Neuron-Aware Data Selection for Annotation-Free LLM Self-Distillation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Post-training large language models (LLMs) without real-world interaction feedback or human-labeled supervision remains challenging, particularly in specialized domains where expert annotations are costly to obtain. Recent annotation-free self-evolution methods address this by using the model's own outputs as supervision signals, constructing a teacher via additional context and aggregating predictions across multiple rollouts through majority voting to produce pseudo-labels. However, these approaches are not without drawbacks: SFT- and GRPO-based variants suffer out-of-domain performance degradation, while reward-based on-policy RL inflates calibration error. In this paper, we propose NEURON ON-POLICY SELF-DISTILLATION (NEURON-OPSD), a data-centric framework for annotation-free self-distillation that leverages internal neuron activations to guide both training-data selection and teacher context construction. The model is then trained via on-policy distillation from the teacher distribution, requiring no ground-truth labels at any stage. Across specialized-domain benchmarks, NEURON-OPSD improves in-domain task performance while preserving cross-domain generalization and mitigating calibration collapse over prior annotation-free baselines. This framework is particularly relevant to settings where online interaction or external supervision is costly or infeasible, and is conceptually distinct from offline RL approaches that rely on logged, reward-labeled trajectories.

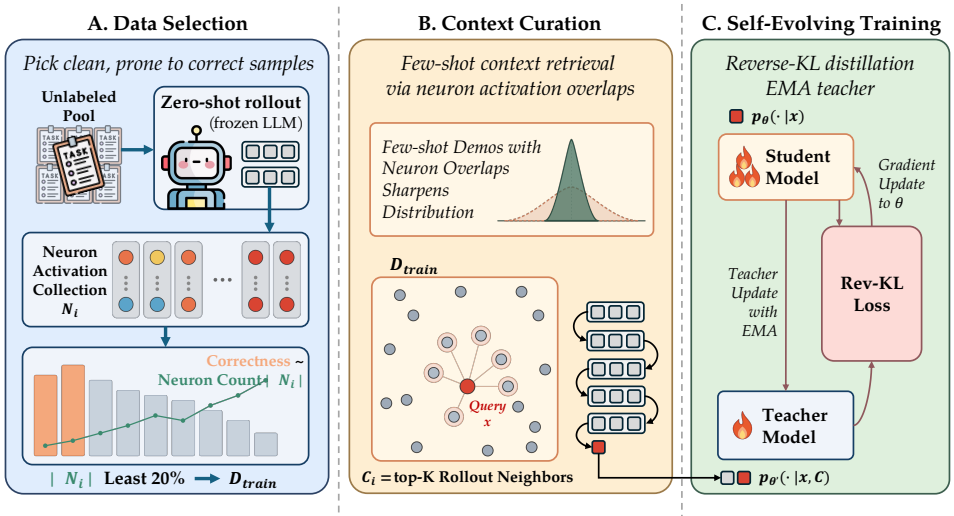

Figure 1: Overview of the Proposed Neuron-OPSD.

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

1 Introduction

Recent advances in Large language models (LLMs) have demonstrated remarkable performances in general tasks, however, adapting them to highly specialized domains such as education, law, or STEM subjects, remains difficult, which mainly due to the scarcity of human expert annotations.

To mitigate the scarcity of expert annotations, recent work has explored annotation-free self-training, where the model improves itself using only unlabeled data and signals derived from the model itself. SFT-based methods such as LMSI fine-tune the model on its own generated responses or rationales [Huang et al., 2023]. GRPO-based methods instead construct scalar rewards from the model's own outputs. TTRL extracts pseudo-labels from majority voting over multiple rollouts and optimizes them with the GRPO algorithm [Zuo et al., 2025], while Intuitor uses entropy-based confidence signals as rewards [Zhao et al., 2025b]. Despite their empirical successes, these annotation-free paradigms still face fundamental limitations. SFT-based self-training can suffer from catastrophic forgetting, especially for out-of domain tasks, while GRPO-based optimization faces challenges such as vanishing gradients under low within-group reward variance and entropy collapse during prolonged training. More fundamentally, GRPO collapses each rollout into a scalar reward, yielding only trajectory-level supervision and offering no guidance on which part of the reasoning should be reinforced or revised.

On-policy distillation (OPD) offers finer supervision by training the student model against a teacher model's token-level predictive distribution rather than a hard final-answer reward. Such logit-level targets densify supervision along the generation trajectory and circumvent the learning collapse caused by low within-group reward variances [Song and Zheng, 2026]. However, OPD's effectiveness relies on the quality of the teacher distribution, which is typically derived from stronger models, verifiers, or curated demonstrations. When such external signals are unavailable, the teacher must be elicited from the model itself through carefully additional information in prompts, i.e contexts around the query. This reframes the problem from designing self-generated rewards alone to deciding which unlabeled samples to train on and which contexts induce a useful teacher–student gap. Crucially, this self-improvement process requires rigorous data selection. Because the model acts as its own teacher, indiscriminately distilling from self-generated signals risks reinforcing its existing miscalibration, spurious reasoning, and hallucinations.

While traditional selection methods rely on surface-level metrics like majority-voting or entropy to filter out such unreliable data, these signals often prove insufficient. Recent work has pivoted towards LLMs' internal dynamics, demonstrating that internal states, particularly neuron activations, exhibit a strong correlation with when a model is genuinely uncertain or hallucinating [Gao et al., 2024, Nostalgebraist, 2020, Chen et al., 2025a, 2026].

Motivated by these observations, we propose NEURON-OPSD (N-OPSD), an annotation-free OPD pipeline for LLM self-improvement. In this work, we focus on a two-fold data construction process that determines whether self-distillation provides a useful signal: which unlabeled samples should be used for training, and which few-shot contexts should be used to induce the teacher distribution.

As illustrated in Figure 1, NEURON-OPSD utilizes two neuron-based mechanisms. First, NEURON CONSENSUS analyzes the reliability of candidate self-training samples through activation statistics. We measure neuron consensus using the number of activated neurons, finding that this signal is strongly associated with hallucinated or unreliable behaviors of LLMs. However, our results show that consensus alone is not a complete selection criterion. Selecting samples solely by activation count does not consistently yield stronger self-distillation, suggesting that training utility depends on both the reliability of the self-generated signal and the room for teacher-induced sharpening. This suggests that useful training data must be reliable enough to avoid reinforcing hallucinations, yet uncertain enough to leave room for LLMs improvement. Second, NEURON-OPSD employs NEURON OVERLAP for the OPD teacher-context construction. For each query, we retrieve the examples that show the most similar reasoning to the query as demonstrations, which are measured using active neuron representations. Since the teacher is the same model conditioned on an augmented context, these retrieved demonstrations directly shape the token-level distribution used for distillation. Neuron-overlap retrieval, therefore serves as a label-free method to elicit a more informative teacher distribution and create a meaningful teacher-student gap. Together, these two components demonstrate how to effectively leverage internal dynamics for annotation-free OPD. NEURON CONSENSUS serves primarily as a diagnostic signal to identify which samples are reliable or improvable, while NEURON

OVERLAP provides a practical mechanism for constructing robust teacher contexts. This makes NEURON-OPSD a practical, data-centric pipeline focused strictly on selecting the optimal data and contexts to form high-quality supervision.

We conduct experiments across three datasets covering six source domains. Results show that NEURON-OPSD achieves competitive self-improvement while better preserving cross-domain generalization and calibration compared with previous baselines. Further analysis suggests that its gains are driven by both the LLM's initial capability on the training dataset and the teacher-student distribution gap induced by the contexts. Ablation studies on four SciKnowEval domains further show that neuron consensus data selection is informative but insufficient as a standalone rule, while neuron-overlap retrieval provides an effective context-curation mechanism.

Our core contributions are as follows:

- We utilize NEURON CONSENSUS as a label-free signal for hallucination and data selection. Activation counts strongly correlate with hallucinated behavior, but selecting neither low- nor high-activation data consistently gives the best self-distillation results. This shows that activation count is informative, but insufficient as the only data-selection rule.
- We propose NEURON OVERLAP for OPD context curation. It retrieves few-shot demonstrations with similar active-neuron patterns, inducing a more informative teacher distribution and a useful teacher-student gap for annotation-free OPD.
- We integrate these two signals into NEURON-OPSD. Experiments across specialized-domain benchmarks show that NEURON-OPSD improves self-distillation while better preserving cross-domain performance and reducing calibration collapse.

