LOOK BEFORE YOU SELECT: RETHINKING VOCABULARY SPARSIFICATION IN ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) uses teacher correction on student-generated responses. Full-vocabulary correction can provide important corrections even for tokens that the student assigns low probability, but backpropagating through all token logits becomes memory-intensive for long sequences. Existing memory-saving approaches estimate corrections from sampled tokens or restrict supervision to the student's TopK tokens, introducing sampling noise or changing the full-vocabulary correction. We introduce **SparseOPD**, which uses full-vocabulary teacher correction to determine which corrections matter before selecting the token logits to differentiate. SparseOPD first constructs the full-vocabulary correction without retaining its backward graph, then selects tokens by correction magnitude rather than student probability. Signed residual compensation preserves the total promoting and suppressing correction mass, while correction-aware budget allocation distributes the sparse support across positions. Finally, the update backpropagates only through the selected token logits. Across six task–scale settings spanning mathematics, chemistry QA, and multimodal reasoning, SparseOPD outperforms Sampled Token and TopK in task-average accuracy and matches or exceeds Full Vocabulary. Gradient cosine similarity reaches 99% on 4B mathematics, while 8K full-parameter profiling shows 70.5% lower backward memory.

1 INTRODUCTION

On-policy distillation (OPD) trains a student language model using teacher correction on responses generated by the student itself (Agarwal et al., 2024; Ko et al., 2024; Ye et al., 2025; Song & Zheng, 2026; Li et al., 2026g). Full-vocabulary OPD compares the complete teacher and student distributions at each prediction position (Gu et al., 2024; Zhong et al., 2024; Zhang et al., 2024c; Wu et al., 2025; Zhang et al., 2025), providing a token-specific correction over the entire vocabulary. Such correction can capture important corrections beyond the student's most likely tokens, but directly backpropagating through the full vocabulary is memory-intensive, especially for long reasoning sequences. This creates an apparent trade-off: preserving full-vocabulary correction seems to require backpropagating through the full vocabulary.

Existing memory-saving paradigms reduce this cost by changing the correction, typically through sampled-token estimation or TopK support restriction (Anshumann et al., 2025; Huang & Wei, 2026). Sampled Token replaces the deterministic full-vocabulary correction with an estimate based on a single sampled token; although unbiased at a fixed prefix, the resulting update can depend strongly on which token happens to be sampled. TopK conditions the objective on a support formed by the student's most probable tokens, leaving tail tokens that may require large corrections outside the support and changing the correction even within the retained support. As illustrated in Figure 1, a sampled action can incorrectly promote "sum", while a TopK support can omit the teacher-preferred tail tokens "overlap" and "intersection".

Motivated by these limitations, we ask: *Can we construct updates from full-vocabulary teacher correction while backpropagating through the logits of only a sparse set of tokens?* To this end, we introduce **SparseOPD**, which uses full-vocabulary correction while backpropagating only through the logits of selected tokens. SparseOPD first constructs the full-vocabulary correction without retaining its backward graph, and then converts it into a fixed sparse update. Because the complete correction is available before sparsification, SparseOPD selects tokens based on their corrections

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

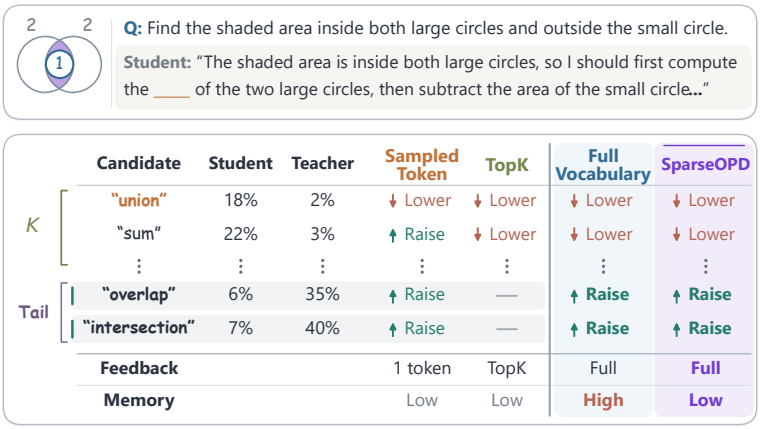


Figure 1: **Full-vocabulary correction need not require full-vocabulary differentiation.** Arrows show logit corrections in a constructed multimodal QA example. Full Vocabulary suppresses "union" and "sum" and promotes "overlap" and "intersection," at high memory cost. Sampled Token incorrectly promotes "sum," while TopK provides no direct correction to "overlap" or "intersection." SparseOPD preserves the important full-vocabulary corrections with low memory use.

rather than student probability, allowing low-probability tokens to be retained when their corrections are large. The discarded positive and negative correction mass is then redistributed among retained corrections of the same sign, preserving the total positive and negative correction mass of the dense update. The resulting update therefore backpropagates only through the logits of the selected tokens.

