with both to test whether its benefits are consistent across RL algorithms. **Difficulty Curriculum Learning** (Narvekar et al., 2020) uses externally provided difficulty labels, testing whether SAE-based signals add value beyond human annotations. **ADARFT** (Shi et al., 2025) estimates difficulty from rollout accuracy, representing rollout-based curriculum methods. **GAINRL** (Wang et al., 2025a) selects data via compressed hidden-state representations, serving as the closest internal-signal baseline to directly test whether sparse SAE features outperform dense alternatives.

4.2 Training Performance

Table 4 shows that SAERL improves average accuracy across RL algorithms, baselines, and model scales. At the 1.5B scale, SAERL improves both GRPO and DAPO, showing that the SAE-based curriculum is not specific to a particular RL algorithm. Compared with Difficulty Curriculum Learning, ADARFT, and GAINRL, SAERL obtains stronger overall performance, indicating that sparse SAE activations provide a more useful signal than external difficulty labels, rollout accuracy, or compressed hidden states. At the 7B scale, SAERL^{G} again achieves the best average result among the compared methods, suggesting that a shared SAE trained on a smaller model can still guide data engineering for larger models.

Model	Method	AIME	AMC	GSM8K	MATH	MNV	OLPD	Avg
Qwen2.5 Math-1.5B	GRPO	30.0	53.4	81.1	71.4	26.0	34.8	49.4
	DAPO	**40.0**	55.6	81.9	70.7	26.6	34.4	51.5
	DIFF	33.3	55.0	81.6	**72.1**	26.5	34.7	50.5
	ADARFT	**40.0**	55.6	78.9	69.4	23.6	32.4	49.9
	GAINRL	33.3	53.1	79.2	70.8	25.5	34.9	49.4
	SAERL^{G}	**40.0**	**56.2**	83.5	72.0	27.3	**35.7**	52.4
	SAERL^{D}	**40.0**	55.9	**84.6**	72.0	**28.4**	34.6	**52.5**
Qwen2.5 Math-7B	GRPO	46.6	68.1	90.3	79.1	33.6	42.0	59.9
	DIFF	50.0	**68.7**	90.9	79.1	32.9	42.6	60.7
	ADARFT	**53.3**	63.1	87.5	76.1	31.8	38.6	58.4
	GAINRL	**53.3**	68.4	90.1	79.8	34.6	41.7	61.3
	SAERL^{G}	**53.3**	68.4	**91.5**	**80.3**	**35.4**	**43.0**	**61.9**

Table 4: Accuracy (%) at step 900. SAERL^{G} and SAERL^{D} denote SAERL trained with GRPO and DAPO, respectively. DIFF denotes Difficulty Curriculum Learning. **Bold** indicates the best result, while underline denotes the second best.

Model	Method	AIME	AMC	GSM8K	MATH	MNV	OLPD	Avg
Qwen2.5 Math-1.5B	GRPO	40	680	540	560	560	440	470
	DAPO	**20**	580	320	400	260	400	330
	DIFF	60	340	440	440	540	420	373
	ADARFT	40	540	900	820	860	900	676
	GAINRL	40	760	780	480	600	480	523
	SAERL^{G}	**20**	580	400	380	520	380	380
	SAERL^{D}	**20**	**100**	**240**	**340**	**220**	**320**	**206**
Qwen2.5 Math-7B	GRPO	40	320	240	180	**220**	200	200
	DIFF	**20**	140	**160**	**160**	420	**180**	180
	ADARFT	**20**	740	360	400	900	440	476
	GAINRL	80	240	220	180	**220**	220	193
	SAERL^{G}	40	**120**	200	200	280	200	**173**

Table 5: Training steps required to reach the target accuracy on each benchmark. For each model–benchmark pair, the target accuracy is set to the minimum final accuracy among all compared methods in Table 4, ensuring that every method can reach it by step 900. Lower values indicate higher training efficiency. Method notation follows Table 4. **Bold** indicates the fewest steps, while underline denotes the second fewest.

4.3 Training Efficiency

SAERL improves training efficiency by reducing both training steps and preparation cost.

Table 5 evaluates convergence speed by measuring how many training steps each method needs to reach a shared target accuracy. At the 1.5B

Rule	Kept	DM	Purity (%)	Recall (%)
FULL	210,043	103,022	49.05	100.00
P95-T	103,121	98,342	95.37	95.46
P99-T	87,664	86,767	98.98	84.22
TOP-50K	50,000	49,962	**99.92**	48.50
TOP-90K	90,000	88,855	98.73	86.25

Table 6: SAE-probe-based DeepMath-like sample selection from the mixed raw pool. DM denotes DeepMath samples; P95-T and P99-T denote percentile-threshold selection rules.

