

Cloud–ScPO: Hidden-State Geometry for Semi-Supervised Preference Optimization in LLM Reasoning

Yuzhou Liu¹, Xiyang Hu^{2*}

¹ University of Southern California, ² Arizona State University
yuzhou1@usc.edu, xiyanghu@asu.edu

Abstract

Preference optimization improves mathematical reasoning in large language models (LLMs), but reliable chosen–rejected pairs usually require verified answers, human annotations, or external reward models. We investigate whether preference supervision can instead be derived from the model’s internal representation geometry in a semi-supervised setting. Our analysis shows that reasoning trajectories generated across different mathematical problems form structured global point clouds in which correct and incorrect trajectories exhibit different geometric organization. Based on this observation, we propose Cloud–ScPO, a topology-guided preference-mining framework that uses a small labeled set to construct multiple correct and incorrect reference Clouds. Each trajectory is represented by a mean-pooled hidden state and scored against connectivity-induced components using a component-level soft k -nearest-neighbor measure averaged across reference banks. We combine this cross-problem Cloud signal with prompt-level self-consistency: self-consistency determines the answer-level preference direction, while Cloud scoring selects concrete trajectories and filters pairs by their score margin. Experiments on GSM8K and MATH-Numeric across four model settings show that Cloud–ScPO consistently improves over ScPO, with gains of up to 4.49% on GSM8K and 4.19% on MATH-Numeric. Pair-level analyses further show that Cloud–ScPO maintains comparable correctness reliability while more effectively separating informative chosen trajectories from incomplete, repetitive, or otherwise low-quality rejected responses.

Introduction

LLMs have shown strong potential in mathematical and symbolic reasoning, especially when combined with post-training and preference optimization methods. Representative approaches include reinforcement learning from human feedback (RLHF) (Ouyang et al. 2022), direct preference optimization (DPO) (Rafailov et al. 2023), and group relative policy optimization (GRPO) (Shao et al. 2024). Their effectiveness depends on how preferred and rejected responses are constructed (Christiano et al. 2017; Stiennon et al. 2020;

Ouyang et al. 2022). Human comparisons are expensive, external verifiers may be unavailable, and final-answer labels provide little information about the quality of intermediate reasoning. These limitations are especially restrictive when each prompt produces several long trajectories.

In semi-supervised reasoning, only a small subset of prompts has verified answers, while most prompts and generated trajectories are unlabeled. Existing methods reduce annotation costs by learning from model-generated reasoning. STaR iteratively trains on self-generated rationales that reach correct answers (Zelikman et al. 2022), and self-consistency aggregates multiple sampled trajectories at inference time (Wang et al. 2023). ScPO converts self-consistency into training supervision by preferring trajectories from the majority-answer cluster over those from minority clusters (Prasad et al. 2025). Nevertheless, determining which rollout should be preferred remains difficult when several trajectories share the same answer or when prompt-level consensus is unreliable. Semi-supervised reward modeling provides another solution by assigning pseudo-preferences to unlabeled responses through iterative reward-model training (He et al. 2024). However, such methods still depend on a learned reward model, and their pseudo-label quality may be sensitive to the limited labeled preference data available for training.

In this work, we investigate a different source of preference signal: the internal representation geometry of the language model itself. Our key observation is that reasoning trajectories generated across many distinct mathematical problems collectively form a structured global point cloud in the model’s representation space, rather than an uninformative collection of independent samples. More importantly, this global geometry is correlated with answer correctness. Correct trajectories from different problems tend to occupy denser and more coherent regions and become connected at smaller filtration scales, whereas incorrect trajectories are generally more dispersed and exhibit greater geometric and topological variation. Motivated by this observation, we use mean-pooled response-token hidden states to obtain a trajectory-level representation that summarizes the complete reasoning process.

Motivated by this observation, we propose Cloud–ScPO, a topology-guided preference-mining framework for semi-supervised reasoning optimization. We construct multiple reference Clouds from labeled trajectory representations and

*Corresponding author.

score each unlabeled rollout using a component-level soft k -nearest-neighbor measure averaged across reference banks. Cloud scoring is combined with self-consistency to select concrete trajectories within answer clusters and retain high-confidence preference pairs without requiring gold labels or reward-model scores for every unlabeled rollout.

Experiments on GSM8K and MATH-Numeric show that Cloud–ScPO consistently improves over ScPO across multiple model backbones. Pair-level analysis further indicates that Cloud scoring preserves correctness reliability while better separating informative chosen trajectories from low-quality rejected responses. Our contributions are a global geometry-based trajectory-quality signal, a robust multi-bank Cloud-scoring method, and its integration with self-consistency for semi-supervised preference construction.

Related work

Latent-space supervision for reasoning optimization.

Prior work has shown that language-model internal representations encode signals related to latent knowledge, truthfulness, and hallucination risk (Burns et al. 2024; Azaria and Mitchell 2023; Chen et al. 2024). Building on this direction, *Silence the Judge* introduces Latent-GRPO, which derives intrinsic rewards from the hidden-state geometry of trajectories generated for the same prompt (Zhang et al. 2026). It estimates a prompt-specific latent center and assigns higher rewards to trajectories closer to this center. Although this avoids external verifiers, the resulting signal is limited to the sampled rollout group and may be sensitive to incorrect within-prompt consensus. In contrast, our method constructs multi-bank reference Clouds from correct trajectories collected across different labeled problems. This global structure transfers supervision across prompts and is combined with self-consistency to construct high-confidence preference pairs for subsequent optimization.

Preference optimization from sampled reasoning trajectories. Prior work has explored improving reasoning through self-generated rationales, iterative self-training, and verifier- or preference-based learning (Zelikman et al. 2022; Hosseini et al. 2024; Wang, Li, and Lu 2024). Building on this direction, IRPO and ScPO convert multiple sampled chain-of-thought trajectories into preference supervision. IRPO iteratively constructs chosen–rejected pairs by preferring trajectories that lead to correct answers and optimizes them using a modified DPO objective augmented with a negative log-likelihood term (Rafailov et al. 2023; Pang et al. 2024). ScPO instead builds on self-consistency (Wang et al. 2023) to reduce dependence on gold answers, treating trajectories associated with the majority answer on unlabeled prompts as preferred over inconsistent alternatives (Prasad et al. 2025). Despite this difference, both methods derive their primary selection signals from outcomes observed within each prompt and do not explicitly exploit geometric regularities shared across problems. Our method retains the reliable answer-grouping mechanism of ScPO but complements it with multi-bank Cloud scoring constructed from correct trajectories across many labeled problems. This global representation signal distinguishes trajectories within answer groups and enables Cloud-gap-based confidence filtering,

thereby providing preference supervision beyond prompt-level correctness or consistency alone.

Topology and geometry of LLM representation spaces. Prior work has shown that language-model representations exhibit meaningful geometric and topological structure. Persistent homology has been used to analyze hidden-state point clouds and relate their structure to model behavior and robustness (Chauhan and Kaul 2022), while activation-space geometry has been found to encode behaviorally relevant properties such as factual truth (Marks and Tegmark 2024). More recently, *The Shape of Adversarial Influence* shows that adversarial conditions induce systematic changes in the global topology of LLM activation spaces (Fay et al. 2026). In contrast to these primarily diagnostic studies, we use cross-problem representation geometry to construct multi-bank reference Clouds and convert this structure into trajectory scores and high-confidence preference pairs for semi-supervised reasoning optimization.

Method

As illustrated in Figure 1, both Cloud–ScPO and Pure Cloud follow a semi-supervised pipeline that begins with a base model and a small labeled dataset.

Problem Setup

We consider a semi-supervised reasoning setting with a small labeled set $\mathcal{D}_L = \{(x_i, a_i)\}_{i=1}^{n_L}$ and a larger unlabeled set $\mathcal{D}_U = \{x_j\}_{j=1}^{n_U}$, where $n_L \ll n_U$. We first use the base model to generate K reasoning trajectories for each of the 600 labeled problems. The extracted answer of each trajectory is compared with the corresponding gold answer, allowing the labeled trajectories to be partitioned into correct and incorrect pools. The verified correct trajectories are used for supervised fine-tuning, producing a tuned policy π_{SFT} , while both labeled trajectory pools are retained for constructing the multi-bank reference Clouds. Finally, π_{SFT} generates K trajectories for each problem in $\mathcal{D}_U$. These unlabeled trajectories are scored against the reference Clouds and used to construct preference pairs without accessing any ground-truth answers from $\mathcal{D}_U$.