2 Related Work

Annotation-free LLM Self-improvement. Recent work studies whether LLMs can improve without human labels by deriving supervision from their own outputs or intrinsic signals. Early self-improvement methods bootstrap rationales or self-generated feedback [Zelikman et al., 2022, Yuan et al., 2024, Huang et al., 2023], while recent annotation-free RL methods construct rewards from rollout agreement, entropy, or self-certainty [Zuo et al., 2025, Zhao et al., 2025b]. Other lines explore self-play or zero-data training regimes [Zhao et al., 2025a]. However, recent analysis of unsupervised RLVR shows that intrinsic rewards tend to sharpen the model's initial distribution and can collapse when confidence is misaligned with correctness [He et al., 2026]. Our work follows this motivation but replaces hard output-level rewards with OPD, using neuron-derived signals to select data and construct teacher contexts without external annotations.

LLM Probing through Internal Neuron Analysis. Internal-state analysis provides a way to inspect model behavior beyond surface outputs. Early probing methods such as the Logit Lens [Nostalgebraist, 2020] and Tuned Lens [Belrose et al., 2023] map intermediate representations into vocabulary space to study how predictions emerge across layers. More recent mechanistic work uses Sparse Autoencoders to disentangle superposed features and obtain more interpretable activation patterns [Gao et al., 2024]. Beyond interpretation, internal activations have also been linked to model reliability. Neuron agreement can signal when models are likely to be correct or hallucinate [Chen et al., 2025b, Cao et al., 2025b], and mechanism-interpretable metrics have been proposed to evaluate utility beyond surface accuracy [Cao et al., 2025a]. Recent work has demonstrated a successful application of neuron-based signals to active few-shot learning, where neuron representations guide the selection of informative examples for annotation, improving few-shot performance with only a small set of high-quality annotated demonstrations Chen et al. [2026]. Our work builds on these findings, using neuron activations not only for analysis but also as label-free signals for data selection and context construction in annotation-free self-improvement.

On-Policy Distillation for LLMs. OPD trains a student on its own generated trajectories using token-level supervision from a teacher distribution [Ye et al., 2026, Song and Zheng, 2026]. This differs from standard off-policy distillation, where the student learns from fixed teacher-generated data, and from outcome-reward RL, where supervision is usually a scalar reward. Recent work has studied OPD for LLM post-training and self-distillation under settings with stronger teachers, privileged information, or context-conditioned teachers [Hübotter et al., 2026, Zhao et al., 2026, Li et al., 2026]. Our work take OPD as the training objective, and focuses on the annotation-free setting where the teacher context and training data must be constructed without external labels.

3 Preliminaries

To establish the foundation for N-OPSD, we formalize the annotation-free self-evolution setting and the mechanics of OPD [Ye et al., 2026, Hübotter et al., 2026].

3.1 Problem Formulation

We consider an annotation-free self-evolution scenario. Given a base LLM with parameters θ and a domain-specific unlabeled dataset $\mathcal{D} = \{x_i\}_{i=1}^{N}$ of input queries, the objective is to improve the base model's performance on the target domain without access to ground-truth labels or any external model, i.e a different stronger teacher model. We frame this as annotation-free post-training from a fixed unlabeled prompt pool: the only source of new signal during training is the model's own on-policy rollouts and intrinsic activation patterns. No environment interaction, human feedback, or external oracle is available. This is distinct from offline RL setups, which learn from logged reward-labeled trajectories, and from online RL post-training such as RLHF, which assumes access to live preference or reward queries.

3.2 On-Policy Distillation

On-Policy Distillation, [Song and Zheng, 2026] trains the base or student model π_θ by minimizing a per-token reverse KL loss to a teacher model π_{teacher} on trajectories sampled from the student model:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[\sum_{t=1}^{|y|} \mathrm{KL}\big(\pi_\theta(\cdot \mid x, y_{<t}) \,\big\|\, \pi_{\text{teacher}}(\cdot \mid x, y_{<t})\big)\right], \tag{1}$$

The effectiveness of the student model largely depends on the quality of π_{teacher}, which is typically a larger model or the same model augmented with additional context, e.g., conditioning on external feedback like compiler traces [Hübotter et al., 2026]. In our annotation-free setting, both teacher and student model are initialized from the same base LLM and no external feedback is available, so constructing a reliable and high-quality π_{teacher} becomes the key challenge. To address this, we condition the teacher model on auxiliary context c constructed from unlabeled domain data alone, without any additional information, improving its next-token distribution on the target task, i.e., $\pi_{\text{teacher}}(\cdot \mid x, y_{<t}) := \pi_\theta(\cdot \mid x, c, y_{<t})$.

3.3 Internal Signals for Data Selection

The additional context c for the teacher model is constructed primarily through few-shot examples with the model's own generated solutions. Rather than using external supervision to verify the correctness of these pseudolabels, we probe the model's internal dynamics as a proxy for correctness. Now we introduce the calculation for the active neuron set.

Internal Neuron Signal Extraction. Building on the logit-lens view that intermediate representations can be read through the LM head [Nostalgebraist, 2020], and recent evidence that neuron-level activation features reflect response reliability [Cao et al., 2025b, Chen et al., 2025a, 2026], we follow Chen et al. [2026] to extract the activated MLP neurons detailed as follows. For layer l, let $\mathbf{h}^l$ be the MLP input, $\mathbf{W}^l_{\text{in}}$ and $\mathbf{W}^l_{\text{out}}$ be the up- and down-projection matrices. We compute the activation as:

$$\mathbf{k}^l = \sigma(\mathbf{h}^l \mathbf{W}^l_{\text{in}}), \tag{2}$$

Next, given a generated token $\hat{y}$, we score each neuron at layer l and index i by projecting its downstream contribution to the unembedding space:

$$S^l_{\hat{y},i} = k^l_i \cdot (\mathbf{w}^l_{\text{out},i} \cdot \mathbf{e}_{\hat{y}}), \tag{3}$$

where $\mathbf{w}^l_{\text{out},i}$ is the i-th row of $\mathbf{W}^l_{\text{out}}$ and $\mathbf{e}_{\hat{y}}$ is the unembedding vector for $\hat{y}$. This early-unembedding score filters for neurons that actively promote the model's generated token. For each generated token y_i, we define the activated neurons by keeping the top-K[1] scored neurons across all layers. We then aggregate the retained neurons across the generated tokens in the response by taking their union set. The resulting sparse set, denoted as $\mathcal{N}(x)$, serves as the internal signal used later for consensus estimation and context retrieval.

[1]We follow Chen et al. [2026] to set K as 5000.

4 Methodology

We present a neuron-activation-guided framework for selecting training data for self-distillation and curate contexts for self-evolving LLMs. Given an unlabeled data pool $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$, our approach selects a subset $\mathcal{S} \subset \mathcal{D}$ for self-distillation training, without requiring ground-truth labels. The framework consists of four stages: neuron activation extraction, sample selection based on neuron-consensus, context curation via Jaccard-nearest few-shot retrieval, and self-evolving training.

4.1 Neuron Activation Extraction

We instantiate the internal signal defined in §3.3 for every sample in the unlabeled pool. Concretely, we collect rollouts with the model in zero-shot, compute early-unembedding contribution scores for MLP neurons, and store the set of activated neurons $\mathcal{N}(x)$ as the sample's sparse activation pattern.

4.2 Sample Selection via Neuron Consensus

Given the activation pattern $\mathcal{N}(x)$, we measure neuron consensus of x by the size of its activation set:

$$s(x) = |\mathcal{N}(x)|, \tag{4}$$

where a smaller $s(x)$ indicates higher consensus, and a larger $s(x)$ indicates diffuse activation that could related to hallucinations Chen et al. [2025b, 2026]. We interpret $s(x)$ as an annotation-free proxy for the model's sample-level reliability. Since samples with lower $s(x)$ tend to be answered more correctly, applying OPD-based self-improvement to these samples helps reinforce the model's already-correct reasoning and predictions, helping to avoid exacerbating hallucinations or self-confirmation bias.

4.3 Neuron Overlap for Teacher Context Construction

To construct relevant few-shot contexts for the self-distillation teacher, we measure the overlap between neuron activation patterns using the Jaccard distance:

$$J(x_i, x_j) = 1 - \frac{|\mathcal{N}(x_i) \cap \mathcal{N}(x_j)|}{|\mathcal{N}(x_i) \cup \mathcal{N}(x_j)|}, \tag{5}$$

A small Jaccard distance indicates that the model processes two samples using similar neurons, suggesting they share similar knowledge circuits or go over similar reasoning path. For each query sample x_q, we construct its K-shot demonstration set by selecting the K nearest neighbors by:

$$\mathcal{C}_K(x_q) = \underset{\substack{S \subset \mathcal{D} \setminus \{x_q\} \\ |S| = K}}{\arg\min} \sum_{x_j \in S} J(x_q, x_j), \tag{6}$$

This selection ensures that few-shot examples are processed through similar neural pathways as the query, providing the teacher with the most relevant contextual knowledge for distillation, helping sharpening its prediction distribution.