Method	AIME	AMC	GSM8K	MATH	MNV	OLPD	Avg
SAERL	**40.0**	**56.2**	**83.5**	**72.0**	**27.3**	**35.7**	**52.4**
– Diff	33.3	52.1	81.6	71.3	25.0	35.0	49.7
– Diff & Mix	33.3	55.3	81.2	71.1	24.8	35.0	50.1
– Clus & Mix	36.6	55.0	82.2	71.3	25.3	34.5	50.8

Table 7: Ablation results at step 900 on `Qwen2.5-Math-1.5B`, reported in accuracy (%). The first row denotes the full SAERL. Rows prefixed with "–" remove the corresponding component(s), where Diff, Mix, and Clus denote difficulty sorting, moderate batch mixing, and cluster-first grouping, respectively. **Bold** indicates the best result.

scale, SAERL accelerates both GRPO and DAPO. SAERL^{D} gives the fastest average convergence, and SAERL^{G} requires fewer average steps than GRPO, ADARFT, and GAINRL. At the 7B scale, SAERL^{G} reaches the target in the fewest average steps. These results show that SAE-guided data engineering improves convergence across different model scales and RL algorithms.

SAERL also demonstrates efficiency gains. The Difficulty baseline and ADARFT achieve comparable convergence speed but require LLM-generated labels or multiple rollouts per problem at substantial cost—ADARFT takes approximately 17.33 H100 GPU hours with a reduced rollout budget (Appendix C.3). In contrast, SAERL trains the difficulty proxy from a small labeled subset of 3,000 samples, and SAE encoding for the full dataset of 103,022 samples takes about 0.5 H100 GPU hours. Thus, SAERL obtains its convergence gains with substantially lower preprocessing overhead.

4.4 Noisy Data Selection

We further evaluate whether SAE activations support the selection of high-quality samples from a target distribution within a larger mixed noisy pool. We use DEEPMATH as the target distribution: it is constructed from NUMINAMATH (Li et al., 2024b) and other open mathematical sources through decontamination, difficulty filtering, and answer-verifiability filtering (He et al., 2025), making recovery from its source family a meaningful test of quality discrimination. We formulate the task as follows. The raw pool $\mathcal{D}_{\text{raw}}$ consists of 103,022 DEEPMATH samples mixed with 107,021 samples from the source corpus NUMINAMATH-1.5, giving $|\mathcal{D}_{\text{raw}}| = 210{,}043$. The probe is trained to recover the DEEPMATH subset using only $d = 960$ SAE features obtained by mean/max pooling over prompt and solution tokens.

The SAE-only source/style probe achieves 0.9911 ROC-AUC and 0.9910 AP on the holdout split, indicating that DEEPMATH-like high-quality samples are highly separable in the SAE activation space. As shown in Table 6, after applying the fixed probe to $\mathcal{D}_{\text{raw}}$, the p_{95} threshold retains 103,121 samples, with 95.37% DEEPMATH purity and 95.46% recall. Direct top-50k selection by the probe score further improves the DEEPMATH purity to 99.92%. These results suggest that the SAE-based probe captures fine-grained DEEPMATH-like activation signatures, enabling stable high-quality data selection from noisy data.

5 Analysis

We analyze the sources of SAERL's gains across four dimensions: component contribution, batch diversity control, robustness, and interpretability.

5.1 Ablation Study

SAERL relies on the joint effect of batching strategy, curriculum ordering, data filtering. Difficulty sorting defines the easy-to-hard trajectory, cluster-first grouping preserves local coherence in SAE activation space, and moderate batch mixing adds limited cross-cluster coverage without disrupting the trajectory.

Table 7 shows that removing difficulty sorting causes the largest degradation, confirming that the easy-to-hard trajectory is central to SAERL. The w/o Clus & Mix variant removes cluster assignments and therefore cannot perform moderate batch mixing, leaving a difficulty-only curriculum. Its drop indicates that difficulty sorting alone is insufficient, and SAE-space grouping provides useful local coherence.

Comparing w/o Diff with w/o Diff & Mix shows that mixing without difficulty sorting does not improve the curriculum and can even weaken it. In contrast, the full SAERL outperforms the variants that remove either difficulty sorting or

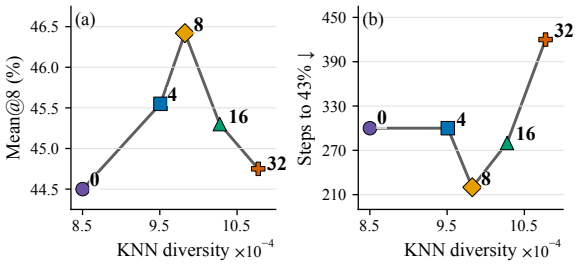

Figure 3: SAE-space batch diversity versus downstream reinforcement learning performance. **(a)** Average `mean@8` at step 800 as a function of the mean in-batch k-NN distance in SAE space, with $k = 5$. **(b)** Number of training steps required to reach the fixed average `mean@8` threshold $\tau = 43.0\%$. Moderate diversity, represented by `mix8`, achieves the best step-800 performance and the fastest threshold crossing.