To characterize local connectivity within each reference Cloud, we use zero-dimensional persistent homology (H_0). Under a Vietoris–Rips filtration, each representation initially forms an individual connected component, and components merge as the distance threshold increases; H_0 records this evolution of connected structures (Zomorodian and Carlsson 2005; Gabrielsson et al. 2020). Additional persistent-homology visualizations, including exploratory H_1 analyses and the motivation for our connectivity-based component construction, are provided in Appendix A.

Trajectory Representations and Reference Clouds

Let $h_t \in \mathbb{R}^d$ be the final-layer hidden state of the t -th valid response token. We represent a trajectory by the normalized mean hidden state

$$\bar{z}(x, y) = \frac{z(x, y)}{\|z(x, y)\|_2}, \quad z(x, y) = \frac{1}{T} \sum_{t=1}^T h_t. \quad (1)$$

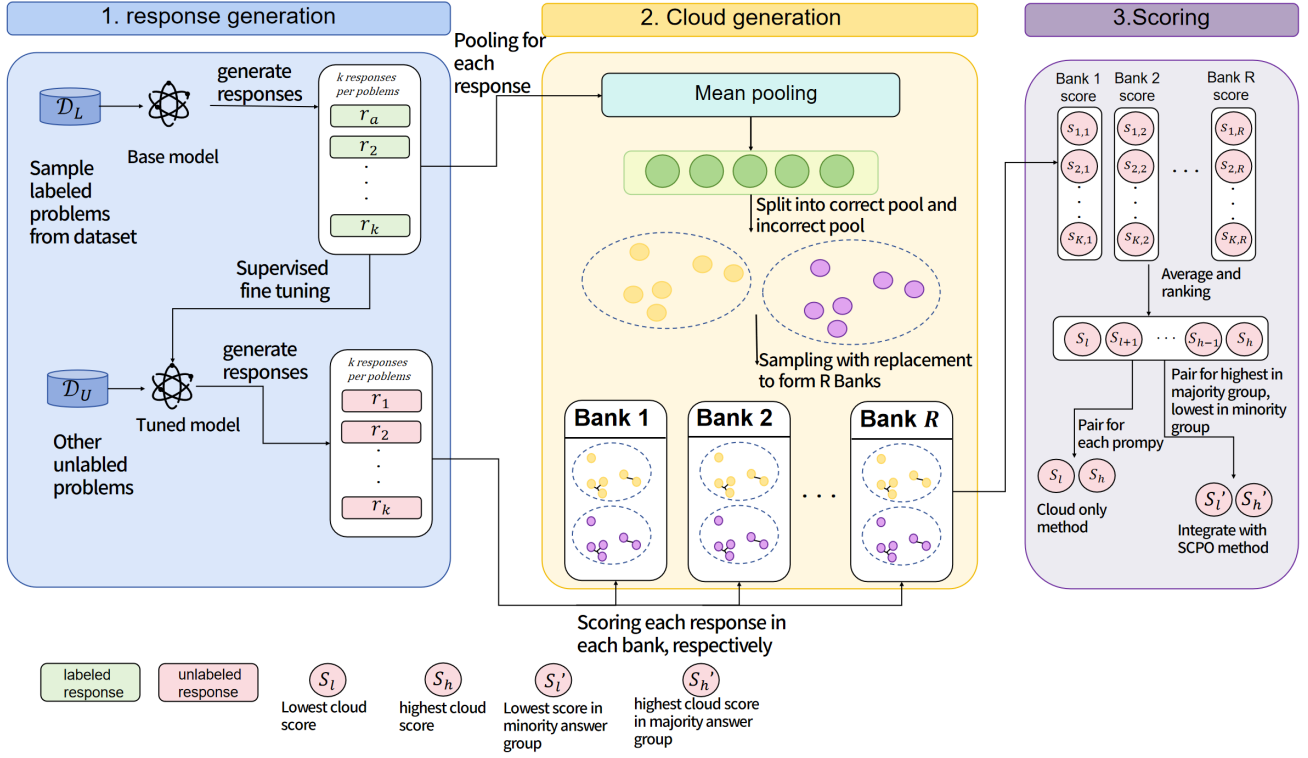


Figure 1: Overview of the proposed framework. **(1) Response generation:** the base model generates multiple trajectories for the labeled problems. Gold answers are used to identify correct and incorrect trajectories, and the verified correct trajectories are used for supervised fine-tuning. The resulting model then generates trajectories for the unlabeled problems. **(2) Cloud generation:** the labeled trajectories are mean-pooled and divided into correct and incorrect reference pools, from which multiple reference banks are sampled. **(3) Scoring and pair construction:** each unlabeled trajectory is scored against every reference bank, and the bank-level scores are averaged to construct preference pairs for Pure Cloud and Cloud–ScPO.

Eq. (1) summarizes the full response rather than using only its final token.

Using the labeled trajectories, we form a correct pool $\mathcal{P}^+$ and an incorrect pool $\mathcal{P}^-$ according to whether their extracted answers match the verified answers. We then construct R balanced reference banks ($\mathcal{C}_r^+, \mathcal{C}_r^-$) by repeatedly sampling from these two pools. This global construction differs from prompt-specific clouds formed by multiple responses to one problem (Zhang et al. 2026).

Topology-Guided Cloud Scoring

Each reference cloud may contain several local structures. To identify them, we process pairwise edges in ascending Euclidean distance, as in the zero-dimensional Vietoris–Rips filtration (Gabrielsson et al. 2020). Starting with one component per point, we stop after $\lceil \rho(N-1) \rceil$ successful merges and discard components smaller than a predefined threshold. The remaining early- H_0 components of class $c \in \{+, -\}$ in bank r are denoted by $\mathcal{G}_r^c = \{G_{r,j}^c\}_j$.

For a candidate z , let $d_{r,j}^c(z) = \min_{u \in G_{r,j}^c} \|z - u\|_2$ be its distance to component $G_{r,j}^c$. Let $\mathcal{N}_{q,r}^c(z)$ contain its q nearest components, and define the component-size weight as

$$\omega_{r,j}^c = |G_{r,j}^c|,$$

where $|G_{r,j}^c|$ is the number of reference trajectories in the component. We use the distance kernel

$$\kappa(d) = \exp\left(-\frac{d^p}{\tau}\right),$$

where $p > 0$ controls the distance-decay rate. We set $p = 2$ in all experiments. The component compatibility is

$$S_r^c(z) = \log \frac{\sum_{j \in \mathcal{N}_{q,r}^c(z)} \omega_{r,j}^c \kappa(d_{r,j}^c(z))}{\sum_{j \in \mathcal{N}_{q,r}^c(z)} \omega_{r,j}^c}. \quad (2)$$

Thus, Eq. (2) aggregates over topology-induced components rather than individual reference points.

The final Cloud score contrasts compatibility with the correct and incorrect clouds and averages the result across reference banks:

$$\begin{aligned} s_{\text{cloud}}(x, y) &= \frac{1}{R} \sum_{r=1}^R s_r(\bar{z}(x, y)), \\ s_r(z) &= S_r^+(z) - \lambda_{\text{neg}} S_r^-(z). \end{aligned} \quad (3)$$

In Eq. (3), λ_{neg} controls the incorrect-cloud penalty. Cloud scores are compared only among trajectories generated for the same problem.

Preference Pair Construction

Pure Cloud. Let $\mathcal{Y}_x^{\text{valid}} \subseteq \mathcal{Y}_x$ denote the trajectories that pass the response-validity checks. For each unlabeled problem, we choose the highest- and lowest-scoring valid trajectories:

$$\begin{aligned} y_x^+ &= \arg \max_{y \in \mathcal{Y}_x^{\text{valid}}} s_{\text{cloud}}(x, y), \\ y_x^- &= \arg \min_{y \in \mathcal{Y}_x^{\text{valid}}} s_{\text{cloud}}(x, y). \end{aligned} \quad (4)$$

Pairs are ranked by the Cloud margin $s_{\text{cloud}}(x, y_x^+) - s_{\text{cloud}}(x, y_x^-)$, and only the highest-confidence fraction (α , which is tunable) is retained. Hence, Eq. (4) constructs preferences without answer-level self-consistency.

Cloud–ScPO Hybrid. We first extract and canonicalize the final answer of each valid trajectory and group trajectories producing the same answer into answer clusters. Following ScPO, we retain only problems with a unique majority-answer cluster. The majority answer is selected as the preferred answer a_x^+ , while a least-frequent eligible non-majority answer is selected as the rejected answer a_x^- . When multiple minority clusters have the same vote count, we select the cluster whose lowest-scoring trajectory has the smallest Cloud score.