4.4 Self-Improvement via On-Policy Distillation

To enable annotation-free self-improvement, we integrate neuron-guided sample selection and context curation with OPD. The student is the zero-shot policy parameterized by the current model θ, while the teacher is an EMA-updated copy $\bar{\theta}$ conditioned on the additional context $\mathcal{C}_K(x)$:

$$\pi_\theta^{\mathrm{stu}}(\cdot \mid x, y_{<t}) := \pi_\theta(\cdot \mid x, y_{<t}), \tag{7}$$

$$q_{\bar{\theta}}^{\mathrm{tea}}(\cdot \mid \mathcal{C}_K(x), x, y_{<t}) := \pi_{\bar{\theta}}(\cdot \mid \mathcal{C}_K(x), x, y_{<t}), \tag{8}$$

$$\bar{\theta} \leftarrow (1 - \tau)\, \bar{\theta} + \tau \theta, \tag{9}$$

where $\mathcal{C}_K(x)$ denotes the additional context guided by neuron-overlap retrieved for sample x. During optimization, the teacher distribution is treated as a fixed target, and gradients are applied only to the student policy $\pi_\theta^{\mathrm{stu}}$. The teacher is updated by exponential moving average (EMA) rate τ.

The OPD training objective minimizes the reverse KL divergence on student-generated rollouts:

$$\mathcal{L}_{\mathrm{OPD}} = \mathbb{E}_{x\sim\mathcal{S},\, y_{<t}\sim\pi_\theta^{\mathrm{stu}}}\left[\sum_{t=1}^{T} D_{\mathrm{KL}}\big(\pi_\theta^{\mathrm{stu}}(\cdot \mid x, y_{<t}) \,\|\, q_{\bar{\theta}}^{\mathrm{tea}}(\cdot \mid c_x, x, y_{<t})\big)\right]. \tag{10}$$

where $\mathcal{S}$ denotes the selected subset, and $c_x = \mathcal{C}_K(x)$ is the neuron-overlap context retrieved for sample x. During optimization, the teacher distribution is treated as a fixed target, and gradients are applied only to the student policy. The teacher parameters $\bar{\theta}$ are updated by exponential moving average of the student parameters, yielding a temporally smoothed model updates.

This objective transfers the distributional benefits of neuron-aligned few-shot contexts into the zero-shot student policy, internalizing context-induced distributions through parameter updates and eliminating reliance on demonstrations at inference. Crucially, the entire pipeline is annotation-free: neuron activations guide both sample selection and context curation, while N-OPSD uses the model's own context-conditioned predictions as supervision. Algorithm 1 summarizes the full procedure.

5 Experiments

To establish the impact of each proposed component, we first analyze the effectiveness of each individually, namely, neuron consensus, neuron overlap, and the OPD training objective when combined with neuron consensus and neuron overlap. For the analyses in Section 5.1 and Section 5.2, we use **SciKnowEval** [Feng et al., 2025], a multiple-choice benchmark spanning four scientific domains, *Biology*, *Material*, *Physics*, and *Chemistry*, each split into 80% train and 20% test sets.

5.1 Analysis: Neuron Consensus based Data Selection

Does Neuron Consensus Correlate with Performance? We validate the relationship between the activation density, i.e., the total count of activated neurons $|\mathcal{N}(x)|$, and the model's hallucination rates. As illustrated in Figure 2, our analysis across all four SciKnowEval domains reveals a strong, robust monotonic correlation: queries that activate a larger number of neurons are significantly more likely to result in incorrect rollouts.

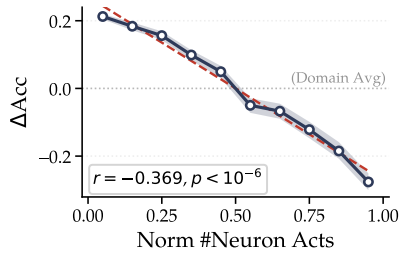

Figure 2: Neuron consensus correlates accuracy. ΔAcc is the gap between average domain accuracy and bin accuracy, while #Neuron Acts is normalized domain-wise. Separate domain-wise relational results shown in Figure 3.

This phenomenon is attributed to Neuron Consensus. Sparser activations with lower $|\mathcal{N}(x)|$ indicate a higher degree of consensus within the network. The model effortlessly retrieves a clear, unconflicted knowledge trace. Conversely, a denser, widespread activation pattern suggests internal conflict and uncertainty, where the model struggles to reconcile disparate neural pathways. This finding fundamentally addresses our data selection bottleneck: by isolating queries with the lowest activation counts, we can filter out hallucination-prone samples and secure pristine targets for self-supervision, even without external labels.

Are Easy Samples Better Training Data for OPD Training? We rank the source pool by neuron count $\mathrm{s}(\mathrm{x})$ and train on the subset that ranked top and bottom 20% following the OPD framework. For evaluation, we sample 8 rollouts for each test query and report two metrics: **Avg@8**, the mean per-sample accuracy; and **ECE**, the expected calibration error over the per-query majority-vote confidence with 15 equal-width bins, lower the better. Table. 1 reports the results on SciKnowEval.

Intuitively, subsets with higher activation numbers, i.e. Top-20%, will generally noisier than the subsets with lower activation numbers, i.e., Bottom-20%. As lower activation numbers indicates a lower hallucination rates. However, in training ablations, with OPD, Top-20% can still have reasonable improvements on some domains such as Bio. and Mat., also with a competitive calibration.

Furthermore, a trade-off exists between learning utility and noise levels. Specifically, samples prone to hallucination theoretically offer higher learning value since they represent areas where the LLM consistently fails. However, in self-improvement scenarios, the model struggles to generate reliable

Selection	Avg@8 (↑)				ECE (↓)			
	CHEM.	BIO.	MAT.	PHYS.	CHEM.	BIO.	MAT.	PHYS.
Qwen3-4B	72.11	73.98	71.12	80.20	0.173	0.195	0.213	0.095
+Top-20%	68.46$_{-3.65}$	**74.59**$_{+0.61}$	**73.82**$_{+2.70}$	82.00$_{+1.80}$	**0.133**$_{-0.040}$	0.198$_{+0.003}$	**0.194**$_{-0.019}$	0.116$_{+0.021}$
+Bottom-20%	**72.03**$_{-0.08}$	74.02$_{+0.04}$	73.39$_{+2.27}$	**83.13**$_{+2.93}$	0.171$_{-0.002}$	**0.193**$_{-0.002}$	0.207$_{-0.006}$	**0.114**$_{+0.019}$

Table 1: Analysis results of consensus-based selection, reported in-domain Avg@8 and ECE.

supervision signals for these difficult cases. Conversely, while the LLM can easily produce correct self-generated signals for simpler samples, their contribution to overall improvement is marginal. Given this persistent trade-off, training on a refined top-20% subset can still yield substantial performance gains, even in the presence of noise.

5.2 Analysis: Neuron Overlap guided Context Curation

In this experiment, we fix the Bottom-20% as OPD training data, we compare neuron-overlap retrieval with uniform in-domain random retrieval. As shown in Table. 2, NEURON-JACCARD improves over the base model on three of the four SciKnowEval sources and outperforms random retrieval on CHEM. and PHYS.. Random in-domain retrieval is also competitive on BIO. and MAT., indicating that in-domain demonstrations already provide a strong context prior.

We further analyze why the marginal gain over random varies across domains. Because both random and neuron-based retrieval are restricted to the same source domain, the difference depends on how discriminative the within-domain contexts are. We measure reasoning diversity by the mean pairwise neuron-Jaccard distance between training samples: CHEM. 0.870, BIO. 0.634, MAT. 0.628, and PHYS. 0.640. In more diverse domains, such as CHEM., neuron-overlap retrieval can select demonstrations that match the query-specific reasoning mode, yielding a clear advantage over random selection. In more homogeneous domains, many examples activate similar neuron sets, so nearest-neighbor and random in-domain contexts become partially redundant, reducing the marginal benefit of Jaccard retrieval. This explains why random retrieval remains competitive on BIO. and MAT., while also showing that NEURON OVERLAP is most useful when neuron similarity can distinguish meaningful reasoning modes within the source pool.