Metric	B	Method	AIME	AMC	GSM8K	MATH	MNV	OLPD
Avg @8	128	GRPO	13.7	53.4	81.1	71.4	26.0	34.8
		SAERL^{G}	**14.1**	**56.2**	**83.5**	**72.0**	**27.3**	**35.7**
	512	GRPO	12.5	51.8	80.3	69.9	24.5	33.8
		SAERL^{G}	**13.7**	**56.5**	**82.1**	**71.1**	**25.4**	**34.9**
Pass @8	128	GRPO	30.0	85.0	94.2	86.2	**43.3**	52.7
		SAERL^{G}	**40.0**	**87.5**	**94.6**	**86.8**	43.0	**54.8**
	512	GRPO	**36.6**	**85.0**	93.4	84.6	40.4	52.4
		SAERL^{G}	33.3	**85.0**	**93.7**	**85.8**	**41.1**	**53.3**

Table 8: Accuracy (%) across batch sizes and evaluation metrics. Results are evaluated on `Qwen2.5-Math-1.5B` at step 900 for $B = 128$ and step 300 for $B = 512$, where B denotes the training batch size. SAERL^{G} denotes SAERL trained with GRPO. **Bold** indicates the best result.

cluster-based batch construction, indicating that moderate batch mixing is most effective when it is applied on top of an already structured cluster-first, easy-to-hard curriculum.

5.2 Batch Diversity Analysis

The cluster-first curriculum introduces moderate cross-cluster batch mixing to balance within-batch gradient coherence and cross-cluster coverage. The mixing strength, controlled by the number of tail samples swapped between batches, directly governs this trade-off. To verify that moderate mixing is indeed optimal and to characterize how sensitivity to mixing strength affects downstream performance, we compare five curriculum variants that differ only in this parameter:

$$\mathcal{M} = \{\texttt{mix0}, \texttt{mix4}, \texttt{mix8}, \texttt{mix16}, \texttt{mix32}\},$$

where `mix0` is the cluster-first curriculum with no mixing, and larger indices correspond to stronger cross-cluster mixing. All other components of SAERL are held fixed.

We quantify batch diversity by the mean in-batch k-NN distance (k=5) computed in the two-dimensional SAE projection space, and measure downstream performance by the average `mean@8` across the six evaluation benchmarks used in the main experiments. Figure 3 reports both peak performance at step 800 and the number of steps required to reach a fixed threshold $\tau = 43.0\%$.

The results reveal a clear non-monotonic relationship. Performance improves steadily from `mix0` to `mix8`, and `mix8` reaches τ in the fewest training steps. Beyond this point, further increasing the mixing strength to `mix16` and `mix32` degrades both final accuracy and convergence speed—despite `mix32` achieving the highest measured diversity. This suggests that beyond a moderate level, cross-cluster mixing disrupts within-batch gradient coherence more than it reduces cluster-local bias.

This pattern is consistent with the bias–variance decomposition in Appendix A, which shows that the mixing utility is a concave function of mixing strength with a unique interior optimum. The practical takeaway is that effective batch construction requires balancing two competing objectives: preserving local SAE-space coherence to stabilize optimization, while introducing limited cross-cluster coverage to reduce directional bias.

5.3 Batch Size Analysis

Table 8 shows that SAERL remains effective across batch sizes. Under `Avg@8`, SAERL outperforms GRPO at both $B = 128$ and $B = 512$, indicating that the curriculum remains effective beyond the default training batch size. Under `Pass@8`, increasing the batch size narrows the gap between the two methods. This suggests that larger batches may dilute the structural benefit of an ordered learning trajectory.

5.4 Interpretability Analysis

Beyond downstream performance, we examine whether SAERL exposes interpretable structure at the cluster and feature levels during curriculum construction. Additional details are provided in Appendix D.

Cluster-level structure. Comparing $\mathcal{C}$ with DEEPMATH topic metadata $\mathcal{T}$ yields low alignment (purity $= 0.1095$, NMI $= 0.0881$), indicating that SAE clusters do not reproduce the human-

defined topic taxonomy. Rather, inspection of cluster statistics and representative examples reveals that clusters capture curriculum-relevant properties including problem format, reasoning structure, solution profile, and difficulty. Several clusters also show enrichment for recognizable mathematical areas such as limits, combinatorics, group theory, and integration. SAE clusters thus characterize the data along axes more relevant to curriculum construction than external topic labels.

Feature-level signals. The difficulty proxy relies primarily on SAE activations rather than shallow metadata. Among the top-20 features ranked by LightGBM (Ke et al., 2017) gain, 19 are SAE-derived and only 1 is a metadata feature; among the top-100, only 3 are metadata features. Within the SAE features, solution-side mean activations dominate, consistent with sustained solution-side patterns providing the strongest correlational signal for difficulty. Prompt-side max activations also contribute, reflecting localized cues in the problem statement such as symbolic structure or diagrammatic format. A high-activation audit in Appendix D further shows that individual high-gain features exhibit recurring semantic tendencies spanning abstract algebra, advanced analysis, geometry, combinatorics, and number-theoretic reasoning.

Taken together, these analyses show that SAERL provides not only an effective curriculum ordering, but also an auditable decision pathway. Each sample can be inspected through its activation group, difficulty-related feature signals, and position within the curriculum.

6 Related Work

We discuss two trends behind our work: post-training data engineering is becoming more adaptive, and model internals are increasingly used as training signals.