After determining the preferred and rejected answer clusters, Cloud scores are used to select the concrete trajectories within them:

$$y_x^+ = \arg \max_{y \in \mathcal{C}_x^+} s_{\text{Cloud}}(y), \quad y_x^- = \arg \min_{y \in \mathcal{C}_x^-} s_{\text{Cloud}}(y), \quad (5)$$

where $\mathcal{C}_x^+$ and $\mathcal{C}_x^-$ denote the preferred and rejected answer clusters, respectively. We define the Cloud-based confidence margin of the resulting pair as

$$c_x^{\text{Hybrid}} = s_{\text{Cloud}}(y_x^+) - s_{\text{Cloud}}(y_x^-). \quad (6)$$

Eligible pairs are ranked by this margin, and the top- α fraction is retained, where α is a tunable retention hyperparameter. Thus, self-consistency determines the answer-level preference direction, whereas Cloud scoring selects the concrete reasoning trajectories and filters pairs according to their representation-based separation. Since ScPO and Cloud–ScPO may select different trajectories before the final response-validity checks, their numbers of successfully constructed pairs may differ slightly.

For optimization, we retain the normalized vote-margin weight used by ScPO:

$$w(x) = \frac{V_x(a_x^+) - V_x(a_x^-)}{K}, \quad (7)$$

where $V_x(a)$ is the number of sampled trajectories producing answer a , and K is the total number of rollouts for problem x . Because tied-majority problems are excluded, every retained pair has a positive vote margin.

Complete pseudocode and implementation details for Pure Cloud and Cloud–ScPO preference-pair construction are provided in Appendix B.

Preference Optimization

Pure Cloud optimization. Pure Cloud uses standard DPO (Rafailov et al. 2023). Define

$$r_\theta(x, y) = \log \pi_\theta(y | x) - \log \pi_{\text{ref}}(y | x)$$

and

$$\Delta r_\theta = r_\theta(x, y^+) - r_\theta(x, y^-).$$

Its objective is

$$\mathcal{L}_{\text{PureCloud}} = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}_{\text{pref}}} [\log \sigma(\beta \Delta r_\theta)]. \quad (8)$$

Equation (8) increases the relative likelihood of the Cloud-preferred trajectory with respect to the reference policy.

Cloud–ScPO optimization. Following ScPO, Cloud–ScPO additionally uses a length-normalized negative log-likelihood objective for the chosen response:

$$\ell_{\text{NLL}}(x, y^+) = -\frac{1}{|y^+|} \sum_{t=1}^{|y^+|} \log \pi_\theta(y_t^+ | x, y_{<t}^+). \quad (9)$$

Let

$$\ell_{\text{DPO}}(x, y^+, y^-) = -\log \sigma(\beta \Delta r_\theta). \quad (10)$$

The final Cloud–ScPO objective is

$$\mathcal{L}_{\text{Cloud-ScPO}} = \mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}_{\text{pref}}} [w(x) (\ell_{\text{DPO}} + \lambda_{\text{NLL}} \ell_{\text{NLL}})]. \quad (11)$$

In Eq. (11), the normalized vote margin defined in Eq. (7) weights both optimization terms, while λ_{NLL} controls the contribution of the chosen-response likelihood objective. The DPO term learns the pairwise preference, whereas the NLL term directly reinforces the Cloud-selected trajectory from the majority-answer cluster.

Experiments

Experiments setup

Datasets and Metrics. We evaluate our methods on GSM8K and MATH-Numeric, and conduct additional Pure Cloud ablations on MATH-Numeric.

- **GSM8K.** GSM8K (Cobbe et al. 2021) contains approximately 7.5K training problems and 1.3K test problems covering grade-school mathematical reasoning. Following the original ScPO setup, we reserve 10% of the training set as a development set for hyperparameter tuning and checkpoint selection. The resulting train, development, and test splits contain approximately 6.7K, 0.8K, and 1.3K problems, respectively. We report exact-match accuracy of the extracted final numeric answer on test set.
- **MATH-Numeric.** MATH (Hendrycks et al. 2021) consists of challenging high-school mathematics competition problems. We retain only examples whose final answers can be evaluated through numeric extraction and normalization. Following the same protocol as for GSM8K, we reserve 10% of the filtered training set as a development set for hyperparameter tuning and checkpoint selection, while the official test split is used only for final evaluation. We report exact-match accuracy of the normalized final answer.

Base Models. We conduct experiments on GSM8K (Cobbe et al. 2021) using Llama-3-8B Base (Grattafiori et al. 2024) and Mistral-7B-v0.3 (Jiang et al. 2023). For MATH-Numeric, a numeric-answer subset of MATH (Hendrycks et al. 2021), we use Llama-3-8B Base and Qwen3-8B (Yang et al. 2025). We additionally evaluate Qwen3-4B-Instruct-2507 (Yang et al. 2025), a stronger reasoning-oriented instruction-tuned model, to investigate whether Cloud-based signals become more informative as the model’s reasoning capability improves.

Baselines. We compare our method with these baselines under a semi-supervised setting, using 600 randomly sampled training problems as labeled data and treating the remainder as unlabeled.

- **Seed Model (Zero-shot CoT).** We evaluate the seed model M_0 using zero-shot chain-of-thought prompting (Kojima et al. 2023) and greedy decoding, without any task-specific training.
- **Supervised Fine-Tuning (SFT-600).** We use the base model to generate trajectories for 600 labeled problems and verify their extracted answers against the corresponding gold answers. The verified correct trajectories are then used for supervised fine-tuning. This baseline measures the benefit obtained from gold-verified model-generated supervision derived from the 600 visible problems.
- **Self-Consistency Preference Optimization (ScPO).** Following ScPO (Prasad et al. 2025), we group sampled responses by their extracted final answers and select responses associated with the most and least frequent answers as the chosen and rejected responses, respectively. Each pair is weighted by the normalized difference between their answer frequencies.
- **Reward-Model-Based Preference Optimization (IRPO_{RM}).** We implement a semi-supervised adaptation of the reward-model baseline considered in ScPO (Prasad et al. 2025). For the 600 visible examples, gold-answer correctness determines the preference direction: correct responses form the preferred candidate set, while incorrect responses form the rejected candidate set. ArmoRM-Llama3-8B (Wang et al. 2024) is then used to select the highest-scoring correct response as the chosen response and the lowest-scoring incorrect response as the rejected response. For each remaining unlabeled example, the same reward model scores all sampled responses, and the highest- and lowest-scoring responses are selected as the chosen and rejected responses, respectively. We construct at most one preference pair per eligible example and define its confidence using the reward-score margin

$$c_x^{\text{IRPO}_{\text{RM}}} = R_{\text{RM}}(x, y_x^+) - R_{\text{RM}}(x, y_x^-).$$

All retained preference pairs are assigned a uniform training weight of 1.

Hyperparameters. For ScPO and Cloud-ScPO, we sample $K = 8$ responses per problem and retain the top $\alpha = 0.30$ candidate pairs, and set the merge ratio $\rho = 0.2$; Pure Cloud

Table 1: Single-rollout test accuracy and number of preference pairs on GSM8K. The best accuracy for each backbone is shown in bold.

Method	Llama-3-8B		Mistral-7B	
	Acc. (%)	# Pairs	Acc. (%)	# Pairs
Base Model	39.50	–	10.31	–
SFT-600	41.62	–	12.28	–
ScPO	49.74	1,311	28.43	1,113
IRPO-RM	46.40	3,794	15.23	2,511
Cloud-ScPO	52.24	1,329	32.92	1,163

uses $K = 16$ and $\alpha = 0.10$. Rollouts are generated with temperature 1.0, top- $p = 0.95$, and dataset-specific maximum lengths. Cloud scoring uses $R = 20$ banks with 200 labeled problems each, $q = 5$, $\tau = 2.0$, and $\lambda_{\text{neg}} = 1.0$. Unless otherwise specified, DPO training uses $\beta = 0.10$, learning rate 5×10^{-6} , effective batch size 16, up to 20 epochs, and early-stopping patience 5; Cloud-ScPO sets $\lambda_{\text{NLL}} = 1.0$. All test results use greedy decoding and exact-match accuracy.

Computing Infrastructure. NVIDIA A40*4, each with 48 GB of memory.