Retriever	CHEM.	BIO.	MAT.	PHYS.
Qwen3-4B-Thinking	72.11	73.98	71.12	80.20
+Random	68.48$_{-3.63}$	**74.14**$_{+0.16}$	**73.79**$_{+2.67}$	81.97$_{+1.77}$
+Neuron-Jaccard (ours)	**72.03**$_{-0.08}$	74.02$_{+0.04}$	73.39$_{+2.27}$	**83.13**$_{+2.93}$

Table 2: Analysis results of the few-shot retriever, reported in-domain Avg@8 on SciKnowEval.

5.3 Experimental Setup

We then perform the evaluations on various self-improvement training algorithms. As we didn't observe a significant performance and calibration difference of using the subsets with the bottom 20% vs the top 20% #neuron activations, therefore we demonstrated the bottom 20% results.

Datasets. We evaluate on three different datasets. **SciKnowEval** [Feng et al., 2025] contains four scientific multiple-choice domains: *Biology*, *Material*, *Physics*, and *Chemistry*, each split 80/20 for training and testing. **Edu-Feedback** [Wu and Schunn, 2023] is a binary feedback-quality classification dataset with 1,799 training and 1,000 testing samples. **MMLU-Pro** [Wang et al., 2024] is a challenging multi-domain multiple-choice benchmark. We split 80% data for self-improvement.

Baselines and Models. We compare with both the off-policy and on-policy self-improvement baselines. Including SFT-based **LMSI** [Huang et al., 2023], on-policy methods **TTRL** [Zuo et al., 2025], and **Intuitor** [Zhao et al., 2025b], an annotation-free RL method using self-certainty as reward. We evaluate on *Qwen3-4B-Thinking-2507* [Yang et al., 2025].

Metrics. We sample 8 rollouts for each test query and report three metrics: **Avg@8**, the mean per-sample accuracy; **Maj@8**, the majority-vote accuracy; and **ECE**, the expected calibration error over the per-query majority-vote confidence with 15 equal-width bins, lower the better.

5.4 Performance Results

To more comprehensively evaluate the efficacy of the self-improving baselines and the proposed method, we focus on both in-domain and cross-domain performance gain and calibration variances. Specifically, we train each method on a single source domain and evaluate on all target test sets. Table. 3 and Table. 4 report Accuracy and ECE on Qwen3-4B-Thinking-2507.

Method	Eval.	SciKnowEval					Edu.	MMLU-Pro	Avg.
		BIO	MAT.	PHYS.	CHEM	*Avg.*			
Qwen3-4B base		73.98	71.12	80.20	72.11	74.35	64.95	72.16	72.42
LMSI	in	74.59$_{+0.61}$	69.08$_{-2.04}$	78.54$_{-1.66}$	73.72$_{+1.61}$	73.98$_{-0.37}$	66.95$_{+2.00}$	68.60$_{-3.56}$	71.91$_{-0.51}$
	cross	68.57$_{-3.54}$	65.48$_{-7.20}$	67.02$_{-3.84}$	71.19$_{-1.29}$	68.06$_{-3.97}$	58.14$_{-15.77}$	73.46$_{+0.99}$	67.31$_{-5.11}$
TTRL	in	75.00$_{+1.02}$	73.26$_{+2.14}$	82.98$_{+2.78}$	72.43$_{+0.32}$	75.92$_{+1.56}$	64.95$_{+0.00}$	72.04$_{-0.12}$	73.44$_{+1.02}$
	cross	**73.31**$_{+1.20}$	**73.32**$_{+0.64}$	71.28$_{+0.41}$	73.46$_{+0.97}$	**72.84**$_{+0.81}$	74.47$_{+0.55}$	**73.76**$_{+1.29}$	73.27$_{+0.84}$
Intuitor[2]	in	**76.21**$_{+2.23}$	**73.41**$_{+2.29}$	**83.43**$_{+3.23}$	**72.63**$_{+0.52}$	**76.42**$_{+2.07}$	—	71.91$_{-0.25}$	75.52$_{+1.60}$
	cross	69.50$_{-2.61}$	72.79$_{+0.11}$	68.33$_{-2.53}$	70.76$_{-1.72}$	70.35$_{-1.69}$	—	73.30$_{+0.83}$	70.94$_{-1.18}$
N-OPSD	in	74.02$_{+0.04}$	73.39$_{+2.27}$	83.13$_{+2.93}$	72.03$_{-0.08}$	75.64$_{+1.29}$	**72.19**$_{+7.24}$	72.04$_{-0.12}$	**74.47**$_{+2.05}$
	cross	72.71$_{+0.61}$	73.27$_{+0.59}$	**71.37**$_{+0.51}$	**73.59**$_{+1.11}$	72.74$_{+0.70}$	**75.22**$_{+1.31}$	73.74$_{+1.27}$	**73.32**$_{+0.90}$

Table 3: Cross-domain evaluation on Qwen3-4B-Thinking-2507, Avg@8. For each method, *in* reports the score of the model trained and evaluated on the same source domain, while *cross* reports the average score of the same source-trained model on the other target domains. Subscripts report Δ against the untrained Qwen3-4B baseline computed on the same target-domain support. The full source-target results are provided in Table 7.

Method	Eval.	SciKnowEval					Edu.	MMLU-Pro	Avg.
		BIO	MAT.	PHYS.	CHEM	*Avg.*			
Qwen3-4B base		0.195	0.213	0.095	0.173	0.169	0.246	0.184	0.184
LMSI	in	0.202$_{+0.007}$	0.225$_{+0.012}$	**0.111**$_{+0.016}$	0.179$_{+0.006}$	0.179$_{+0.010}$	0.255$_{+0.009}$	**0.171**$_{-0.013}$	0.191$_{+0.007}$
	cross	**0.171**$_{-0.011}$	0.179$_{+0.001}$	0.202$_{+0.000}$	**0.182**$_{-0.004}$	**0.184**$_{-0.004}$	**0.095**$_{-0.077}$	**0.162**$_{-0.022}$	**0.165**$_{-0.019}$
TTRL	in	0.204$_{+0.009}$	0.208$_{-0.005}$	0.123$_{+0.028}$	0.192$_{+0.019}$	0.182$_{+0.013}$	0.248$_{+0.002}$	0.198$_{+0.014}$	0.196$_{+0.012}$
	cross	0.186$_{+0.004}$	0.187$_{+0.009}$	0.204$_{+0.002}$	0.199$_{+0.012}$	0.194$_{+0.007}$	0.178$_{+0.006}$	0.190$_{+0.006}$	0.191$_{+0.007}$
Intuitor	in	0.208$_{+0.013}$	0.212$_{-0.001}$	0.117$_{+0.022}$	0.198$_{+0.026}$	0.184$_{+0.015}$	—	0.174$_{-0.010}$	0.182$_{+0.010}$
	cross	0.226$_{+0.044}$	0.187$_{+0.008}$	0.234$_{+0.032}$	0.224$_{+0.038}$	0.218$_{+0.030}$	—	0.185$_{+0.000}$	0.211$_{+0.024}$
N-OPSD	in	**0.193**$_{-0.003}$	**0.207**$_{-0.006}$	0.114$_{+0.019}$	**0.171**$_{-0.002}$	**0.171**$_{+0.002}$	**0.191**$_{-0.056}$	0.180$_{-0.004}$	**0.176**$_{-0.008}$
	cross	0.177$_{-0.005}$	**0.177**$_{-0.001}$	**0.194**$_{-0.008}$	0.187$_{+0.000}$	0.184$_{-0.004}$	0.177$_{+0.005}$	0.186$_{+0.002}$	0.183$_{-0.001}$

Table 4: Cross-domain calibration on Qwen3-4B-Thinking-2507, reported as ECE, where lower is better, computed from per-query majority-vote confidence. For each method, *in* reports the ECE of the model trained and evaluated on the same source domain, while *cross* reports the average ECE of the same source-trained model on the other target domains. Subscripts report Δ against the untrained Qwen3-4B baseline computed on the same target-domain support, where negative values indicate improved calibration. The full results are provided in Table 8.

Performance. N-OPSD improves source-domain performance on four of the six training sources in Table. 3, with especially clear gains on MAT., PHYS., and EDU. On MAT., PHYS., and EDU., N-OPSD matches or exceeds TTRL's source-domain gain, and its averaged row attains the highest overall Avg@8 among the compared methods. The main advantage is not uniform source-domain dominance, but a stronger accuracy-preservation trade-off: every SciKnow-trained N-OPSD model improves EDU. over the base model, whereas TTRL is mixed. LMSI and Intuitor often suffer large drops. Thus, N-OPSD delivers source-domain improvement where the teacher provides useful sharpening while better preserving non-source capability.