Empirically, we validate SparseOPD on mathematical reasoning, chemistry question answering, and multimodal mathematical reasoning. With an average support budget of K entries per position, SparseOPD outperforms Sampled Token and TopK in all six task–scale settings and matches or exceeds Full Vocabulary (Figure 3). Full-parameter profiling of the 4B model shows 70.5% lower peak backward memory than Full Vocabulary at 8K sequence length, with larger relative savings across longer measured sequences. Gradient analysis in the 4B mathematical reasoning setting further shows a training-averaged cosine similarity of 0.989 to the Full Vocabulary gradient.

Our contributions are as follows:

- We rethink vocabulary sparsification in OPD by treating the full-vocabulary logit correction as the reference for sparse updates, showing that Sampled Token introduces sampling variance while TopK omits tail corrections and changes corrections through restricted-support conditioning.
- We introduce SparseOPD, which constructs full-vocabulary corrections before converting them into sparse updates using signed correction selection, residual compensation, and budget allocation across the batch.
- Across mathematical reasoning, chemistry QA, and multimodal mathematical reasoning, SparseOPD achieves higher task-average accuracy than Sampled Token and TopK. Complementary profiling and gradient measurements on the 4B mathematical reasoning setting show substantially lower backward memory and close agreement with the Full Vocabulary gradient.

2 Rethinking Vocabulary Sparsification

2.1 Preliminaries: Full-Vocabulary OPD

Let π_θ be a student with trainable parameters θ and π_T a frozen teacher, sharing a vocabulary $\mathcal{V}$. On-policy distillation (OPD) obtains conditioning prefixes from the student's own generations (Agarwal et al., 2024). For a prompt $x \sim \mathcal{D}$ from the training distribution, let $y \sim \pi_\theta(\cdot \mid x)$ denote the student rollout and $s_t = (x, y_{<t})$ its prefix at position t. Both models are evaluated on the same prefix, yielding $p_{t,v} = \pi_\theta(v \mid s_t)$ and $q_{t,v} = \pi_T(v \mid s_t)$ for each token $v \in \mathcal{V}$. Training averages per-position losses over valid completion tokens and treats the sampled rollout as observed data, without

differentiating through discrete sampling. Within this common setting, the paired distributions can yield training signals with different computational costs and statistical properties.

Full-vocabulary OPD. The full-vocabulary formulation compares the complete distributions at each prefix using reverse Kullback–Leibler (KL) divergence. Let $z_{t,v}$ denote the student logit and $r_{t,v} = \log(p_{t,v}/q_{t,v})$ the student-to-teacher log-ratio. The per-position loss and its logit-gradient coefficients are

$$D_t = \mathrm{KL}(p_t \| q_t) = \sum_{v \in \mathcal{V}} p_{t,v} r_{t,v}, \qquad C_{t,v} := \frac{\partial D_t}{\partial z_{t,v}} = p_{t,v}(r_{t,v} - D_t). \tag{1}$$

The vector $C_t = (C_{t,v})_{v \in \mathcal{V}}$ specifies a deterministic, token-specific correction over the entire vocabulary. Positive coefficients suppress the corresponding logits under gradient descent, while negative coefficients promote them. The coefficients are zero-sum, $\sum_{v \in \mathcal{V}} C_{t,v} = 0$, so the total suppressive and promotive correction masses are equal. The chain rule maps these coefficients to the model's parameter gradient. Directly differentiating this loss retains vocabulary-sized intermediates for backward, with memory cost growing with both vocabulary size and the number of valid positions.

2.2 What Existing Sparsification Changes

Sampled-token OPD. One way to avoid evaluating every token-specific discrepancy is to reuse the generated action $a_t = y_t \sim p_t$. Its single-action coefficient estimator satisfies

$$\widehat{C}^{\mathrm{sample}}_{t,v} = r_{t,a_t}\big(\mathbf{1}\{v = a_t\} - p_{t,v}\big), \qquad \mathbb{E}_{a_t \sim p_t}\Big[\widehat{C}^{\mathrm{sample}}_{t,v}\Big] = C_{t,v}. \tag{2}$$

Thus the full-vocabulary correction is recovered in expectation at a fixed prefix. In one realization, however, every coordinate depends on the single observed ratio r_{t,a_t}. The $-p_{t,v}$ term couples unsampled logits to that action; it does not evaluate their own teacher–student discrepancies. Their token-specific corrections emerge through the sampling expectation, introducing sampling variance.