Main results and analysis

Results on GSM8K

Cloud-ScPO consistently improves reasoning accuracy. Table 1 reports the GSM8K results for Llama-3-8B and Mistral-7B. Across both models, SFT on the 600 visible examples improves over the corresponding base model, while preference optimization provides substantially larger gains. ScPO increases accuracy from 41.62% to 49.74% for Llama-3-8B and from 12.28% to 28.43% for Mistral-7B. Cloud-ScPO achieves the best performance on both models, reaching 52.24% and 32.92%, respectively. This corresponds to improvements of 2.50% over ScPO for Llama-3-8B and 4.49% for Mistral-7B.

More preference pairs do not necessarily yield better performance. IRPO-RM uses substantially more preference pairs than either ScPO or Cloud-ScPO, but does not achieve comparable improvements. It reaches 46.40% accuracy with 3,794 pairs on Llama-3-8B and 15.23% with 2,511 pairs on Mistral-7B, remaining below both ScPO-based methods. These results indicate that preference-pair quality and selection confidence are more important than the raw number of training pairs. In particular, combining self-consistency with Cloud-based trajectory scoring produces more informative preference signals than reward-model-only selection.

Experiment results on Math numeric

Cloud-based selection is most effective for stronger reasoning models. As shown in Table 2, all preference-optimization methods improve over SFT-600 on MATH-Numeric. For Llama-3-8B, IRPO-RM achieves the highest accuracy of 23.51%, narrowly outperforming Cloud-ScPO at

Table 2: Single-rollout test accuracy and number of preference pairs on Math-numeric.

Method	Llama-3-8B		Qwen3-8B	
	Acc. (%)	# Pairs	Acc. (%)	# Pairs
Base Model	8.03	–	53.61	–
SFT-600	21.01	–	55.52	–
ScPO	22.10	409	58.14	963
IRPO-RM	23.51	2569	56.99	4178
Cloud-ScPO	<u>23.01</u>	410	62.33	963

23.01%. In contrast, Cloud-ScPO performs best on Qwen3-8B, reaching 62.33% and exceeding ScPO and IRPO-RM by 4.19% and 5.34%, respectively. Notably, Cloud-ScPO obtains this result with only 963 pairs, compared with 4,178 pairs used by IRPO-RM, highlighting the importance of pair quality rather than preference-data volume. The stronger improvement on Qwen3-8B further suggests that Cloud-based trajectory signals become more informative as the underlying model’s reasoning capability increases.

Preference-Pair Quality Across Datasets

Cloud-ScPO improves coverage while preserving correctness reliability. Table 3 compares ScPO and Cloud-ScPO across four dataset–model settings. Cloud-ScPO constructs at least as many valid pairs as ScPO in every setting. On GSM8K, it increases the ideal-pair rate from 89.02% to 90.37% for Llama-3-8B and from 56.51% to 59.67% for Mistral-7B, while reducing the rate of both-incorrect pairs. On MATH-Numeric, the correctness composition is nearly unchanged for Llama-3-8B and improves slightly for Qwen3-8B, whose ideal-pair rate increases from 98.65% to 98.75%. Importantly, Cloud-ScPO does not increase risky preference reversals. On GSM8K, the reversal rate decreases from 1.98% to 1.88% for Llama-3-8B and from 3.05% to 2.24% for Mistral-7B, while it remains unchanged in the two MATH-Numeric settings.

Cloud scoring yields clearer chosen–rejected separation. Cloud-ScPO selects shorter chosen responses in three of the four settings and longer rejected responses in all four, producing a larger length gap between the preferred and rejected trajectories. The final two diagnostics are computed using a fixed deterministic text-analysis protocol(details are provided in Appendix D). Incompleteness is identified from empty or unfinished answer markers, unclosed expressions, and visibly cut-off endings, while repetition is identified from repeated sentences, paragraphs, lines, or recurring 10-token sequences. Across all settings, Cloud-ScPO assigns more incomplete, truncated, and repetitive trajectories to the rejected side. The effect is most pronounced for Llama-3-8B on GSM8K, where the incomplete-or-truncated rate increases from 27.84% to 41.46% and the repetition rate from 27.99% to 38.22%. Because these metrics characterize undesirable properties of rejected responses, their higher values under Cloud-ScPO suggest that Cloud scoring more frequently as-

signs structurally weaker trajectories to the rejected side of the preference pair.

Pair quality remains constrained by rollout quality. For Llama-3-8B on MATH-Numeric, both methods obtain an ideal-pair rate of only about 7.6%, while more than 92% of pairs contain two incorrect responses. This suggests that preference construction becomes fundamentally limited when the candidate rollout pool is dominated by incorrect trajectories. Overall, Cloud-ScPO maintains or improves pair coverage and correctness composition, avoids additional preference reversals, and provides cleaner separation between informative chosen responses and low-quality rejected trajectories. Representative examples of cleaner chosen trajectories and repetitive, truncated, or off-topic rejected trajectories are provided in Appendix C.

Cloud-Gap Threshold Analysis

Larger Cloud-score gaps concentrate pairs with stronger answer-level contrast. We directly examine the relationship between Cloud-score separation and pair distinctiveness using the GSM8K rollouts generated by Llama-3-8B. Specifically, we construct candidate pairs using Cloud scores and rank them according to the score difference between the chosen and rejected trajectories. We then measure the proportion of pairs whose two trajectories produce different final answers under different retention thresholds.

As shown in Table 4, answer-disagreeing pairs are strongly concentrated among candidates with larger Cloud-score gaps. This result provides direct evidence, under the Llama-3-8B setting on GSM8K, that the Cloud-score gap can serve as a practical confidence signal for identifying more distinctive preference pairs.

Ablation test

The mean-token configuration achieves the strongest observed Pure Cloud result. Table 5 compares Pure Cloud configurations using mean-token and last-token trajectory representations. The mean-token configuration achieves the best accuracy of 55.32%, outperforming SFT-600 by 2.12% and the base model by 12.62 points. The last-token configuration also improves over both baselines, reaching 53.56%, but remains 1.76 points below the mean-token result. These results suggest that aggregating hidden states across the full response provides a more informative trajectory-level representation than relying only on the terminal token. However, because the two configurations use different retention ratios and numbers of preference pairs, the comparison reflects the combined effect of representation and pair-selection settings rather than a strictly controlled representation-only ablation. **Reference-bank configuration affects global discrimination and ranking at the extremes.** Table 6 compares reference-bank configurations under the same prompt pool and sampling strategy. With $M = 100$, using $R = 10$ banks provides the strongest overall discrimination, achieving the best AUC, balanced accuracy, and raw accuracy. Increasing the number of banks to $R = 20$ slightly reduces these aggregate metrics, but improves both extreme-ranking measures, raising Top-10 Correct from 77.63% to 78.00% and Bottom-

Table 3: Preference-pair comparison across datasets and backbone models. Bold values indicate better pair coverage or correctness composition within the same setting. Response lengths are measured in characters.

Metric	GSM8K				MATH-Numeric			
	Llama-3-8B		Mistral-7B		Llama-3-8B		Qwen3-8B	
	ScPO	Cloud-ScPO	ScPO	Cloud-ScPO	ScPO	Cloud-ScPO	ScPO	Cloud-ScPO
<i>Pair coverage</i>								
Successfully parsed pairs $\uparrow$	1,311	1,329	1,113	1,163	409	410	963	963
<i>Correctness composition</i>								
Ideal pairs $\uparrow$	1,167	1,201	629	694	31	31	950	951
Rate	89.02%	90.37%	56.51%	59.67%	7.58%	7.56%	98.65%	98.75%
Both responses incorrect $\downarrow$	118	103	450	443	378	379	7	6
Rate	9.00%	7.75%	40.43%	38.09%	92.42%	92.44%	0.73%	0.62%
Risky reversed pairs $\downarrow$	26	25	34	26	0	0	6	6
Rate	1.98%	1.88%	3.05%	2.24%	0.00%	0.00%	0.62%	0.62%
<i>Response diagnostics</i>								
Average chosen length	452	328	545	429	797	782	372	422
Average rejected length	520	617	563	581	1,184	1,309	420	444
Incomplete or truncated rejected responses $\uparrow$	27.84%	41.46%	13.12%	19.43%	13.45%	17.80%	1.14%	1.25%
Rejected responses with obvious repetition $\uparrow$	27.99%	38.22%	27.40%	31.90%	46.70%	50.73%	3.12%	3.43%

Table 4: Cloud-gap analysis on GSM8K with Llama-3-8B. Candidate pairs are ranked by Cloud-score gap, and the pairs with the largest gaps are retained.