Calibration. Table. 4 shows that TTRL and Intuitor generally increase average ECE, while N-OPSD keeps the calibration cost smaller and slightly reduces overall ECE in the averaged row. The strongest calibration effect appears on EDU.: every N-OPSD row lowers EDU. ECE relative to the base model, whereas TTRL is mixed and Intuitor often worsens it substantially. Several N-OPSD rows simultaneously improve overall Avg@8 and reduce overall ECE, and N-OPSD$_{\text{EDU.}}$ gives the largest overall accuracy gain together with the lowest overall ECE among N-OPSD rows. These results show where the method works best, it improves accuracy without the broad calibration inflation seen in reward-based annotation-free RL, although it can still increase ECE on some SciKnowEval targets.

[2]The Intuitor training on the Edu. dataset yields no improvement across validation steps, thus we drop the results.

5.5 What drives the self-improvement, and when does it benefit most?

In terms of self-training, He et al. [2026] characterise intrinsic self-supervision methods as sharpening the model's prior distribution rather than adding new information, which succeeds only when that prior is already aligned with correctness. Our consensus stratification makes the trade concrete at the sample level, with high-consensus samples supplying a reliable prior to sharpen against and low-consensus samples supplying an unreliable one. Meanwhile, OPD derives its training signal from a soft teacher-student distributional gap rather than from hard scalar rewards. This allows part of the teacher's uncertainty to be preserved during distillation, yielding a more favorable trade-off between calibration and correctness. Here, we conduct a data-centric analysis of the selected training pool to examine how data dynamics drive the gains of OPD-based LLM self-training.

Domain	Base Test Avg@8	Train Maj@8	Train Avg@8	Maj−Avg	in-domain gain
CHEM.	72.11	98.23	98.18	0.04	−0.08
BIO.	73.98	98.73	98.64	0.09	+0.04
MAT.	71.12	91.93	91.59	0.34	+2.27
PHYS.	80.20	96.31	96.27	0.04	+2.93

Table 5: Base rollout statistics on the training pool versus N-OPSD's in-domain Avg@8 gain. All statistics are computed with Qwen3-4B-Thinking-2507: Test Avg@8 is measured on the held-out test set, while training-pool statistics are measured on the bottom-20% subset by neuron count.

Domain	H_{zs} Student	H_{neur} Teacher	ΔH T−S	Avg@8 Gain
CHEM.	0.240	0.237	−0.002	−0.08
BIO.	0.238	0.240	+0.001	+0.04
MAT.	0.268	0.251	−0.017	+2.27
PHYS.	0.257	0.247	−0.010	+2.93

Table 6: Teacher-student per-token logprob entropy on the selected training pool. The teacher conditions on $K{=}10$ neuron-Jaccard nearest demonstrations, and the student is zero-shot. $\Delta H < 0$ means the teacher is sharper at the token level. The two domains with significant teacher sharpening, MAT. and PHYS., are also the two with the largest in-domain Avg@8 gain.

Sharpening Room on High-Consensus Samples. A necessary precondition for N-OPSD to deliver a meaningful in-domain gain is that the high neuron-consensus pool used for distillation still contains enough gap for sharpening, i.e., carries disagreement across the rollouts. Table. 5 reports the $n{=}8$ rollout stats of the untrained model on this pool. Across CHEM., BIO., and MAT., the Maj−Avg gap on the high-consensus training pool tracks N-OPSD's in-domain gain: larger residual rollout disagreement gives OPD distillation more uncertainty to resolve. However, PHYS. is already near-unanimous and still obtains a large gain, showing that vote-level sharpening room alone is not sufficient to explain all improvements.

Teacher-Student Gap Drives Gain. The sharpening-room analysis captures part of when N-OPSD has signal to exploit, but not whether the few-shot teacher provides a useful target. On the selected pool, vote-level metrics saturate, as both the zero-shot student and the neuron-Jaccard teacher reach $\sim$98% Avg@8 in Table. 5, so the teacher-student gap must be measured at the token level. Table. 6 reports per-token logprob entropy, averaged over the response trajectory and paired across queries. On MAT. and PHYS. the teacher meaningfully sharpens the token distribution, with $\Delta H = -0.017$ and -0.010, and these are exactly the two domains where N-OPSD delivers the strongest in-domain gains of $+2.27$ and $+2.93$. On BIO. and CHEM. the teacher provides no token-level sharpening, with $|\Delta H| < 0.003$ and no significant difference, and the in-domain gain collapses correspondingly.

6 Conclusion

In this work, we propose NEURON-OPSD for annotation-free post-training from a fixed unlabeled prompt pool, where no ground-truth annotations are available. The method uses Neuron Consensus to select more reliable self-improvement data and Neuron Overlap to construct neuron-aligned teacher contexts for on-policy distillation. These two components are proposed for the central challenge of unreliable self-supervision, which can otherwise amplify hallucinations, miscalibration, or negative transfer. Across our evaluations, NEURON-OPSD achieves competitive in-domain gains while better preserving cross-domain capability and calibration than prior annotation-free methods.

Limitations

There are several limitations remaining in this work. First, in-domain gains are not uniform across domains: BIO. and CHEM. show negligible improvement, consistent with Tab. 6 showing no teacher token-distribution sharpening on those domains. Second, although NEURON CONSENSUS correlates with hallucinated behavior, our top- and bottom-activation ablations show that activation count alone is insufficient for data selection. This poses a fundamental question: how do we balance self-improvement against signal reliability? Identifying the precise conditions under which LLMs can be trusted to self-supervise is left for future work. Third, NEURON OVERLAP's retrieval coherence depends on domain reasoning diversity; as the case study in App. C.2 shows, domains with homogeneous reasoning patterns can make neuron-Jaccard retrieval less discriminative and push contexts toward generic examples. Fourth, although we compare against uniform in-domain random few-shot retrieval, we do not compare against other alternative teacher-context construction strategies, such as retrieval-augmented or prompt-engineered contexts, that might produce similar teacher sharpening without neuron overlap. Also, the evaluation covers three datasets, SciKnowEval, Edu-Feedback, and MMLU-Pro, so the observed side-effect tradeoffs may change under broader domain coverage and different context-construction mechanisms.

References

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Yixin Cao, Jiahao Ying, Yaoning Wang, Xipeng Qiu, Xuanjing Huang, and Yugang Jiang. Model utility law: Evaluating llms beyond performance through mechanism interpretable metric, 2025a. URL https://arxiv.org/abs/2504.07440.

Yixin Cao, Jiahao Ying, Yaoning Wang, Xipeng Qiu, Xuanjing Huang, and Yugang Jiang. Model utility law: Evaluating LLMs beyond performance through mechanism interpretable metric, 2025b. URL http://arxiv.org/abs/2504.07440.

Kang Chen, Yaoning Wang, Kai Xiong, Zhuoka Feng, Wenhe Sun, Haotian Chen, and Yixin Cao. Do llms signal when they're right? evidence from neuron agreement. *arXiv preprint arXiv:2510.26277*, 2025a.

Kang Chen, Yaoning Wang, Kai Xiong, Zhuoka Feng, Wenhe Sun, Haotian Chen, and Yixin Cao. Do LLMs signal when they're right? evidence from neuron agreement, 2025b. URL http://arxiv.org/abs/2510.26277.

Zhuowei Chen, Liwei Chen, Christian Schunn, Raquel Coelho, and Xiang Lorraine Li. Neuron-aware active few-shot learning for LLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2026.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. Sciknoweval: Evaluating multi-level scientific knowledge of large language models, 2025. URL https://arxiv.org/abs/2406.09098.

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

Bingxiang He, Yuxin Zuo, Zeyuan Liu, Shangziqi Zhao, Zixuan Fu, Junlin Yang, Cheng Qian, Kaiyan Zhang, Yuchen Fan, Ganqu Cui, Xiusi Chen, Youbang Sun, Xingtai Lv, Xuekai Zhu, Li Sheng, Ran Li, Huan-ang Gao, Yuchen Zhang, Bowen Zhou, Zhiyuan Liu, and Ning Ding. How far can unsupervised RLVR scale LLM training?, 2026. URL http://arxiv.org/abs/2603.08660.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.67. URL https://aclanthology.org/2023.emnlp-main.67/.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation, 2026. URL http://arxiv.org/abs/2601.20802.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

Nostalgebraist. Interpreting GPT: The logit lens. https://www.lesswrong.com/posts/AcCRPn5w6xmtPRD7j/interpreting-gpt-the-logit-lens, 2020. LessWrong.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models, 2026. URL https://arxiv.org/abs/2604.00626.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

Yong Wu and Christian D Schunn. Passive, active, and constructive engagement with peer feedback: A revised model of learning from peer feedback. *Contemporary Educational Psychology*, 73: 102160, 2023.