6.1 Post-training Data Engineering

Post-training data engineering controls which examples are used, when they appear, and how training budget is allocated, which strongly shapes final model behavior (Ouyang et al., 2022; Zhou et al., 2023; Lambert et al., 2024; DeepSeek-AI et al., 2025). Existing methods have evolved from (1) task-, goal-, or replay-based curricula (Narvekar et al., 2020; Li et al., 2024a; Tzannetos et al., 2024), to (2) quality- and capability-aware selection of compact high-value data or boundary-level prompts (Ye et al., 2025; Li et al., 2025b; Chen et al., 2025; Shi et al., 2025; Sun et al., 2025; Zhao et al., 2025; Gao et al., 2025), and then to (3) optimization- and resource-aware selection using gradients, influence, rollout utility, or distribution schedules (Li et al., 2025a; Yang et al., 2026; Zhu et al., 2025; Wang et al., 2025a; Xu et al., 2025; Zheng et al., 2025; Wang et al., 2025b; Rajaraman et al., 2026).

However, they still rely mainly on external or scalar signals; SAERL instead grounds data engineering in model-internal structure.

6.2 Model Internals for LLM Training

Model internals have moved from post-hoc analysis toward training-time feedback. Existing work uses (1) logit- or loss-based signals for data filtering and token or instruction selection (Li et al., 2024c; Lin et al., 2024), (2) gradients or influence estimates for data selection and weighting (Xia et al., 2024; Li et al., 2025a; Yang et al., 2026; Zhu et al., 2025), and (3) hidden states or activations for efficient example selection or representation-level intervention (Wang et al., 2025a; Wu et al., 2024).

These approaches show that internals can guide training, but they often reduce internal structure to coarse signals. SAERL instead uses sparse autoencoder features (Bricken et al., 2023; Templeton et al., 2024; Gao et al., 2024), which provide sparse and fine-grained activation signals for data engineering, extending beyond prior SAE-based uses for tuning-data diversity (Yang et al., 2025b) and preference modeling (Liu et al., 2025) to the RLVR post-training setting.

7 Conclusion

We propose SAERL, a post-training data engineering framework grounded in model-internal sparse representations. SAERL uses SAE activations as a shared representation space to model three intrinsic data properties—diversity, difficulty, and quality—and grounds each in a concrete data engineering operation: batching strategy, curriculum ordering, and data filtering. Experiments on mathematical reasoning demonstrate consistent gains in accuracy and convergence efficiency across model scales and RL algorithms, with a single SAE transferring effectively across model families. These results suggest that model internals are a powerful and practical source of signals for post-training data engineering, opening a direction complementary to external feedback-based approaches.

Limitations

Domain scope. Our empirical validation focuses on mathematical reasoning with verifiable rewards. This setting provides a controlled testbed for studying curriculum construction, since both training feedback and evaluation outcomes can be measured reliably. However, the extent to which the same SAE-space structure transfers to other post-training settings remains to be established, including code-centric RL, agentic RL, tool-use and multi-step decision-making, and general instruction-following.

Limited supervision. Although SAERL reduces the need for large-scale labeling or rollout-based scoring, it is not fully unsupervised. The difficulty proxy uses a small difficulty-labeled subset, and the quality probe relies on source or distribution labels as supervision. Future work may explore weaker forms of supervision, self-calibrated scoring, or fully unsupervised criteria for constructing SAE-guided curricula.

Theoretical scope. Our analysis treats proximity in SAE space as a proxy for semantic similarity and gradient coherence. This yields an optimization-level interpretation of the observed coherence–coverage trade-off, but it does not prove a causal relationship between SAE distance and training dynamics. Establishing such a guarantee would require direct gradient-level measurements on the representation space.

Ethical Considerations

This section discusses the ethical considerations and broader impact of this work.

Potential Risks. SAERL uses model-internal SAE representations for post-training data engineering. Although intended to improve efficiency and inspectability, such signals could be misused to optimize data for unsafe behaviors. We restrict our experiments to mathematical reasoning with verifiable rewards and recommend safety filtering and human oversight for broader applications.

Intellectual Property. The models, datasets, benchmarks, and software frameworks used in this work are publicly available research artifacts and are used in accordance with their respective licenses and terms of use. Any released code or processed artifacts will follow the corresponding license requirements.

Intended Use. SAERL is intended as a research framework for post-training data engineering, including curriculum construction, batch organization, data filtering, and interpretability-oriented analysis. It is not intended for developing harmful models, evading safety mechanisms, or optimizing data for malicious capabilities.

Documentation of Artifacts. We will document the released artifacts with sufficient detail to support reproducibility, including the data-processing pipeline, SAE feature extraction, curriculum construction, data filtering, and evaluation setup.

AI Assistants in Research or Writing. We use AI assistants for code development assistance and language polishing. All AI-assisted content is reviewed and edited by the authors, who remain responsible for the final scientific claims, experiments, and writing.

References

Anthropic. 2026. Claude Opus 4.6. https://www.anthropic.com/transparency. Model transparency report, released February 2026.

Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT 2010*, pages 177–186. Physica-Verlag HD.

Trenton Bricken et al. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Biyik, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *Preprint*, arXiv:2307.15217.

Xin Chen, Junchao Wu, Shu Yang, Runzhe Zhan, Zeyu Wu, Min Yang, Shujian Huang, Lidia S. Chao, and Derek F. Wong. 2026. Neuron-aware data selection in instruction tuning for large language models. *Preprint*, arXiv:2603.13201.