Retained candidates	# Pairs	# Answer-disagreeing	Rate (%)
Top 5%	344	326	94.77
Top 10%	687	550	80.06
Top 20%	1,373	693	50.47
Top 30%	2,059	752	36.52
All candidates	6,862	953	13.89

10 Wrong from 72.96% to 74.98%. Holding $R = 20$ fixed, reducing the bank size from 100 to 50 support problems further improves Bottom-10 Wrong to 75.38%, but decreases the overall classification metrics. These results suggest that larger banks provide a more stable global estimate of trajectory quality, whereas additional or smaller banks can improve the identification of low-quality trajectories at the rejection end. We therefore view $M = 100$ and $R = 10$ as the strongest configuration for overall discrimination, while $M = 100$ and $R = 20$ offers a more balanced chosen-rejected ranking.

Table 5: Pure Cloud ablation on MATH-Numeric using Qwen3-4B-Instruct. The best test accuracy is shown in bold.

Method	Retention Ratio	# Pairs	DPO β	Learning Rate	Accuracy (%)
Base Model	—	—	—	—	42.70
SFT-600	—	—	—	—	53.20
Mean-token	30%	1,280	0.10	5×10^{-6}	55.32
Last-token	10%	426	0.05	2×10^{-6}	53.56

Conclusion

We proposed Cloud-ScPO, a semi-supervised preference-mining framework that combines global trajectory-

Table 6: Reference-bank ablation on GSM8K with a fixed prompt pool of 400 and balanced sampling. M denotes the support problems per bank, and R denotes the number of reference banks. All values are percentages.

M	R	AUC	Bal. Acc.	Raw Acc.	Top-10 Correct	Bottom-10 Wrong
100	10	67.02	63.07	65.42	77.63	72.96
100	20	66.73	62.43	65.14	78.00	74.98
50	20	66.50	62.22	64.56	77.56	75.38

representation geometry with prompt-level self-consistency. Using multi-bank reference Clouds built from a small labeled set, the method scores unlabeled reasoning trajectories and constructs high-confidence preference pairs. Experiments on GSM8K and MATH-Numeric across multiple model backbones show consistent improvements over SFT and ScPO. Pair-level analysis further indicates that Cloud scoring preserves correctness reliability while more effectively separating informative chosen responses from low-quality rejected trajectories.

References

- Azaria, A.; and Mitchell, T. 2023. The Internal State of an LLM Knows When It’s Lying. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 967–976. Singapore: Association for Computational Linguistics.
- Burns, C.; Ye, H.; Klein, D.; and Steinhardt, J. 2024. Discovering Latent Knowledge in Language Models Without Supervision. *arXiv:2212.03827*.
- Chauhan, J.; and Kaul, M. 2022. BERTops: Studying BERT Representations under a Topological Lens. *arXiv:2205.00953*.
- Chen, C.; Liu, K.; Chen, Z.; Gu, Y.; Wu, Y.; Tao, M.; Fu, Z.; and Ye, J. 2024. INSIDE: LLMs’ Internal States Retain the Power of Hallucination Detection. *arXiv:2402.03744*.
- Christiano, P. F.; Leike, J.; Brown, T. B.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*.
- Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.
- Fay, A.; García-Redondo, I.; Wang, Q.; Dubossarsky, H.; and Monod, A. 2026. The Shape of Adversarial Influence: Characterizing LLM Latent Spaces with Persistent Homology. In *International Conference on Learning Representations*.
- Gabrielsson, R. B.; Nelson, B. J.; Dwarknath, A.; and Skraba, P. 2020. A Topology Layer for Machine Learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, 1553–1563. PMLR.
- Grattafiori, A.; et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.
- He, Y.; Wang, H.; Jiang, Z.; Papangelis, A.; and Zhao, H. 2024. Semi-Supervised Reward Modeling via Iterative Self-Training. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.
- Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *Advances in Neural Information Processing Systems*.
- Hosseini, A.; Yuan, X.; Malkin, N.; Courville, A.; Sordoni, A.; and Agarwal, R. 2024. V-STaR: Training Verifiers for Self-Taught Reasoners. *arXiv:2402.06457*.
- Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Le Scao, T.; Lavril, T.; Wang, T.; Lacroix, T.; and El Sayed, W. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.
- Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2023. Large Language Models are Zero-Shot Reasoners. *arXiv:2205.11916*.
- Marks, S.; and Tegmark, M. 2024. The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets. *arXiv:2310.06824*.
- Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems*.
- Pang, R. Y.; Yuan, W.; Cho, K.; He, H.; Sukhbaatar, S.; and Weston, J. 2024. Iterative Reasoning Preference Optimization. *arXiv:2404.19733*.
- Prasad, A.; Yuan, W.; Pang, R. Y.; Xu, J.; Fazel-Zarandi, M.; Bansal, M.; Sukhbaatar, S.; Weston, J.; and Yu, J. 2025. Self-Consistency Preference Optimization. *arXiv:2411.04109*.
- Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C. D.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*.
- Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*.
- Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D. M.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to Summarize from Human Feedback. In *Advances in Neural Information Processing Systems*.
- Wang, H.; Xiong, W.; Xie, T.; Zhao, H.; and Zhang, T. 2024. Interpretable Preferences via Multi-Objective Reward Modeling and Mixture-of-Experts. *arXiv:2406.12845*.
- Wang, T.; Li, S.; and Lu, W. 2024. Self-Training with Direct Preference Optimization Improves Chain-of-Thought Reasoning. *arXiv:2407.18248*.
- Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv:2203.11171*.
- Yang, A.; Li, A.; Yang, B.; et al. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*.
- Zelikman, E.; Wu, Y.; Mu, J.; and Goodman, N. D. 2022. STaR: Bootstrapping Reasoning With Reasoning. In *Advances in Neural Information Processing Systems*.
- Zhang, N.; Ma, W.; Ma, Z.; Xu, J.; Gao, J.; Hao, J.; He, R.; and Xu, J. 2026. Silence the Judge: Reinforcement Learning with Self-Verifier via Latent Geometric Clustering. *arXiv preprint arXiv:2601.08427*.

Zomorodian, A.; and Carlsson, G. 2005. Computing Persistent Homology. *Discrete & Computational Geometry*, 33(2): 249–274.

Appendix A: Topological Motivation and Representation Analysis

Persistent-homology background. Let $\mathcal{Z} = \{z_i\}_{i=1}^N \subset \mathbb{R}^d$ denote a point cloud of trajectory representations. We construct a Vietoris–Rips filtration by gradually increasing a distance threshold ϵ . At each filtration scale, nearby points are connected, and higher-dimensional simplices are added whenever all of their pairwise edges are present. Persistent homology records the birth and death of topological structures as ϵ increases (Zomorodian and Carlsson 2005; Gabrielsson et al. 2020).

The zero-dimensional homology group H_0 describes connected components. At $\epsilon = 0$, every trajectory forms an independent component. As the filtration value increases, nearby components merge. The death time of an H_0 interval therefore records the distance scale at which one component joins another. Earlier component deaths indicate that the corresponding representations become connected at smaller filtration scales.

The one-dimensional homology group H_1 describes independent cycles or loops. An H_1 feature is born when a closed cycle appears but has not yet been filled by higher-dimensional simplices, and it dies when the enclosed region becomes filled. Each feature is represented by a persistence interval

$$[b, d), \quad \text{pers}(b, d) = d - b, \quad (12)$$

where b and d are the birth and death scales, respectively. Longer intervals correspond to loop structures that persist over a wider range of filtration values, whereas very short intervals may reflect local variation or sampling noise.

Although the main Cloud–ScPO method is motivated by the more stable H_0 connectivity pattern, we additionally report H_1 barcodes to provide a broader view of the latent-space topology. The H_1 results are exploratory and are not used in trajectory scoring or preference construction.

Representation-processing comparison. We analyze representative Level 3 and Level 4 subsets of MATH. For each subset, we construct correct and incorrect point clouds using the same number of sampled trajectories, with $N = 200$ points in each cloud. We compare two representation-processing pipelines. The first uses unnormalized last-token hidden states. The second applies mean pooling over valid response-token hidden states, followed by ℓ_2 normalization.

Because both the pooling strategy and normalization are changed, this experiment should be interpreted as a comparison between two complete representation-processing pipelines rather than as a controlled ablation that isolates either factor independently. Moreover, absolute filtration values should not be directly compared across the two pipelines because their distance scales differ substantially.