TopK OPD. An alternative restricts the objective to the student's highest-probability tokens. Let $S_t \subset \mathcal{V}$ contain the K largest student probabilities, with indices fixed during backpropagation. Define the support masses $Z^p_t = \sum_{v \in S_t} p_{t,v}$ and $Z^q_t = \sum_{v \in S_t} q_{t,v}$, and distributions $\bar{p}^S_{t,v} = p_{t,v}/Z^p_t$ and $\bar{q}^S_{t,v} = q_{t,v}/Z^q_t$ for $v \in S_t$. TopK minimizes $D^S_t = \mathrm{KL}(\bar{p}^S_t \| \bar{q}^S_t)$.

Renormalization changes which discrepancies the objective measures. Let $D^{S^c}_t$ be the analogous conditional KL on $S^c_t = \mathcal{V} \setminus S_t$, and let D^{mass}_t be the binary KL between the support probabilities Z^p_t and Z^q_t. Separating support membership from the distributions within each part gives

$$D_t = \underbrace{D^{\mathrm{mass}}_t}_{\text{support–tail mass}} + \underbrace{Z^p_t D^S_t}_{\text{support shape}} + \underbrace{(1 - Z^p_t) D^{S^c}_t}_{\text{tail shape}}. \tag{3}$$

TopK optimizes only the conditional shape D^S_t, leaving out the support–tail mass discrepancy, the tail shape, and the support weight. This restriction also appears directly in its logit coefficients:

$$C^{\mathrm{TopK}}_{t,v} = \bar{p}^S_{t,v}\left(\log \frac{\bar{p}^S_{t,v}}{\bar{q}^S_{t,v}} - D^S_t\right) \ (v \in S_t), \qquad 0 \ (v \notin S_t). \tag{4}$$

Tail coordinates receive no direct correction, and the retained coefficients are derived from a conditional objective rather than by truncating C_t.

Full-vocabulary OPD provides a complete reference correction, but its direct realization is memory-intensive. Sampling and support conditioning limit the token-specific feedback in different ways, yielding a noisy realization or a restricted objective.

2.3 Full-Vocabulary Correction with Sparse Execution

Despite these different mechanisms, both memory-saving paradigms alter the reference correction before deciding which token logits to backpropagate through: Sampled Token estimates it from

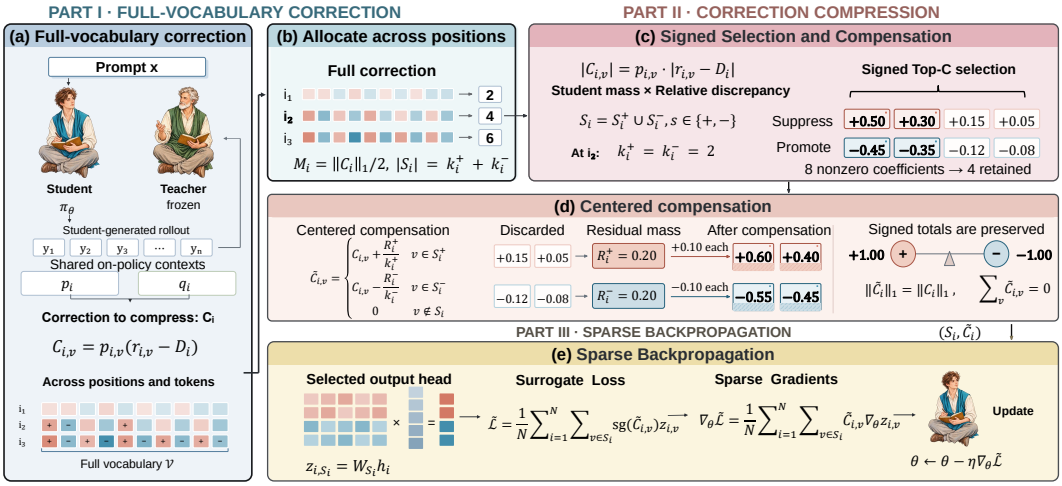


Figure 2: **Overview of SparseOPD. (a)** Full-vocabulary correction $C_{i,v} = p_{i,v}(r_{i,v} - D_i)$. **(b)** Correction mass $M_i = \|C_i\|_1/2$ allocates support. **(c)** Signed Top-C selects $S_i = S_i^+ \cup S_i^-$. **(d)** Centered compensation yields $\widetilde{C}_i$ while preserving signed totals. **(e)** Sparse backpropagation passes only through selected logits $z_{i,S_i} = W_{S_i} h_i$.

a sampled action, while TopK defines it only after restricting the vocabulary support. This suggests a different order of operations: first construct the full-vocabulary correction, and only then decide which token logits to retain for backpropagation. The task is therefore to approximate the full-vocabulary correction as faithfully as possible under a shared budget on token logits for backpropagation, while retaining access to the complete teacher and student distributions. This reframes vocabulary sparsification as correction compression rather than simple support restriction.

Target: Full-vocabulary correction with sparse execution.