Xinjie Chen, Minpeng Liao, Guoxin Chen, Chengxi Li, Biao Fu, Kai Fan, and Xinggao Liu. 2025. From data-centric to sample-centric: Enhancing LLM

reasoning via progressive optimization. *Preprint*, arXiv:2507.06573.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence. https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf. Technical report.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, et al. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Preprint*, arXiv:2501.12948.

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. *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.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850. Association for Computational Linguistics.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. DeepMath-103K: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *Preprint*, arXiv:2504.11456.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. 2025. Large-scale data selection for instruction tuning. *Preprint*, arXiv:2503.01807.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, pages 3146–3154.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. 2024. Tulu 3: Pushing frontiers in open language model post-training. *Preprint*, arXiv:2411.15124.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. *Preprint*, arXiv:2206.14858.

Hu Li, Xuezhong Qian, and Wei Song. 2024a. Prioritized experience replay based on dynamics priority. *Scientific Reports*, 14(6014).

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024b. NuminaMath 1.5. https://huggingface.co/datasets/AI-MO/NuminaMath-1.5.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024c. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14255–14273. Association for Computational Linguistics.

Shipeng Li, Zhiqin Yang, Shikun Li, Xiaobo Xia, Hengyu Liu, Xinghua Zhang, Gaode Chen, Dong Fang, Ying Tai, and Zhe Peng. 2025a. LearnAlign: Data selection for LLM reinforcement learning with improved gradient alignment. *Preprint*, arXiv:2506.11480.

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

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri 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*.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2024. Rho-1: Not all tokens are what you need. *Preprint*, arXiv:2404.07965.

Dengcan Liu, Jiahao Li, Zheren Fu, Yi Tu, Jiajun Li, Zhendong Mao, and Yongdong Zhang. 2025. SparseRM: A lightweight preference modeling with sparse autoencoder. *Preprint*, arXiv:2511.07896.

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.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Preprint*, arXiv:2202.05262.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. 2020. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50.

OpenAI. 2026. GPT-5.5 System Card. `https://deploymentsafety.openai.com/gpt-5-5`. Published April 23, 2026.

Hadas Orgad, Fazl Barez, Tal Haklay, Isabelle Lee, Marius Mosbach, Anja Reusch, Naomi Saphra, Byron Wallace, Sarah Wiegreffe, Eric Wong, et al. 2026. Interpretability can be actionable. *arXiv preprint arXiv:2605.11161*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849.

Nived Rajaraman, Audrey Huang, Miro Dudik, Robert Schapire, Dylan J. Foster, and Akshay Krishnamurthy. 2026. Learning to reason with curriculum I: Provable benefits of autocurriculum. *Preprint*, arXiv:2603.18325.

Neil Rathi and Alec Radford. 2026. Shaping capabilities with token-level data filtering. *Preprint*, arXiv:2601.21571.

Dylan Sam, Ayan Chakrabarti, Afshin Rostamizadeh, Srikumar Ramalingam, Gui Citovsky, and Sanjiv Kumar. 2025. Analyzing similarity metrics for data selection for language model pretraining. *Preprint*, arXiv:2502.02494.

D. Sculley. 2010. Web-scale k-means clustering. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1177–1178. Association for Computing Machinery.

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. *Preprint*, arXiv:2402.03300.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. 2025. Efficient reinforcement fine-tuning via adaptive curriculum learning. *Preprint*, arXiv:2504.05520.

Shriyank Somvanshi, Md Monzurul Islam, Amir Rafe, Anannya Ghosh Tusti, Arka Chakraborty, Anika Baitullah, Tausif Islam Chowdhury, Nawaf Alnawmasi, Anandi Dutta, and Subasish Das. 2026. Bridging the black box: a survey on mechanistic interpretability in ai. *ACM Computing Surveys*, 58(8):1–35.

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.

Adly Templeton et al. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. Transformer Circuits Thread.

Georgios Tzannetos, Parameswaran Kamalaruban, and Adish Singla. 2024. Proximal curriculum with task correlations for deep reinforcement learning. *Preprint*, arXiv:2405.02481.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. *Preprint*, arXiv:2211.00593.

Qinsi Wang, Jinghan Ke, Hancheng Ye, Yueqian Lin, Yuzhe Fu, Jianyi Zhang, Kurt Keutzer, Chenfeng Xu, and Yiran Chen. 2025a. Angles don't lie: Unlocking training-efficient RL through the model's own signals. *Preprint*, arXiv:2506.02281.

Zhenting Wang, Guofeng Cui, Yu-Jhe Li, Kun Wan, and Wentian Zhao. 2025b. DUMP: Automated distribution-level curriculum learning for RL-based LLM post-training. *Preprint*, arXiv:2504.09710.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. 2024. ReFT: Representation finetuning for language models. *Preprint*, arXiv:2404.03592.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 54104–54132. PMLR.

Yixuan Even Xu, Yash Savani, Fei Fang, and J. Zico Kolter. 2025. Not all rollouts are useful: Down-sampling rollouts in LLM reinforcement learning. *Preprint*, arXiv:2504.13818.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Ningyuan Yang, Weihua Du, Weiwei Sun, Sean Welleck, and Yiming Yang. 2026. GradAlign: Gradient-aligned data selection for LLM reinforcement learning. *Preprint*, arXiv:2602.21492.