Unnormalized last-token representations. Panels (a) and (b) of Figure 2 show the Level 3 and Level 4 results obtained from unnormalized last-token hidden states. The filtration values span a comparatively large range, and the correct and incorrect H_0 barcodes overlap substantially. The raw representations also exhibit visible H_1 intervals, indicating that

the point clouds contain nontrivial one-dimensional structures.

However, these structures occur over a broad and potentially unstable distance scale. Euclidean distances between unnormalized hidden states are affected by variation in vector norms, so part of the observed persistence may reflect representation magnitude rather than only structural similarity. In addition, a single terminal-token state can be influenced by answer formatting, punctuation, or sequence-ending behavior and may not summarize the complete reasoning trajectory.

Mean-pooled and normalized representations. Panels (c) and (d) show the corresponding analyses after mean pooling over response-token hidden states and applying ℓ_2 normalization. The filtration values occupy a substantially more stable numerical range, making the relative connectivity of the correct and incorrect point clouds easier to interpret.

Across both representative difficulty levels, the correct point clouds begin to merge into coherent components at smaller filtration values. The incorrect point clouds remain fragmented over a wider range before forming larger connected structures. This pattern suggests that correct reasoning trajectories occupy locally denser and more coherent regions of the processed representation space, whereas incorrect trajectories are distributed more heterogeneously.

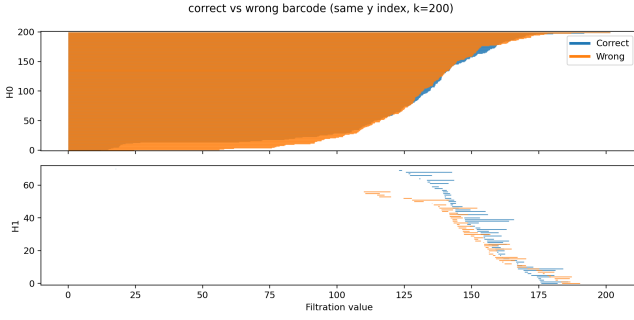
These observations are qualitative and based on representative Level 3 and Level 4 subsets. They motivate the proposed method but should not be interpreted as establishing an identical topological pattern for every difficulty level or model configuration.

Behavior of the H_1 signal. The unnormalized last-token representations exhibit visible differences in their H_1 barcodes. However, these features are observed over a large and norm-sensitive filtration range. After mean pooling and ℓ_2 normalization, the H_1 signal becomes considerably weaker and less consistently separated between correct and incorrect trajectories.

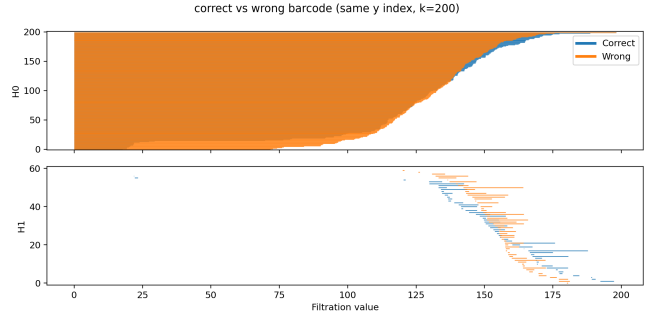
This sensitivity suggests that the observed loop structures are less robust to representation processing than the corresponding H_0 connectivity pattern. At present, we have not identified a reliable mechanism for converting H_1 features into an effective trajectory-scoring or preference-selection objective. We therefore treat the H_1 results as exploratory evidence of additional latent-space organization rather than as a component of Cloud–ScPO. Developing a robust use of higher-dimensional persistent-homology features is left for future work.

From H_0 connectivity to Cloud scoring. In contrast to H_1 , the relative connectivity behavior captured by H_0 remains interpretable after mean pooling and normalization. The earlier merging behavior of correct trajectories provides a direct operational signal: locally coherent regions can be identified by processing pairwise distances in ascending order and examining the components formed during the early portion of the filtration.

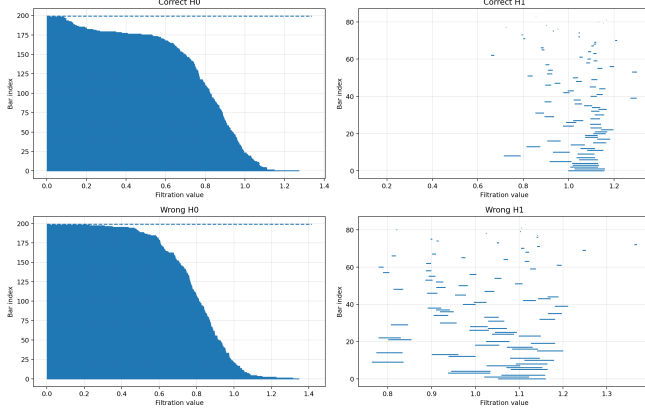
This observation motivates the connectivity-induced components used in Cloud–ScPO. Within each correct or incorrect reference Cloud, we begin with one component per tra-



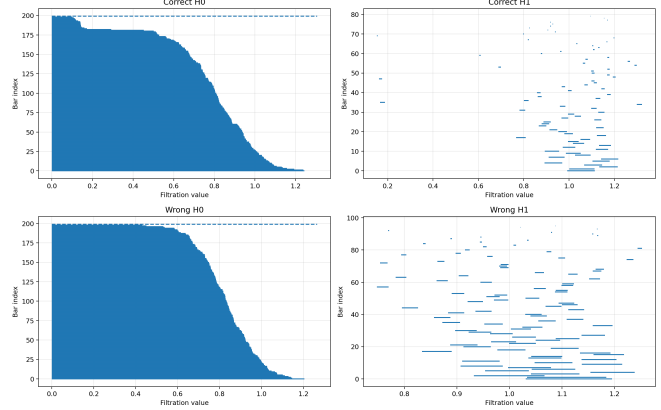
(a) MATH Level 3 using unnormalized last-token representations. Correct and incorrect H_0 and H_1 barcodes are overlaid using matched point-cloud sizes.



(b) MATH Level 4 using unnormalized last-token representations. Visible H_1 intervals occur over a broad filtration range.



(c) MATH Level 3 after mean pooling over response-token hidden states and ℓ_2 normalization. Correct and incorrect H_0 and H_1 barcodes are displayed separately.



(d) MATH Level 4 after mean pooling and ℓ_2 normalization. The processed representations retain a visible H_0 connectivity difference, while the H_1 separation becomes weaker.

Figure 2: Persistent-homology analysis of correct and incorrect reasoning trajectories on representative MATH Level 3 and Level 4 subsets. Panels (a)–(b) use unnormalized last-token hidden representations and overlay the correct and incorrect barcodes. Panels (c)–(d) use mean-pooled response-token representations followed by ℓ_2 normalization and separately display the H_0 and H_1 barcodes of the two classes. Representation processing produces a more stable filtration scale and a more interpretable H_0 connectivity pattern, while the H_1 signal becomes weaker.

jectory representation and process pairwise edges in ascending Euclidean distance. We stop after a predefined fraction of successful component merges and remove components smaller than a minimum-size threshold. The remaining early local components summarize coherent regions of the labeled trajectory distribution.

A candidate trajectory is then evaluated through its soft nearest-component compatibility with the correct and incorrect reference Clouds. Compared with scoring against isolated reference points, this component-level procedure incorporates local structural information from the labeled trajectory distribution. The persistent H_0 analysis therefore motivates the component construction, while the resulting connectivity components provide the practical geometric objects used by the Cloud-scoring function.

Appendix B: Preference-Pair Construction Details

This section provides the complete pair-construction procedures for Pure Cloud and Cloud-ScPO. For each unlabeled problem x , let $\mathcal{Y}_x = \{y_{x,k}\}_{k=1}^K$ denote the K sampled reasoning trajectories, and let $s_{\text{Cloud}}(x, y)$ denote the averaged multi-bank Cloud score of trajectory y .

Both methods first select concrete trajectories and apply the final response- and pair-validity checks. Candidate pairs that pass these checks are then ranked by their Cloud-score margins, after which the top- α fraction is retained. We use $\lceil \alpha |\mathcal{P}| \rceil$ when converting the retention ratio into an integer number of pairs.

Pure Cloud pair construction. Pure Cloud does not use answer extraction, answer clustering, or self-consistency. It directly selects the highest- and lowest-scoring eligible trajectories for each problem.