Let $i = 1, \ldots, N$ index the valid completion positions in one optimizer update, with full-vocabulary reference coefficients C_i. Given an average support budget K, use the complete C_i to construct supports $S_i \subseteq \mathcal{V}$ and coefficients $\widetilde{C}_i$ supported on S_i, subject to

$$\sum_{i=1}^{N} |S_i| = KN.$$

The goal is to approximate C_i in coefficient space while restricting output-head differentiation to the selected entries S_i.

3 SparseOPD

Following the reformulation of vocabulary sparsification as correction compression in Section 2, we propose **SparseOPD**, which takes the full-vocabulary correction as the reference and backpropagates only through a sparse set of token logits. SparseOPD first constructs the full-vocabulary reference coefficients without building an autograd graph, and then compresses this correction into a sparse execution plan that approximates the full-vocabulary update with far fewer differentiable entries. With the full correction available before support selection, the compression is guided by the correction itself. SparseOPD consists of three stages: (1) batch-global budget allocation distributes the shared support budget according to correction mass; (2) signed selection and compensation retains large-magnitude coefficients within each sign and redistributes discarded mass while preserving the signed totals; (3) Sparse Backpropagation executes the resulting compressed correction through selected output-head rows. Figure 2 provides an overview of our method.

3.1 BATCH-GLOBAL BUDGET ALLOCATION

We first compute the full-vocabulary coefficients C_i at each valid position using Equation (1), without building an autograd graph for this computation. Uniform support may over-allocate capacity to positions with weak teacher–student disagreement and under-allocate it where corrections are stronger. We therefore share the KN budget across positions according to their total coefficient magnitude. By the zero-sum property of C_i, its suppressive and promotive corrections carry equal total mass:

$$M_i := \sum_{v:C_{i,v}>0} C_{i,v} = \sum_{v:C_{i,v}<0} (-C_{i,v}) = \frac{1}{2}\|C_i\|_1. \tag{5}$$

We use the common mass M_i as a position-level measure of correction strength, while the two signs distinguish suppressive and promotive corrections.

Each active sign needs at least one retained entry for the compensation below. For $s \in \{+,-\}$, let $\ell_i^s = \mathbf{1}\{M_i > 0\}$ be this minimum and $u_i^s \geq \ell_i^s$ a prescribed capacity, no larger than the number of coefficients with that sign. Assuming the total budget is feasible, $\sum_{i,s} \ell_i^s \leq KN \leq \sum_{i,s} u_i^s$, let $\alpha \geq 0$ denote a global allocation scale. We distribute the remaining budget in proportion to M_i until each group reaches its capacity:

$$\bar{k}_i^s = \ell_i^s + \min\{u_i^s - \ell_i^s,\, \alpha M_i\}, \qquad \sum_{i,s} \bar{k}_i^s = KN. \tag{6}$$

We choose α to meet the shared budget and round $\bar{k}_i^s$ to integer cardinalities k_i^s while preserving both the bounds and $\sum_{i,s} k_i^s = KN$. Thus positions share capacity according to correction demand rather than receiving the same support size independently.

3.2 SIGNED SELECTION AND COMPENSATION

With the support sizes fixed, we prioritize coordinates whose removal most distorts C_i. Student probability alone does not measure this importance, because the coefficient magnitude factorizes as

$$|C_{i,v}| = \underbrace{p_{i,v}}_{\text{student mass}} \cdot \underbrace{|r_{i,v} - D_i|}_{\text{relative discrepancy}}. \tag{7}$$

A token's update importance depends on both its student mass and its discrepancy relative to the KL baseline. A high-probability token can require little correction, while a tail token can matter when its relative discrepancy is large.

For a fixed support size k, the direct truncation error is $\sum_{v\notin S} C_{i,v}^2$. Selecting the k largest magnitudes $|C_{i,v}|$ minimizes this error. This coefficient ranking, which we call Top-C, determines which individual entries are most costly to discard.

With the cardinalities fixed, signed Top-C retains the k_i^+ largest positive coefficients and the k_i^- largest-magnitude negative coefficients, forming S_i^+ and S_i^- with union S_i.

Centered compensation. Simply truncating to these supports would discard correction mass and can destroy the zero-sum structure of C_i. The missing masses are $R_i^+ = M_i - \sum_{v\in S_i^+} C_{i,v}$ and $R_i^- = M_i - \sum_{v\in S_i^-}(-C_{i,v})$. To preserve the total correction in each direction, centered compensation redistributes each residual over retained entries of the same sign:

$$\widetilde{C}_{i,v} = C_{i,v} + R_i^+/k_i^+ \ (v \in S_i^+), \qquad C_{i,v} - R_i^-/k_i^- \ (v \in S_i^-), \qquad 0 \ (v \notin S_i). \tag{8}$$