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

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. LIMO: Less is more for reasoning. *Preprint*, arXiv:2502.03387.

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

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. 2026. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*.

Yang Zhao, Kai Xiong, Xiao Ding, et al. 2025. UFO-RL: Uncertainty-focused optimization for efficient reinforcement learning data selection. *Preprint*, arXiv:2505.12457.

Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. 2025. Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts. *Preprint*, arXiv:2506.02177.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. *Preprint*, arXiv:2305.11206.

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.

Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

A A Bias–Variance View of Moderate Batch Mixing

We provide a bias–variance perspective on why moderate cross-cluster batch mixing can improve optimization.

Setup. For each sample x_i, let $g_i = \nabla_\theta \ell(x_i; \theta)$ denote its per-sample gradient. Since SAE activations z_i approximate the model's internal representation of x_i, we assume $g_i = G(z_i) + \varepsilon_i$ for a locally Lipschitz $G\colon \mathbb{R}^d \to \mathbb{R}^p$, where ε_i captures SAE approximation error and residual nonlinear effects. Under this assumption, samples nearby in SAE space tend to produce similar gradients.

Pure-cluster bias. Consider a batch of size b drawn from cluster c, with gradient mean μ_c and covariance Σ_c, and let G_t denote the target gradient. The MSE of $\hat{G}_c = \frac{1}{b}\sum_{i=1}^{b} g_i$ decomposes as

$$\mathrm{MSE}_0 = \|G_t - \mu_c\|^2 + \frac{1}{b}\,\mathrm{tr}(\Sigma_c), \tag{7}$$

where the two terms are cluster-local bias and estimation variance, respectively. Pure-cluster batches have low variance but may be biased when μ_c deviates from G_t.

Effect of mixing. Suppose the mixed batch contains $\lfloor \rho b \rfloor$ samples from cluster d and $(1-\rho)b$ from cluster c, with the two clusters uncorrelated. Let $r_c = G_t - \mu_c$ and $v = \mu_d - \mu_c$. The net MSE change relative to equation 7 is

$$\Delta(\rho) = -2\rho\langle r_c, v\rangle + \rho^2\|v\|^2 + \frac{\rho}{b}\,\mathrm{tr}(\Sigma_d - \Sigma_c), \tag{8}$$

When $v \neq 0$, $\Delta(\rho)$ is a convex quadratic in ρ with minimizer $\rho^\dagger = \mathrm{clip}(\rho^*, 0, 1)$, where $\rho^* = A/2C$, $A = 2\langle r_c, v\rangle - \frac{1}{b}\,\mathrm{tr}(\Sigma_d - \Sigma_c)$, and $C = \|v\|^2$; when $v = 0$, the optimum is attained at an endpoint. If $\rho^* \in (0, 1)$, the mixing utility $U(\rho) = -\Delta(\rho)$ admits an interior maximum: too little mixing leaves cluster-local bias uncorrected, while too much weakens within-batch gradient coherence.

B Technical Details of the SAERL Pipeline

This appendix provides additional technical details for the offline data-processing steps used in Section 3.

B.1 SAE Sample Representation

Each sample x_i is divided into a prompt span S_i^p and a solution spanS_i^s. Let $\mathcal{F}$ denote the retained SAE feature set, and let $a_t \in \mathbb{R}^{|\mathcal{F}|}$ be the SAE activation vector at token t. For each span, we summarize token-level SAE activations using mean pooling and max pooling. The SAE representation z_i is obtained by concatenating the mean-pooled and max-pooled activations from both the prompt and solution spans:

$$z_i = \left[\bar{a}_{S_i^p}; a^{\max}_{S_i^p}; \bar{a}_{S_i^s}; a^{\max}_{S_i^s}\right]. \tag{9}$$

For clustering and difficulty estimation, we further append shallow metadata features m_i, including length statistics, TeX ratio, and digit ratio, and use $\phi_i = [z_i; m_i]$ as the full feature vector. In our experiments, z_i is 960-dimensional and m_i is 26-dimensional.

B.2 Clustering and Difficulty Calibration

We cluster samples in the feature space using MiniBatchKMeans with $K = 10$. Each sample is assigned to the nearest cluster centroid based on its full feature vector ϕ_i.

To estimate sample difficulty, we use a small difficulty-labeled subset $\mathcal{L}$ with $|\mathcal{L}| = 3000$. For each labeled sample, the difficulty label is denoted by $d_i^\star$. We train an ElasticNet difficulty proxy f_D on the labeled subset and use it to produce the raw difficulty prediction $\hat{d}_i = f_D(\phi_i)$.

Because only a limited number of difficulty labels are available, the raw proxy score is calibrated using a global calibration map fitted on the small labeled subset, together with a shrinkage-based cluster residual. Let g denote the global calibration map fitted on $\mathcal{L}$.