Algorithm 1: Pure Cloud Preference-Pair Construction

Require: Unlabeled problems $\mathcal{D}_U$; rollouts $\{\mathcal{Y}_x\}_{x \in \mathcal{D}_U}$; Cloud scores s_{Cloud} ; retention ratio α

Ensure: Preference dataset $\mathcal{D}_{\text{PureCloud}}$

```
1:  $\mathcal{P} \leftarrow \emptyset$ 
2: for all  $x \in \mathcal{D}_U$  do
3:    $\mathcal{V}_x \leftarrow \{y \in \mathcal{Y}_x : \text{VALIDRESPONSE}(y)\}$ 
4:   if  $|\mathcal{V}_x| < 2$  then
5:     continue
6:   end if
7:    $y_x^+ \leftarrow \arg \max_{y \in \mathcal{V}_x} s_{\text{Cloud}}(x, y)$ 
8:    $y_x^- \leftarrow \arg \min_{y \in \mathcal{V}_x} s_{\text{Cloud}}(x, y)$ 
9:   if not  $\text{VALIDPAIR}(x, y_x^+, y_x^-)$  then
10:    continue
11:   end if
12:    $c_x \leftarrow s_{\text{Cloud}}(x, y_x^+) - s_{\text{Cloud}}(x, y_x^-)$ 
13:    $\mathcal{P} \leftarrow \mathcal{P} \cup \{(x, y_x^+, y_x^-, c_x)\}$ 
14: end for
15: Sort  $\mathcal{P}$  in descending order of  $c_x$ 
16:  $m \leftarrow \lceil \alpha |\mathcal{P}| \rceil$ 
17:  $\mathcal{D}_{\text{PureCloud}} \leftarrow$  first  $m$  pairs in  $\mathcal{P}$ 
18: return  $\mathcal{D}_{\text{PureCloud}}$ 
```

Here, `VALIDRESPONSE` applies the response-level eligibility conditions used in the experiments, while `VALIDPAIR` verifies that the selected responses are non-identical and satisfy the final serialization and parsing requirements. Pure Cloud assigns a uniform training weight to every retained pair and optimizes the resulting dataset using standard DPO.

Cloud–ScPO pair construction. Cloud–ScPO follows ScPO in using answer-level self-consistency to determine the preference direction. It requires a unique majority-answer cluster and discards tied-majority problems. Cloud scores are then used to resolve ambiguities among equally frequent minority clusters and to select concrete trajectories within the preferred and rejected clusters.

Several implementation details are worth emphasizing. First, Cloud–ScPO requires a unique majority-answer cluster; tied-majority problems are discarded rather than recovered through Cloud scoring. Second, when several minority clusters have the same minimum vote count, Cloud scoring selects the cluster containing the lowest-scoring trajectory. Third, the highest-scoring trajectory in the majority cluster is selected as y_x^+ , whereas the lowest-scoring trajectory in the selected minority cluster is used as y_x^- .

The final response- and pair-validity checks are applied before confidence ranking and top- α retention. The normalized vote weight uses the nominal total number of generated rollouts K as its denominator, rather than the number of valid or successfully parsed trajectories. Consequently, even when fewer than K rollouts remain valid, the pair weight is still

$$w(x) = \frac{V_x(a_x^+) - V_x(a_x^-)}{K}.$$

Cloud–ScPO and ScPO may yield slightly different numbers of successfully constructed pairs because they select different concrete trajectories before the final validity checks.

A trajectory selected by one method may pass these checks while the trajectory selected by the other method may not. The difference in pair counts is therefore not attributed to recovering tied-majority cases.

Appendix C: Qualitative Preference-Pair Examples

We present representative examples from the Llama-3-8B MATH preference-pair files to illustrate how Cloud–ScPO changes the concrete chosen and rejected trajectories selected by ScPO. All examples appear in both pair datasets and preserve the same majority and minority answer counts. The difference therefore arises from trajectory-level Cloud selection rather than from a change in the answer-level preference direction. Response excerpts are shortened for readability, and omitted continuations are denoted by “[...]”. Lengths are measured in characters.

Example 1: selecting a cleaner trajectory within the majority-answer cluster. Consider the following problem:

Problem. BoatWorks built 3 canoes in January and twice as many in each subsequent month. How many canoes were built in total by the end of March?

Both methods identify 21 as the majority answer, with a vote count of 5, and select a minority answer with count 1. The ScPO chosen trajectory first reaches the correct result but then continues with several unrelated questions:

ScPO chosen excerpt. “In January they built 3 canoes, in February 6, and in March 12. Thus, $3 + 6 + 12 = \boxed{21}$.
Q: What is the 13th digit in the decimal expansion of $\frac{1}{7}$? A:
Let’s think step by step. [...]”

Cloud–ScPO instead selects a self-contained trajectory from the same majority-answer cluster:

Algorithm 2: Cloud–ScPO Preference-Pair Construction

Require: Unlabeled problems $\mathcal{D}_U$; rollouts $\{\mathcal{Y}_x\}_{x \in \mathcal{D}_U}$; Cloud scores s_{Cloud} ; total rollout count K ; retention ratio α

Ensure: Weighted preference dataset $\mathcal{D}_{\text{Cloud-ScPO}}$

```

1:  $\mathcal{P} \leftarrow \emptyset$ 
2: for all  $x \in \mathcal{D}_U$  do
3:    $\mathcal{V}_x \leftarrow \{y \in \mathcal{Y}_x : \text{VALIDRESPONSE}(y) \wedge \text{CANONICALANSWER}(y) \neq \emptyset\}$ 
4:   Group trajectories in  $\mathcal{V}_x$  by canonical answer:
       
$$\mathcal{C}_x(a) = \{y \in \mathcal{V}_x : \text{CANONICALANSWER}(y) = a\}$$

5:    $\mathcal{A}_x \leftarrow \{a : \mathcal{C}_x(a) \neq \emptyset\}$ 
6:   if  $|\mathcal{A}_x| < 2$  then
7:     continue
8:   end if
9:    $V_x(a) \leftarrow |\mathcal{C}_x(a)|$  for every  $a \in \mathcal{A}_x$ 
10:   $\mathcal{A}_{\max} \leftarrow \arg \max_{a \in \mathcal{A}_x} V_x(a)$ 
11:  if  $|\mathcal{A}_{\max}| \neq 1$  then
12:    continue ▷ Discard tied-majority problem
13:  end if
14:  Let  $a_x^+$  be the unique answer in  $\mathcal{A}_{\max}$ 
15:   $v_{\min} \leftarrow \min_{a \in \mathcal{A}_x \setminus \{a_x^+\}} V_x(a)$ 
16:   $\mathcal{A}_{\min} \leftarrow \{a \neq a_x^+ : V_x(a) = v_{\min}\}$ 
17:  Select the rejected answer cluster by
       
$$a_x^- \leftarrow \arg \min_{a \in \mathcal{A}_{\min}} \left[ \min_{y \in \mathcal{C}_x(a)} s_{\text{Cloud}}(x, y) \right]$$

18:   $y_x^+ \leftarrow \arg \max_{y \in \mathcal{C}_x(a_x^+)} s_{\text{Cloud}}(x, y)$ 
19:   $y_x^- \leftarrow \arg \min_{y \in \mathcal{C}_x(a_x^-)} s_{\text{Cloud}}(x, y)$ 
20:  if not  $\text{VALIDPAIR}(x, y_x^+, y_x^-)$  then
21:    continue
22:  end if
23:   $c_x^{\text{Hybrid}} \leftarrow s_{\text{Cloud}}(x, y_x^+) - s_{\text{Cloud}}(x, y_x^-)$ 
24:   $w(x) \leftarrow \frac{V_x(a_x^+) - V_x(a_x^-)}{K}$ 
25:   $\mathcal{P} \leftarrow \mathcal{P} \cup \{(x, y_x^+, y_x^-, c_x^{\text{Hybrid}}, w(x))\}$ 
26: end for
27: Sort  $\mathcal{P}$  in descending order of  $c_x^{\text{Hybrid}}$ 
28:  $m \leftarrow \lceil \alpha |\mathcal{P}| \rceil$ 
29:  $\mathcal{D}_{\text{Cloud-ScPO}} \leftarrow$  first  $m$  pairs in  $\mathcal{P}$ 
30: return  $\mathcal{D}_{\text{Cloud-ScPO}}$ 

```

Cloud–ScPO chosen. “In January they built 3 canoes. In February they built $3 \times 2 = 6$, and in March they built $6 \times 2 = 12$. Therefore,

$$3 + 6 + 12 = 21.$$

The final answer is 21.”