Preserved structure and approximation error. Compensation restores signed sums M_i and $-M_i$, hence $\sum_v \widetilde{C}_{i,v} = 0$ and $\|\widetilde{C}_i\|_1 = \|C_i\|_1$. For fixed supports $S_i^\pm$, the uniform shift is the Euclidean projection of the retained coefficients onto these constraints on correction mass. The remaining approximation reflects losing tail coordinates and representing their aggregate mass on the support. These two effects are separated exactly by

$$\|\widetilde{C}_i - C_i\|_2^2 = \underbrace{\sum_{v\notin S_i} C_{i,v}^2}_{\text{support truncation}} + \underbrace{\frac{(R_i^+)^2}{k_i^+} + \frac{(R_i^-)^2}{k_i^-}}_{\text{residual compression}}. \tag{9}$$

Empty sign-group terms are zero. Top-C retains the coordinates contributing most to the first term. Additional support reduces the second term by retaining more correction mass and spreading the residual across more entries. The decomposition explains why selection and compensation are complementary and why positions carrying larger correction mass can benefit from more capacity.

3.3 Sparse Backpropagation

The selected support and compensated coefficients form a fixed sparse plan $(S_i, \widetilde{C}_i)$ for backpropagation. Let $h_i \in \mathbb{R}^{d_h}$ be the student hidden state and $W \in \mathbb{R}^{|\mathcal{V}| \times d_h}$ the output projection. The planning pass materializes $z_i = Wh_i$ transiently under no gradient to construct the full-vocabulary statistics. These vocabulary-sized tensors are discarded once the sparse plan is fixed.

The differentiable pass reconstructs only $z_{i,S_i} = W_{S_i} h_i$, using the selected rows $W_{S_i} \in \mathbb{R}^{|S_i| \times d_h}$. To realize the planned coefficients without differentiating through their construction, we use a linear surrogate with stop-gradient $\mathrm{sg}(\cdot)$:

$$\widetilde{\mathcal{L}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{v \in S_i} \mathrm{sg}(\widetilde{C}_{i,v}) z_{i,v}, \qquad \nabla_\theta \widetilde{\mathcal{L}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{v \in S_i} \widetilde{C}_{i,v} \nabla_\theta z_{i,v}. \tag{10}$$

Equation (10) realizes the prescribed sparse coefficients exactly; the approximation is fixed by compression. Full-vocabulary statistics inform this update, while autograd stores vocabulary-axis intermediates only for the selected entries.

Distributed implementation. Allocation covers all valid positions in one optimizer update, collecting position-level masses across gradient-accumulation microbatches and data-parallel workers. Once the budgets are fixed, support construction and selected-head backpropagation remain local.

4 Experiments

4.1 Experimental Setup

Datasets. We train on DAPO14k (Guanning-AI, n.d.) for mathematics, the Chemistry L3 multiple-choice partition of SciKnowEval V2 (Feng et al., 2024) for chemistry, and MMR1-Math-RL-Data-v0 (Leng et al., 2025) for multimodal reasoning. Mathematics evaluation covers MATH500 (Lightman et al., 2023), Minerva (Lewkowycz et al., 2022), AMC23 (Math-AI Team, 2025), and AIME25 (Zhang & Math-AI Team, 2025). Chemistry evaluation uses the held-out SciKnowEval questions and the chemistry subset of GPQA-Diamond (Rein et al., 2023). Multimodal evaluation covers MathVision (Wang et al., 2024), MathVista (Lu et al., 2024), We-Math (Qiao et al., 2024), and MathVerse (Zhang et al., 2024b).

Implementation. For mathematics and chemistry, we distill Qwen3-1.7B/4B into their Base counterparts (Yang et al., 2025) and report Mean@8 accuracy over eight decoding runs. For multimodal reasoning, MMFineReason-2B/4B (Lin et al., 2026) teach Qwen3-VL-2B/4B-Instruct (Bai et al., 2025), evaluated by single-generation accuracy with LMMs-Eval (Zhang et al., 2024a). Further details are in Appendix B.

4.2 Main Results

Tables 1 and 2 compare SparseOPD with Full Vocabulary, Sampled Token, and TopK across tasks and model scales.

Research Question 1: Can sparse execution preserve the training benefit of full-vocabulary correction? SparseOPD matches or exceeds the Full Vocabulary average in five of the six task–scale settings at the reported precision. On Qwen3-4B mathematics, it achieves 43.2 compared with 41.6 for Full Vocabulary; both multimodal student scales also exceed the dense reference. The exception is 1.7B chemistry, where SparseOPD scores 35.8 against 37.0. This pattern supports the separation underlying SparseOPD: much of the training benefit of full-vocabulary correction survives when its correction is compressed before differentiation.

Table 1: **Mathematical reasoning and chemistry QA.** Superscripts show percentage-point changes from Base; bold indicates the best result.