For each cluster, we compute the average residual between the labeled difficulty $d_i^\star$ and the globally calibrated prediction $g(\hat{d}_i)$ over labeled samples in that cluster. If a cluster has no labeled samples, its residual is set to zero. The residual is then scaled by a shrinkage weight $\lambda_c = n_c/(n_c + \tau_{\mathrm{sh}})$, where n_c is the number of labeled samples in cluster c and $\tau_{\mathrm{sh}} > 0$ controls the strength of shrinkage toward the global calibration.

The final difficulty score is

$$r_i = g(\hat{d}_i) + \lambda_{c_i}\Delta_{c_i}. \tag{10}$$

A larger r_i corresponds to a higher estimated difficulty.

B.3 Curriculum Ordering and Moderate Batch Mixing

Within each cluster, samples are sorted by the calibrated difficulty score r_i in ascending order, re-

sulting in a local easy-to-hard curriculum. The ordered samples in each cluster are then partitioned into fixed-size batches. The global curriculum interleaves these batches across clusters stage by stage, which preserves local easy-to-hard trajectories while maintaining coverage across different clusters.

After the cluster-first curriculum has been constructed, we apply moderate batch mixing. For an ordered batch of size b, we keep the first $b - h$ samples fixed and exchange only the last h tail samples with another batch. The partner batch is selected from a local curriculum window and must satisfy three conditions: it should have similar average calibrated difficulty, similar average sequence length, and a different dominant cluster. The dominant cluster of a batch is defined as the most frequent cluster label among its samples.

Since only the tail block is exchanged, this operation introduces limited cross-cluster mixing while largely preserving the local curriculum structure within each batch.

B.4 SAE-based Data Selection

Raw data selection is formulated as a binary classification problem over SAE representations. Let $\mathcal{D}_{\text{raw}}$ denote the raw candidate pool. For each candidate sample, $y_i = 1$ indicates membership in the target DeepMath-like distribution, while $y_i = 0$ indicates otherwise.

The quality probe uses only the SAE representation z_i, without metadata. We train a linear classifier with SGD and use its predicted probability as the selection score:

$$s_i = p_\psi(y_i = 1 \mid z_i) = \sigma(w^\top z_i + b), \qquad (11)$$

where $\psi = (w, b)$ and σ is the logistic sigmoid.

Given the score s_i, we use either threshold-based selection or fixed-size top-k selection. The threshold rule selects samples with scores above a quality threshold γ, which controls selection precision. The top-k rule selects the k highest-scoring samples, which fixes the number of selected samples.

C Implementation Detail

This appendix provides the training hyperparameters and baseline implementation details used in the experiments.

C.1 Sparse Autoencoder Training Details

We train the SAE using the OpenSAE framework on layer-27 activations of `Qwen3-1.7B`, with an expansion factor of 64 to support fine-grained feature extraction. The training corpus consists of `FineWeb-Edu` (Penedo et al., 2024) and `Wikipedia`, totaling 80GB. Training was conducted on 4 A100 GPUs and completed in approximately 29 hours, giving a total cost of 116 A100 GPU hours.

C.2 Hyperparameters

We maintain consistent core hyperparameters across both model scales (1.5B and 7B) to support a controlled comparison. Table 9 summarizes the detailed training configuration on `verl`.

Hyperparameter	Value
Algorithm	GRPO
Learning Rate	1×10^{-6}
Train Batch Size	128
Max Prompt Length	1024
Max Response Length	3072
Sampling Temperature	0.6
Rollouts per Sample (N)	8

Table 9: Default hyperparameters for training with the `verl` framework.

C.3 Baseline Implementations

For the Difficulty Curriculum Learning baseline (Narvekar et al., 2020), we sort the `DeepMath-103K` dataset (He et al., 2025) by its provided difficulty labels in ascending order and sample progressively throughout training.

To ensure a fair comparison, we train the ADARFT baseline (Shi et al., 2025) using GRPO (Shao et al., 2024). ADARFT estimates problem difficulty from rollout accuracy, originally using `Avg@128`. Given the scale of the 103K problem dataset, we adapt this standard and use `Avg@16` as the difficulty proxy for our ADARFT implementation, allowing us to simulate the ADARFT curriculum strategy within computational viability constraints.

D Interpretability Details of SAE-Guided Curriculum Construction

This appendix supplements the interpretability analysis in Section 5.4. We examine three types of evidence: the relation between SAE clusters and external topic annotations, the feature groups that provide predictive signal for difficulty estimation, and the intermediate variables exposed by the curriculum construction procedure.

Cluster	Conservative label	Main evidence	Interpretation
0	Derivative-centered calculus	Derivative applications; medium-high difficulty; moderate length	Local change, optimization, and derivative-based transformations.
1	Number-theoretic symbolic reasoning	Congruences; long solutions; medium difficulty	Congruence-style reasoning with symbolic manipulation and extended derivations.
2	Abstract algebra and proof structure	Group theory; high topic entropy; medium-high difficulty	Algebraic structures and proof-oriented reasoning beyond a single subfield.
3	Discrete and combinatorial reasoning	Combinatorics; highest mean difficulty	Counting, construction, and combinatorial proof patterns with high difficulty.
4	Integral and continuous reasoning	Integral applications; long solutions; high difficulty	Integration-related problems with continuous reasoning and longer derivations.
5	Broad calculus and transformations	Derivative-related topics; high entropy; long solutions	A heterogeneous continuous-math group involving functions and transformations.
6	Limits and sequence-style reasoning	Limits; highest top-topic share; lowest entropy	The most topic-concentrated cluster, centered on limits and sequences.
7	High-load limits and analysis	Limits; higher difficulty than Cluster 6; long solutions	More complex and heterogeneous variants of limit or analysis-style reasoning.
8	Short-form elementary algebra	Simple equations; lowest mean difficulty; shortest solutions; highest entropy	A mixed low-to-mid difficulty group with short algebraic structure.
9	Integration and symbolic procedures	Integration techniques; medium difficulty; moderate length	Procedural symbolic transformation, especially integration-related reasoning.