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

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models, 2026. URL http://arxiv.org/abs/2602.12275.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning, 2022. URL http://arxiv.org/abs/2203.14465.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. *arXiv preprint arXiv:2505.03335*, 2025a.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models, 2026. URL http://arxiv.org/abs/2601.18734.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025b.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-time reinforcement learning, 2025. URL http://arxiv.org/abs/2504.16084.

A Implementation Details

Neuron contribution computation. We register forward hooks on the activation function of each transformer MLP layer, `model.layers[l].mlp.act_fn`. For each response token position t, we capture the post-activation hidden state $\mathbf{a}_{l,t} \in \mathbb{R}^{d_{\text{inter}}}$ and compute contribution scores via Eq. 3. We retain the top 2,000 neurons per layer per chunk, then apply global Top-K deduplication with $K = 5{,}000$ across all layers. When the same layer-neuron pair appears across multiple chunks, namely groups of response positions, we keep the maximum contribution score.

N-OPSD training configuration. All N-OPSD models are trained using the veRL framework with Ray-based FSDP distributed training across 4 GPUs, batch size 16, micro-batch size 4, and 150 training steps. We use reverse KL divergence and EMA teacher updates with rate 0.01.

TTRL training configuration. TTRL models follow the standard open-source setup of Zuo et al. [2025], using GRPO with majority-vote pseudo-labels. We train with 4 GPUs, batch size 16, actor learning rate 5×10^{-7} with cosine warmup and 3% warmup ratio, critic learning rate 9×10^{-6}, KL coefficient 0.0, rollout temperature 1.0, and 8 votes per prompt. Validation uses temperature 0.6 with 8 samples. Models are trained for 1 epoch with checkpoint selection based on the best validation accuracy, maj@8.

Data selection configuration. For the neuron-guided data selection used by N-OPSD, we apply the same rule across all domains: *bot20*, which retains the bottom-20% of the source pool by zero-shot sensitivity, where $s(x) = |\mathcal{N}_{\text{0-shot}}(x)|$ is the number of unique neurons activated without context. The selected pool is then paired with $K = 10$ Jaccard-nearest few-shot demonstrations during training; see §4.3. No ground-truth labels are used at any point during selection.

B Methodology

B.1 Algorithm

Algorithm 1: Neuron-OPSD

Input: Unlabeled pool $\mathcal{D}$; LLM π_θ; percentile b; shots K; training steps T; EMA rate ρ; learning rate η.
Output: Updated model π_θ.

`// Stage 1: neuron activation extraction (§4.1)`
foreach $x \in \mathcal{D}$ **do**
  Generate $y \sim \pi_\theta(\cdot \mid x)$ and score neurons via Eq. 3
  Form activation set $\mathcal{N}(x)$ and neuron count $s(x) = |\mathcal{N}(x)|$

`// Stage 2: high-consensus selection (§4.2)`
$\tau_b \leftarrow b$-th percentile of $\{s(x)\}_{x \in \mathcal{D}}$
$\mathcal{S} \leftarrow \{x \in \mathcal{D} \mid s(x) \le \tau_b\}$

`// Stage 3: context construction (§4.3)`
Precompute $J(x_i, x_j)$ for all pairs in $\mathcal{D}$ via Eq. 5
foreach $x \in \mathcal{S}$ **do**
  $\mathcal{C}_K(x) \leftarrow K$-nearest neighbors of x in $\mathcal{D} \setminus \{x\}$ under J

`// Stage 4: self-training (§4.4)`
Initialise EMA teacher $\theta' \leftarrow \theta$
for $t = 1, \ldots, T$ **do**
  Sample batch $B \subset \mathcal{S}$
  Generate student rollouts $y \sim \pi_\theta(\cdot \mid x)$ for $x \in B$
  Compute $\mathcal{L}_{\text{OPD}}$ between $\pi_\theta(\cdot \mid x)$ and $\pi_{\theta'}(\cdot \mid x, \mathcal{C}_K(x))$ via Eq. 10
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{OPD}}$
  $\theta' \leftarrow (1 - \rho)\, \theta' + \rho\, \theta$ `// EMA update`

return π_θ

C Experiments

C.1 Evaluation

All evaluations sample 8 responses per question with temperature 0.6 using vLLM. We report majority-vote accuracy, maj@8, and Expected Calibration Error (ECE), where the confidence for each question is $c = \text{maxcount}/8$.

C.2 Case study: neuron-Jaccard retrieval on CHEM and PHYS.

To make the reasoning-diversity claim concrete, we run the same $K{=}10$ neuron-Jaccard retriever on two different queries within each of CHEM and PHYS., and inspect the retrieved few-shot contexts.

The contrast is not in the *overlap* between the two queries' demo sets, which is near-zero given the large sample pool, but in the *internal coherence* of each query's retrieved set.

CHEM: retrieval locks onto query-specific reasoning modes. Query A asks about iodine-131 half-life calculation. Its retrieved set is dominated by radioactivity, isotope decay, molarity, mass spectrometry, and other per-sample formula-application chemistry tasks. Query B asks about catalytic oxidation depolymerisation. Its retrieved set contains ten demonstrations all concerning polymer breakdown: polymer recycling, controlled-radical-polymerisation reverse processes, dynamic-covalent gel degradation, silicone degradation, and lignin depolymerisation, among others. Each query's retrieved context is internally thematic and distinct from the other query's.

PHYS.: retrieval returns generic grab-bags regardless of query. Query A asks for the acceleration of a 15 kg object under a 10 N force. Its retrieved set is thematically unrelated: an ML-framework question, a stress-intensity-factor method, a heat-engine efficiency problem, a Material-Point-Method description, a specific-heat calculation, a quantum-heat-engine note, a diffuse-interface theory passage, and a PEDOT:PSS conductivity question. Query B asks about the Good Regulator Theorem. Its retrieved set is similarly scattered, spanning van der Waals force, ionizing radiation, cosmic-ray acceleration, resonance-frequency design, CMB temperature, magnetic materials, plasma electron density, and IR polarization. The two sets differ in specific questions but are structurally indistinguishable as generic physics expository or MCQ snippets.

Interpretation. In CHEM, samples traverse genuinely different reasoning paths, so neuron-level similarity can surface query-specific clusters. In PHYS., samples reuse largely the same reasoning neurons across distinct topics, so K-nearest neurons cannot differentiate queries and the retriever collapses toward generic contexts.

C.3 Relational Analysis of Neuron for Hallucination Detection

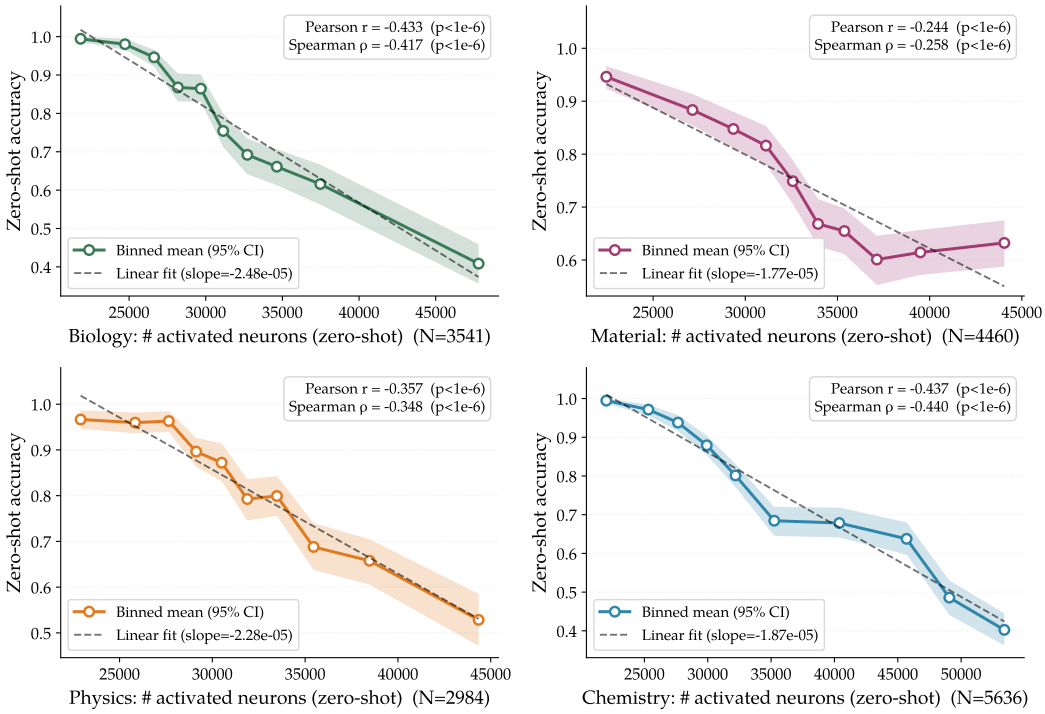

Figure 3: Relational regression results on SciKnowEval.