Method	Mathematical Reasoning					Chemistry QA		
	MATH	Minerva	AMC23	AIME25	**Avg.**	Chemistry	GPQA	**Avg.**
Qwen3-1.7B → Qwen3-1.7B-Base								
Teacher	72.3	28.8	44.4	8.3	38.4	42.0	26.5	34.2
Base	46.5	15.3	26.9	2.9	22.9	24.9	22.9	23.9
Sampled Token	54.1$^{+7.6}$	18.2$^{+2.9}$	32.8$^{+5.9}$	3.8$^{+0.9}$	27.2$^{+4.3}$	34.3$^{+9.4}$	28.6$^{+5.7}$	31.5$^{+7.6}$
TopK	63.6$^{+17.1}$	25.7$^{+10.4}$	34.1$^{+7.2}$	4.2$^{+1.3}$	31.9$^{+9.0}$	40.7$^{+15.8}$	29.2$^{+6.3}$	34.9$^{+11.0}$
SparseOPD(Ours)	**64.5**$^{+18.0}$	**26.1**$^{+10.8}$	**35.9**$^{+9.0}$	**5.8**$^{+2.9}$	**33.1**$^{+10.2}$	**41.1**$^{+16.2}$	**30.5**$^{+7.6}$	**35.8**$^{+11.9}$
Full Vocabulary	*63.4*$^{+16.9}$	*26.5*$^{+11.2}$	*35.3*$^{+8.4}$	*6.3*$^{+3.4}$	*32.9*$^{+10.0}$	*42.9*$^{+18.0}$	*31.1*$^{+8.2}$	*37.0*$^{+13.1}$
Qwen3-4B → Qwen3-4B-Base								
Teacher	83.3	38.8	64.1	19.6	51.4	41.5	31.9	36.7
Base	56.8	18.7	41.6	8.3	31.3	34.3	29.7	32.0
Sampled Token	70.6$^{+13.8}$	29.7$^{+11.0}$	46.3$^{+4.7}$	12.1$^{+3.8}$	39.7$^{+8.4}$	38.0$^{+3.7}$	31.3$^{+1.6}$	34.7$^{+2.7}$
TopK	**77.4**$^{+20.6}$	**30.1**$^{+11.4}$	45.9$^{+4.3}$	11.7$^{+3.4}$	41.3$^{+10.0}$	38.8$^{+4.5}$	29.8$^{+0.1}$	34.3$^{+2.3}$
SparseOPD(Ours)	76.9$^{+20.1}$	**30.1**$^{+11.4}$	**50.0**$^{+8.4}$	**15.8**$^{+7.5}$	**43.2**$^{+11.9}$	**39.7**$^{+5.4}$	**33.2**$^{+3.5}$	**36.5**$^{+4.5}$
Full Vocabulary	*77.8*$^{+21.0}$	*29.8*$^{+11.1}$	*46.3*$^{+4.7}$	*12.5*$^{+4.2}$	*41.6*$^{+10.3}$	*40.7*$^{+6.4}$	*32.3*$^{+2.6}$	*36.5*$^{+4.5}$

Table 2: **Multimodal mathematical reasoning.** Superscripts show percentage-point changes from Student; bold indicates the best result.

Method	MathVision	MathVista	WeMath	MathVerse	**Avg.**
MMFineReason-2B → Qwen3-VL-2B-Instruct					
Teacher	24.4	65.1	66.1	46.0	50.4
Student	15.9	59.3	56.2	36.5	42.0
Sampled Token	22.2$^{+6.3}$	60.8$^{+1.5}$	62.9$^{+6.7}$	40.7$^{+4.2}$	46.7$^{+4.7}$
TopK	23.9$^{+8.1}$	62.2$^{+2.9}$	62.2$^{+6.0}$	**43.3**$^{+6.7}$	47.9$^{+5.9}$
SparseOPD(Ours)	**24.2**$^{+8.3}$	**63.8**$^{+4.5}$	**64.8**$^{+8.6}$	42.0$^{+5.5}$	**48.7**$^{+6.7}$
Full Vocabulary	*24.9*$^{+9.0}$	*62.3*$^{+3.0}$	*63.2*$^{+7.0}$	*39.2*$^{+2.7}$	*47.4*$^{+5.4}$
MMFineReason-4B → Qwen3-VL-4B-Instruct					
Teacher	38.3	75.3	74.2	51.4	59.8
Student	27.3	67.8	60.8	41.1	49.3
Sampled Token	30.5$^{+3.2}$	70.5$^{+2.7}$	68.5$^{+7.7}$	45.5$^{+4.4}$	53.8$^{+4.5}$
TopK	35.7$^{+8.4}$	71.3$^{+3.5}$	**75.5**$^{+14.7}$	54.1$^{+13.0}$	59.2$^{+9.9}$
SparseOPD(Ours)	**35.8**$^{+8.5}$	**72.9**$^{+5.1}$	**75.5**$^{+14.7}$	**54.8**$^{+13.7}$	**59.7**$^{+10.5}$
Full Vocabulary	*36.1*$^{+8.8}$	*73.0*$^{+5.2}$	*75.0*$^{+14.2}$	*54.4*$^{+13.3}$	*59.6*$^{+10.4}$

Research Question 2: Does correction-aware compression make better use of the same sparse budget? With an average budget of 20 output-head coordinates per valid position, SparseOPD outperforms TopK on all six task averages by 0.58–2.2 percentage points. It also improves over Sampled Token by 1.8–5.97 points. The comparison with TopK holds support size fixed while changing how sparse updates are formed. Its consistent advantage supports using the full-vocabulary correction to guide compression, making the same differentiable support budget more effective.