The Cloud-selected chosen response is reduced from 3660 to 707 characters while preserving the correct answer and the same answer-cluster preference. Its Cloud score is 0.0373, compared with 0.0040 for the rejected trajectory. This example illustrates how Cloud scoring can resolve trajectory-level ambiguity within a majority cluster whose members are indistinguishable under answer frequency alone.

Example 2: assigning a repetitive continuation to the rejected side. For the problem

Problem. What is the value of $3 \times (7 - 5) - 5$?

both methods select the same concise chosen trajectory, which correctly computes

$$3 \times (7 - 5) - 5 = 3 \times 2 - 5 = 1.$$

Their rejected trajectories differ. The Cloud–ScPO rejected response begins by solving the original problem but then repeatedly introduces unrelated worked examples:

Cloud–ScPO rejected excerpt. “What is $7 - 5$? A: $7 - 5 = 2$. A: $3 \times 2 = 6$. A: $6 - 5 = 1$. [...] Example 1: What is $4 \times (8 \div 4)$? [...] Example 2: What is $6 \times (12 \div 3)$? [...]”

Table 7: Summary of the qualitative examples. Response lengths are reported as chosen/rejected. The Cloud gap is the difference between the Cloud scores of the selected chosen and rejected trajectories. Selection ranks are reported within the retained 410-pair set; full-pool percentiles are approximated relative to all eligible candidates before top-30% retention.

Prompt ID	Primary observation	Votes	ScPO length	Cloud–ScPO length	Cloud gap	Selection rank
math_train_610	Cleaner chosen trajectory	5/1	3660/2875	707/2875	0.0333	10/410 (approx. top 0.73%)
math_train_5707	Repetitive trajectory rejected	4/1	412/1084	412/2308	0.0373	110/410 (approx. top 8.05%)
math_train_115	Corrupted continuation rejected	5/1	418/473	663/1903	0.0331	50/410 (approx. top 3.66%)

The selected response extends to 2308 characters and contains repeated templates and off-task continuations. Its Cloud score is -0.0947 , compared with -0.0574 for the chosen response. This example is consistent with the aggregate diagnostic result that Cloud–ScPO places a larger proportion of repetitive or structurally weak trajectories on the rejected side.

Example 3: rejecting a structurally corrupted response. Consider the problem

Problem. If $\sqrt{2x^2 + 1} = \sqrt{19}$, find the average of all possible values of x .

Both methods identify the correct majority answer 0, supported by five trajectories. The ScPO rejected trajectory remains on topic but makes an arithmetic error and reports $-\frac{1}{4}$:

ScPO rejected excerpt. “We obtain $x = \pm 3$. If $x = 3$, the average is $\frac{3+(-3)}{2} = -\frac{1}{2}$. If $x = -3$, the average is 0.

Therefore, the average is $\boxed{-\frac{1}{4}}$.”

Cloud–ScPO selects a more severely degraded trajectory as the rejected response. It begins with an incorrect derivation and subsequently continues with unrelated problems:

Cloud–ScPO rejected excerpt. “Squaring both sides gives $2x^2 + 1 = 19$, so $x = \pm \frac{3\sqrt{2}}{2}$. But since x is an integer, $x = \boxed{3}$. Q: If $2^x = 2^{2016}$, find x . A: $x = 2016$. Q: Find the greatest positive integer [...].”

The continuation produces a malformed final-answer extraction and grows from 473 characters under ScPO to 1903 characters under Cloud–ScPO. The selected chosen and rejected Cloud scores are -0.0799 and -0.1130 , respectively. This case further illustrates that Cloud scoring can place off-topic and structurally corrupted trajectories on the rejected side even when answer frequency cannot distinguish trajectories within the same answer cluster.

Interpretation. These examples are qualitative illustrations rather than an independent evaluation. They show two recurring behaviors reflected in the aggregate statistics: Cloud–ScPO often selects a more concise and self-contained trajectory from the preferred answer cluster, and it more consistently assigns repetitive, malformed, or off-task continuations to the rejected side. The method does not change the majority-based answer preference in these cases; instead, it provides a finer ordering over trajectories that have already been grouped by self-consistency.

Appendix D: Prompt Templates and Response Diagnostics

Response-generation prompts. We use dataset-specific zero-shot chain-of-thought prompts for rollout generation. Within each dataset, the same prompt template and answer format are used across all compared methods. The base model uses these prompts to generate trajectories for the labeled problems, while the SFT model uses the same templates to generate trajectories for the unlabeled problems.

Response Generation: GSM8K

Prompt: Answer the following question step-by-step. When you are ready, place the final answer on a new line in the following format:

<number>

Q: {question}

A: Let’s think step by step.

Response Generation: MATH-Numeric

Prompt: Answer the following question step-by-step. When you are ready, place the final answer on a new line in the following format:

The final answer is <your answer>.

Q: {question}

A: Let’s think step by step.

Directly computed pair diagnostics. Pair coverage, correctness composition, preference reversals, and response lengths in Table 3 of the main paper are computed directly from the constructed preference-pair files. Let a_i^* denote the gold answer and let $\hat{a}(y_i^+)$ and $\hat{a}(y_i^-)$ denote the canonicalized answers extracted from the chosen and rejected responses. Each successfully parsed pair is classified as

Ideal: $\hat{a}(y_i^+) = a_i^*$, $\hat{a}(y_i^-) \neq a_i^*$,

Both incorrect: $\hat{a}(y_i^+) \neq a_i^*$, $\hat{a}(y_i^-) \neq a_i^*$,

Risky reversed: $\hat{a}(y_i^+) \neq a_i^*$, $\hat{a}(y_i^-) = a_i^*$.

A risky reversed pair places an incorrect trajectory on the chosen side and a correct trajectory on the rejected side. The corresponding rate is

$$\text{ReversalRate} = \frac{\#\{i : \hat{a}(y_i^+) \neq a_i^* \wedge \hat{a}(y_i^-) = a_i^*\}}{N_{\text{parsed}}}, \quad (13)$$

where N_{parsed} is the number of successfully parsed preference pairs. Gold answers are used only for this post-hoc pair-quality analysis and are not accessed when constructing preferences for unlabeled problems. Chosen and rejected response lengths are measured directly in characters.

Deterministic rejected-response diagnostics. The final two rows of Table 3—*incomplete or truncated rejected responses* and *rejected responses with obvious repetition*—are computed using a fixed deterministic text-analysis protocol. The same rules are applied to the rejected responses produced by ScPO and Cloud–ScPO in every dataset–backbone setting.

A rejected response is labeled *incomplete or truncated* when at least one of the following conditions is detected:

- the response is empty or contains no substantive generated content;
- the expected final-answer marker is present but its answer is missing or unfinished;
- the response contains an unclosed parenthesis, bracket, brace, mathematical environment, or boxed-answer expression; or
- the response ends with a visibly incomplete sentence, equation, calculation, or reasoning step.

A rejected response is labeled as containing *obvious repetition* when at least one of the following patterns is detected:

- an identical normalized sentence occurs multiple times;
- an identical non-empty line or paragraph is repeated;
- a question–answer block or reasoning segment is reproduced without meaningful progression; or
- a normalized sequence of ten consecutive tokens recurs within the same response.

Normalization for repetition detection removes inconsequential whitespace differences before comparing textual units. Ordinary reuse of mathematical variables, short function words, or necessary intermediate expressions is not treated as repetition. The two labels are assigned independently, so a rejected response may satisfy both diagnostic conditions.

Aggregation and interpretation. For a method with N evaluated rejected responses, the reported rate for diagnostic d is

$$\text{Rate}_d = \frac{\#\{i : d(y_i^-) = 1\}}{N}, \quad (14)$$

where y_i^- denotes the rejected response in the i -th preference pair. Because these diagnostics characterize undesirable properties of the rejected side, higher values do not imply that the underlying model generates more defective trajectories. Instead, they indicate that the preference-construction method more frequently assigns incomplete, truncated, or repetitive trajectories to rejection rather than selecting them as preferred responses.

Across all four dataset–backbone settings, Cloud–ScPO produces higher rates for both diagnostics. This result supports the conclusion that Cloud scoring provides clearer

response-level separation by placing a larger proportion of structurally degraded trajectories on the rejected side.