C.4 Additional Results

Model	SciKnowEval					Edu.	MMLU-Pro	Avg.
	BIO	MAT.	PHYS.	CHEM	*Avg.*			
Qwen3-4B	73.98	71.12	80.20	72.11	74.35	64.95	72.16	72.42
LMSI [Huang et al., 2023]								
LMSI_{BIO}	$74.59_{+0.61}$	$71.22_{+0.10}$	$78.43_{-1.77}$	$68.60_{-3.51}$	$73.21_{-1.14}$	$66.00_{+1.05}$	$58.59_{-13.57}$	$69.57_{-2.85}$
$\text{LMSI}_{\text{MAT.}}$	$72.75_{-1.23}$	$69.08_{-2.04}$	$73.80_{-6.40}$	$69.32_{-2.79}$	$71.23_{-3.12}$	$60.32_{-4.63}$	$51.19_{-20.97}$	$66.08_{-6.34}$
$\text{LMSI}_{\text{PHYS.}}$	$73.70_{-0.28}$	$72.30_{+1.18}$	$78.54_{-1.66}$	$71.61_{-0.50}$	$74.04_{-0.31}$	$59.49_{-5.46}$	$58.01_{-14.15}$	$68.94_{-3.48}$
$\text{LMSI}_{\text{CHEM}}$	$74.56_{+0.58}$	$72.20_{+1.08}$	$81.40_{+1.20}$	$73.72_{+1.61}$	$75.47_{+1.12}$	$61.79_{-3.16}$	$65.99_{-6.17}$	$71.61_{-0.81}$
$\text{LMSI}_{\text{EDU.}}$	$62.85_{-11.13}$	$58.44_{-12.68}$	$64.98_{-15.22}$	$55.06_{-17.05}$	$60.34_{-14.02}$	$66.95_{+2.00}$	$49.37_{-22.79}$	$59.61_{-12.81}$
$\text{LMSI}_{\text{MMLU}}$	$74.27_{+0.29}$	$72.63_{+1.51}$	$81.06_{+0.86}$	$71.51_{-0.60}$	$74.87_{+0.52}$	$67.84_{+2.89}$	$68.60_{-3.56}$	$72.65_{+0.23}$
$\text{LMSI}_{\text{Avg.}}$	$72.12_{-1.86}$	$69.31_{-1.81}$	$76.37_{-3.83}$	$68.30_{-3.81}$	$71.53_{-2.83}$	$63.73_{-1.22}$	$58.62_{-13.53}$	$68.08_{-4.34}$
TTRL [Zuo et al., 2025]								
TTRL_{BIO}	$75.00_{+1.02}$	$73.03_{+1.92}$	$83.28_{+3.09}$	$72.85_{+0.74}$	$76.04_{+1.69}$	$65.31_{+0.36}$	$72.08_{-0.08}$	$73.59_{+1.17}$
$\text{TTRL}_{\text{MAT.}}$	$74.21_{+0.22}$	$73.26_{+2.14}$	$83.25_{+3.05}$	$72.03_{-0.08}$	$75.69_{+1.33}$	$64.92_{-0.03}$	$72.20_{+0.04}$	$73.31_{+0.89}$
$\text{TTRL}_{\text{PHYS.}}$	$74.14_{+0.16}$	$72.61_{+1.49}$	$82.98_{+2.79}$	$72.61_{+0.50}$	$75.58_{+1.23}$	$65.09_{+0.14}$	$71.94_{-0.22}$	$73.23_{+0.81}$
$\text{TTRL}_{\text{CHEM}}$	$74.94_{+0.95}$	$72.81_{+1.69}$	$83.40_{+3.20}$	$72.43_{+0.32}$	$75.89_{+1.54}$	$64.21_{-0.74}$	$71.92_{-0.24}$	$73.28_{+0.86}$
$\text{TTRL}_{\text{EDU.}}$	$74.56_{+0.57}$	$72.88_{+1.76}$	$83.55_{+3.35}$	$71.91_{-0.20}$	$75.72_{+1.37}$	$64.95_{+0.00}$	$69.43_{-2.73}$	$72.88_{+0.46}$
$\text{TTRL}_{\text{MMLU}}$	$74.94_{+0.95}$	$72.88_{+1.76}$	$83.89_{+3.69}$	$72.17_{+0.06}$	$75.97_{+1.62}$	$64.91_{-0.04}$	$72.04_{-0.12}$	$73.47_{+1.05}$
$\text{TTRL}_{\text{Avg.}}$	$74.63_{+0.65}$	$72.91_{+1.79}$	$83.39_{+3.19}$	$72.33_{+0.22}$	$75.82_{+1.46}$	$64.90_{-0.05}$	$71.60_{-0.56}$	$73.29_{+0.87}$
Intuitor [Zhao et al., 2025b][3]								
$\text{Intuitor}_{\text{BIO}}$	$76.21_{+2.22}$	$72.10_{+0.98}$	$82.23_{+2.03}$	$71.31_{-0.80}$	$75.46_{+1.11}$	$55.70_{-9.25}$	$66.17_{-5.99}$	$70.62_{-1.80}$
$\text{Intuitor}_{\text{MAT.}}$	$73.98_{+0.00}$	$73.41_{+2.29}$	$82.83_{+2.64}$	$71.67_{-0.44}$	$75.47_{+1.12}$	$63.82_{-1.13}$	$71.63_{-0.53}$	$72.89_{+0.47}$
$\text{Intuitor}_{\text{PHYS.}}$	$74.49_{+0.51}$	$72.53_{+1.41}$	$83.43_{+3.24}$	$72.07_{-0.04}$	$75.63_{+1.28}$	$54.81_{-10.14}$	$67.76_{-4.41}$	$70.85_{-1.57}$
$\text{Intuitor}_{\text{CHEM}}$	$74.37_{+0.38}$	$72.20_{+1.08}$	$83.21_{+3.01}$	$72.63_{+0.52}$	$75.60_{+1.25}$	$54.94_{-10.01}$	$69.10_{-3.06}$	$71.08_{-1.34}$
$\text{Intuitor}_{\text{MMLU}}$	$74.11_{+0.13}$	$73.39_{+2.27}$	$83.51_{+3.31}$	$72.57_{+0.46}$	$75.89_{+1.54}$	$62.92_{-2.03}$	$71.91_{-0.25}$	$73.07_{+0.65}$
$\text{Intuitor}_{\text{Avg.}}$	$74.63_{+0.65}$	$72.73_{+1.61}$	$83.04_{+2.85}$	$72.05_{-0.06}$	$75.61_{+1.26}$	$58.44_{-6.51}$	$69.32_{-2.85}$	$71.70_{-0.72}$
Neuron-OPSD (ours)								
$\text{N-OPSD}_{\text{BIO}}$	$74.02_{+0.03}$	$72.96_{+1.84}$	$82.87_{+2.67}$	$71.97_{-0.14}$	$75.45_{+1.10}$	$66.31_{+1.36}$	$69.46_{-2.71}$	$72.93_{+0.51}$
$\text{N-OPSD}_{\text{MAT.}}$	$73.83_{-0.16}$	$73.39_{+2.27}$	$83.13_{+2.94}$	$72.83_{+0.72}$	$75.79_{+1.44}$	$65.34_{+0.39}$	$71.22_{-0.94}$	$73.29_{+0.87}$
$\text{N-OPSD}_{\text{PHYS.}}$	$74.78_{+0.79}$	$73.71_{+2.60}$	$83.13_{+2.94}$	$71.93_{-0.18}$	$75.89_{+1.54}$	$65.20_{+0.25}$	$71.24_{-0.92}$	$73.33_{+0.91}$
$\text{N-OPSD}_{\text{CHEM}}$	$74.24_{+0.26}$	$73.69_{+2.57}$	$83.21_{+3.01}$	$72.03_{-0.08}$	$75.79_{+1.44}$	$65.45_{+0.50}$	$71.36_{-0.80}$	$73.33_{+0.91}$
$\text{N-OPSD}_{\text{EDU.}}$	$75.13_{+1.14}$	$73.14_{+2.02}$	$83.02_{+2.82}$	$72.85_{+0.74}$	$76.03_{+1.68}$	$72.19_{+7.24}$	$71.96_{-0.21}$	$74.71_{+2.29}$
$\text{N-OPSD}_{\text{MMLU}}$	$74.43_{+0.44}$	$73.46_{+2.34}$	$83.47_{+3.28}$	$73.03_{+0.92}$	$76.10_{+1.75}$	$64.33_{-0.62}$	$72.04_{-0.12}$	$73.46_{+1.04}$
$\text{N-OPSD}_{\text{Avg.}}$	$74.40_{+0.42}$	$73.39_{+2.27}$	$83.14_{+2.94}$	$72.44_{+0.33}$	$75.84_{+1.49}$	$66.47_{+1.52}$	$71.21_{-0.95}$	$73.51_{+1.09}$