Research Question 3: Do the task-average gains reflect broad improvements across benchmarks? The advantage over TopK extends to 16 of the 20 benchmark–scale comparisons, with two ties at the reported precision. Gains include AMC23 and AIME25 on Qwen3-4B, as well as We-Math on Qwen3-VL-2B. The remaining trade-offs occur on 4B MATH500 and 2B MathVerse,

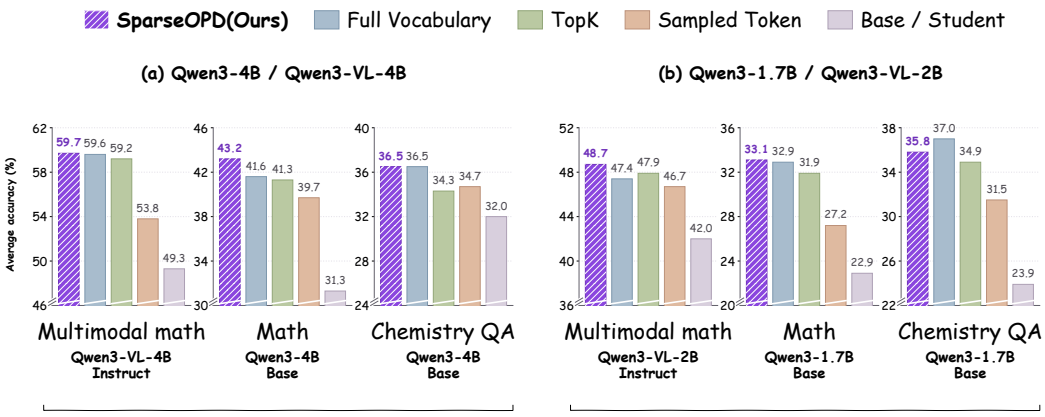


Figure 3: **Task-average accuracy across student scales.** SparseOPD outperforms both sparse baselines in all six settings and matches or exceeds Full Vocabulary in five.

Table 3: **Ablation of SparseOPD on mathematical reasoning.** Results use Qwen3-4B → Qwen3-4B-Base under the same support budget. Red superscripts show drops from SparseOPD.

Variant	MATH500	Minerva	AMC23	AIME25	Avg.
SparseOPD(Ours)	**76.9**	**30.1**	**50.0**	**15.8**	**43.2**
Student Probability Ranking	76.0$^{-0.9}$	29.0$^{-1.1}$	45.6$^{-4.4}$	14.2$^{-1.6}$	41.2$^{-2.0}$
w/o Centered Compensation	76.3$^{-0.6}$	29.2$^{-0.9}$	47.5$^{-2.5}$	14.2$^{-1.6}$	41.8$^{-1.4}$
w/o Batch-global Allocation	76.6$^{-0.3}$	29.6$^{-0.5}$	48.1$^{-1.9}$	15.0$^{-0.8}$	42.3$^{-0.9}$

where TopK retains a lead. Task-average improvements therefore reflect broad, but not uniform, gains across individual benchmarks.

4.3 Ablation Study

Research Question 4: Do selection, compensation, and global allocation each contribute under the same sparse budget? Table 3 tests the role of selection, compensation, and allocation in preserving useful corrections. Each variant changes one component on Qwen3-4B mathematical reasoning, holding the remaining components and total support budget KN fixed, with $K = 20$.

Signed Top-C selection. The largest ablation drop comes from replacing coefficient-magnitude ranking with student-probability ranking within each sign group: average accuracy falls by 2.0 percentage points. Student likelihood alone does not capture the teacher–student discrepancy that drives correction strength. The result supports retaining coordinates according to their contribution to the full-vocabulary update.

Centered compensation. Retaining strong coefficients does not recover all useful correction. Using their original values without compensation lowers average accuracy by 1.4 percentage points. Compensation preserves the aggregate promoting and suppressing signal carried by discarded candidates, allowing the retained support to represent more than its selected coefficients alone.

Batch-global allocation. The support budget is more effective when it can follow correction demand across positions. Replacing global allocation with a fixed budget of K coordinates per valid position lowers average accuracy by 0.9 percentage points. Since total support is unchanged, this comparison supports redistributing capacity toward positions with stronger corrections.