Table 10: Cluster-level semantic audit of SAE activation groups on DeepMath. Summaries are generated by a GPT-5.4-based agent and manually reviewed by the authors. Labels are conservative summaries rather than one-to-one topic annotations.

Feature group	Top 20	Top 100
`sol_mean`	10	36
`prompt_max`	5	26
`prompt_mean`	2	24
`sol_max`	2	11
`meta`	1	3

Table 11: Feature-group composition of the LightGBM difficulty proxy by gain.

D.1 Cluster-Level Structure

We compare the SAE cluster assignments on DeepMath with human-annotated topic metadata. The overall cluster–topic alignment is low: cluster-topic purity is 0.1095, NMI is 0.0881, and ARI is 0.0394. These results indicate that SAE clusters do not simply reproduce the human-defined mathematical topic taxonomy.

We therefore further inspect each cluster using top leaf topics, topic entropy, difficulty distribution, solution length, and representative samples. Table 10 provides a summary of the resulting cluster structure.

Some clusters exhibit clear topical enrichment, such as limits, combinatorics, group theory, and integration. At the same time, clusters also differ in problem format, solution length, symbolic density, proof orientation, and mean difficulty. These summaries should therefore be understood as coarse descriptions of activation-space structure, rather than as one-to-one topic labels.

This distinction is necessary because the external topic taxonomy and the SAE representation characterize different aspects of a sample. Topic labels describe the subject category assigned by the dataset, whereas SAE clusters are formed according to model-internal activation patterns. A cluster may therefore contain multiple mathematical topics while still preserving structure relevant to curriculum construction, such as similar solution profiles, reasoning formats, or difficulty ranges.

D.2 Feature-Level Signals

We next examine which input feature groups provide predictive signal for difficulty estimation. The curriculum pipeline uses prompt-side and solution-side SAE features, together with a small set of metadata features. For feature-group analysis, we train an auxiliary LightGBM model on the same difficulty-labeled subset and rank input columns by gain. This model is used only for interpretability analysis; the actual curriculum ranking scores are still produced by the calibrated difficulty proxy described in Section 3.

Table 11 reports the source distribution of the top-20 and top-100 gain-ranked features.

The top-ranked features come primarily from SAE activations rather than metadata. Among

High-gain feature	Observed high-activation pattern	Interpretation
`sol_mean/50831`	Abstract algebra, rings, groups, and proof-heavy problems	Solution-side activation associated with abstract algebraic structure and proof load.
`sol_mean/28006`	Measure, integration, and high-level analysis	Solution-side pattern related to advanced analysis and integration-style reasoning.
`prompt_max/16071`	Graphs, geometry, and Asymptote-style prompts	Prompt-side activation capturing localized diagrammatic or formatting cues.
`sol_mean/122476`	Combinatorics, graphs, and discrete structures	Solution-side signal associated with discrete reasoning and combinatorial structure.
`sol_mean/60349`	Simple equations and algebraic word problems	Lower- to mid-difficulty algebraic patterns involving short symbolic reasoning.
`sol_mean/88548`	Congruences, primes, and numeric expressions	Solution-side number-theoretic and numeric reasoning patterns.

Table 12: Semantic audit of individual high-gain features using high-activation samples. Summaries are produced by a GPT-5.4-based agent and manually reviewed by the authors.

the top-20 features, 19 are SAE-derived and 1 is metadata; among the top-100 features, 97 are SAE-derived and 3 are metadata. Within these features, solution-side mean features form the largest group, which is consistent with sustained solution-side activations providing correlational signals for difficulty estimation. Prompt-side max features also appear frequently, indicating that localized strong activations in the problem statement provide additional predictive cues.

To complement the feature-group statistics, we further inspect high-activation samples for several high-gain SAE features. Table 12 summarizes the semantic tendencies observed in these samples.

These high-activation samples exhibit several recurring patterns, including abstract algebra, measure and integration, discrete combinatorics, geometric formats, elementary algebra, and number-theoretic expressions. These descriptions indicate semantic tendencies only. SAE features are not strictly monosemantic: the same SAE feature may be activated in different contexts, and the same mathematical pattern may be distributed across multiple SAE features.

D.3 Procedure-Level Inspectability

The final curriculum order is constructed from a sequence of explicit intermediate variables. For each sample, the pipeline records its SAE representation, metadata, cluster assignment, raw difficulty prediction, calibrated ranking score, batch assignment, and, when moderate mixing is applied, its tail-swap partner. These variables allow researchers to inspect how a sample moves from the representation space to its final position in the curriculum.