Table 7: Cross-domain evaluation on Qwen3-4B-Thinking-2507, performance reported in Avg@8. Each row reports a single model trained on one source domain and evaluated on all remaining test sets. Subscripts give the Δ vs. the untrained baseline.

Model	SciKnowEval					Edu.	MMLU-Pro	Avg.
	Bio	Mat.	Phys.	Chem	*Avg.*			
Qwen3-4B-think	0.195	0.213	0.095	0.173	0.169	0.246	0.184	0.184
TTRL [Zuo et al., 2025]								
TTRL_{Bio}	$0.204_{+0.009}$	$0.220_{+0.007}$	$0.118_{+0.023}$	$0.169_{-0.004}$	$0.178_{+0.009}$	$0.247_{+0.000}$	$0.178_{-0.006}$	$0.189_{+0.005}$
$\text{TTRL}_{\text{Mat.}}$	$0.203_{+0.008}$	$0.208_{-0.005}$	$0.124_{+0.029}$	$0.181_{+0.008}$	$0.179_{+0.010}$	$0.242_{-0.005}$	$0.186_{+0.002}$	$0.191_{+0.006}$
$\text{TTRL}_{\text{Phys.}}$	$0.214_{+0.018}$	$0.215_{+0.002}$	$0.123_{+0.028}$	$0.166_{-0.006}$	$0.180_{+0.011}$	$0.239_{-0.008}$	$0.188_{+0.004}$	$0.191_{+0.007}$
$\text{TTRL}_{\text{Chem}}$	$0.204_{+0.009}$	$0.223_{+0.010}$	$0.118_{+0.023}$	$0.192_{+0.019}$	$0.184_{+0.015}$	$0.251_{+0.005}$	$0.197_{+0.013}$	$0.198_{+0.013}$
$\text{TTRL}_{\text{Edu.}}$	$0.199_{+0.003}$	$0.215_{+0.003}$	$0.111_{+0.015}$	$0.164_{-0.009}$	$0.172_{+0.003}$	$0.248_{+0.002}$	$0.202_{+0.018}$	$0.190_{+0.005}$
$\text{TTRL}_{\text{MMLU}}$	$0.200_{+0.005}$	$0.224_{+0.011}$	$0.114_{+0.019}$	$0.169_{-0.003}$	$0.177_{+0.008}$	$0.243_{-0.004}$	$0.198_{+0.014}$	$0.191_{+0.007}$
$\text{TTRL}_{\text{Avg.}}$	$0.204_{+0.009}$	$0.218_{+0.005}$	$0.118_{+0.023}$	$0.174_{+0.001}$	$0.178_{+0.009}$	$0.245_{-0.002}$	$0.191_{+0.007}$	$0.192_{+0.007}$
Intuitor [Zhao et al., 2025b]								
$\text{Intuitor}_{\text{Bio}}$	$0.208_{+0.013}$	$0.236_{+0.023}$	$0.141_{+0.046}$	$0.210_{+0.038}$	$0.199_{+0.030}$	$0.312_{+0.066}$	$0.229_{+0.045}$	$0.223_{+0.039}$
$\text{Intuitor}_{\text{Mat.}}$	$0.209_{+0.014}$	$0.212_{-0.001}$	$0.124_{+0.029}$	$0.166_{-0.007}$	$0.178_{+0.009}$	$0.246_{+0.000}$	$0.188_{+0.004}$	$0.191_{+0.007}$
$\text{Intuitor}_{\text{Phys.}}$	$0.213_{+0.018}$	$0.229_{+0.016}$	$0.117_{+0.021}$	$0.176_{+0.004}$	$0.184_{+0.015}$	$0.341_{+0.095}$	$0.210_{+0.026}$	$0.214_{+0.030}$
$\text{Intuitor}_{\text{Chem}}$	$0.213_{+0.017}$	$0.225_{+0.012}$	$0.130_{+0.034}$	$0.198_{+0.026}$	$0.191_{+0.022}$	$0.319_{+0.073}$	$0.234_{+0.050}$	$0.220_{+0.036}$
$\text{Intuitor}_{\text{MMLU}}$	$0.197_{+0.002}$	$0.205_{-0.008}$	$0.113_{+0.018}$	$0.142_{-0.031}$	$0.164_{-0.005}$	$0.267_{+0.021}$	$0.174_{-0.010}$	$0.183_{-0.001}$
$\text{Intuitor}_{\text{Avg.}}$	$0.208_{+0.013}$	$0.221_{+0.008}$	$0.125_{+0.030}$	$0.179_{+0.006}$	$0.183_{+0.014}$	$0.297_{+0.051}$	$0.207_{+0.023}$	$0.206_{+0.022}$
Neuron-OPSD, ours								
$\text{N-OPSD}_{\text{Bio}}$	$0.193_{-0.003}$	$0.199_{-0.014}$	$0.111_{+0.015}$	$0.159_{-0.013}$	$0.165_{-0.004}$	$0.221_{-0.025}$	$0.194_{+0.010}$	$0.180_{-0.005}$
$\text{N-OPSD}_{\text{Mat.}}$	$0.206_{+0.011}$	$0.207_{-0.006}$	$0.121_{+0.026}$	$0.156_{-0.017}$	$0.173_{+0.004}$	$0.219_{-0.028}$	$0.185_{+0.001}$	$0.182_{-0.002}$
$\text{N-OPSD}_{\text{Phys.}}$	$0.196_{+0.001}$	$0.198_{-0.015}$	$0.114_{+0.019}$	$0.170_{-0.003}$	$0.170_{+0.001}$	$0.217_{-0.029}$	$0.189_{+0.005}$	$0.181_{-0.004}$
$\text{N-OPSD}_{\text{Chem}}$	$0.198_{+0.003}$	$0.215_{+0.002}$	$0.117_{+0.022}$	$0.171_{-0.002}$	$0.175_{+0.006}$	$0.225_{-0.021}$	$0.180_{-0.004}$	$0.184_{+0.000}$
$\text{N-OPSD}_{\text{Edu.}}$	$0.195_{-0.000}$	$0.218_{+0.005}$	$0.122_{+0.027}$	$0.171_{-0.002}$	$0.177_{+0.007}$	$0.191_{-0.056}$	$0.179_{-0.005}$	$0.179_{-0.005}$
$\text{N-OPSD}_{\text{MMLU}}$	$0.206_{+0.010}$	$0.215_{+0.002}$	$0.113_{+0.018}$	$0.158_{-0.014}$	$0.173_{+0.004}$	$0.239_{-0.008}$	$0.180_{-0.004}$	$0.185_{+0.001}$
$\text{N-OPSD}_{\text{Avg.}}$	$0.199_{+0.004}$	$0.209_{-0.004}$	$0.116_{+0.021}$	$0.164_{-0.009}$	$0.172_{+0.003}$	$0.219_{-0.027}$	$0.185_{+0.001}$	$0.182_{-0.002}$

Table 8: Expected Calibration Error (ECE), where lower is better, on the same test sets as Tab. 7. Each cell shows the absolute ECE with the delta vs the *Base* row in subscript, with green for lower ECE and better calibration and red for worse calibration. Gray cells mark the source domain. Shaded rows show the column-wise average across the trained models per method.