4.4 Analysis and Discussion

Gradient fidelity tests preservation of the reference update, while efficiency profiling tests the cost of realizing it. Both analyses use Qwen3-4B as the teacher and Qwen3-4B-Base as the student, with mathematical reasoning batches for the gradient comparison.

Research Question 5: Does SparseOPD preserve the full-vocabulary update? At matched student states and on-policy batches, let g and g_F denote the candidate and Full Vocabulary parameter gradients before clipping. We report cosine similarity, relative ℓ_2 error $\|g - g_F\|_2/\|g_F\|_2$, norm ratio $\|g\|_2/\|g_F\|_2$, and sign agreement, excluding jointly zero coordinates and treating other zeros as nonnegative. Metrics are computed at each recorded step and averaged over full-parameter training.

SparseOPD is closest to the reference on all four metrics (Table 4). Its relative ℓ_2 error is approximately one third of TopK's (0.15 versus 0.45), while cosine similarity and norm ratio are both near one. SparseOPD also achieves the highest sign agreement, supporting preservation of the dense update through sparse execution.

Table 4: **Training-averaged gradient fidelity to Full Vocabulary OPD.** Bold marks the best.

Method	Cos. ↑	Rel. ℓ_2 ↓	Norm → 1	Sign ↑
Full Vocabulary	1.00	0.00	1.00	1.00
Sampled Token	0.29	5.43	5.63	0.59
TopK	0.90	0.45	1.05	0.83
SparseOPD(Ours)	**0.99**	**0.15**	**0.99**	**0.91**

Research Question 6: How much backward memory does sparse execution save? Decoupling correction construction from differentiation reduces memory (Table 5).

Full-vocabulary statistics are computed without an autograd graph and discarded after planning; backpropagation uses only selected output-head rows. This reduces backward memory by 70.5% at 8K, with larger savings at longer sequences. Backward time improves more modestly because the Transformer backbone still requires full backpropagation, while additional planning adds overhead (Appendix B.4).

Table 5: **Memory usage.** Peak extra GPU allocation above the pre-step baseline (Appendix B.4).

Length	Method	Memory (GiB)	
		Forward	Backward
4K	Full Vocabulary	12.82	10.09
	SparseOPD(Ours)	**1.94** (↓84.9%)	**3.59** (↓64.5%)
8K	Full Vocabulary	25.50	20.05
	SparseOPD(Ours)	**3.88** (↓84.8%)	**5.91** (↓70.5%)
16K	Full Vocabulary	50.85	39.96
	SparseOPD(Ours)	**7.76** (↓84.7%)	**10.54** (↓73.6%)

5 Related Work

On-policy distillation trains on student-generated trajectories with teacher feedback (Agarwal et al., 2024; Ko et al., 2024; Ye et al., 2025). For large language models, reverse-KL objectives provide full-vocabulary supervision (Gu et al., 2024; Zhong et al., 2024). Prior work reduces this cost by sampling vocabulary entries or restricting supervision to a small support (Anshumann et al., 2025; Huang & Wei, 2026). SparseOPD instead forms the full-vocabulary correction first and compresses it for sparse backpropagation. See Appendix A for details.

6 Conclusions

We introduced SparseOPD, an on-policy distillation method that separates the construction of full-vocabulary corrections from sparse backpropagation. By selecting entries according to correction magnitude and compensating for discarded correction mass, SparseOPD compresses the update while retaining information from the complete teacher and student distributions. Across mathematical reasoning, chemistry question answering, and multimodal reasoning, our experiments show that sparse execution can preserve much of the training benefit of full-vocabulary supervision. Complementary analyses demonstrate close agreement with the full-vocabulary gradient and substantially lower peak backward memory in the measured setting. Overall, SparseOPD preserves full-vocabulary correction while substantially reducing the memory required for backpropagation.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing of the paper. We did not use generative AI tools for any other part of the research. All AI-assisted edits were reviewed by the authors for accuracy and consistency with the intended technical meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

SparseOPD and the comparison baselines are specified in Sections 2 and 3; the datasets, model configurations, evaluation metrics, and benchmark-level results are reported in Section 4. Training splits, optimization and distillation settings, evaluation and decoding protocols, and efficiency-measurement procedures are provided in Appendix B.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

Rishabh Agrawal, Jacob Fein-Ashley, and Paria Rashidinejad. Reinforcement learning from rich feedback with distributional DAgger, 2026. URL https://arxiv.org/abs/2606.05152.

Anshumann, Mohd Abbas Zaidi, Akhil Kedia, Jinwoo Ahn, Taehwak Kwon, Kangwook Lee, Haejun Lee, and Joohyung Lee. Sparse logit sampling: Accelerating knowledge distillation in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18085–18108, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.885. URL https://aclanthology.org/2025.acl-long.885/.

Mohammadreza Armandpour, Fatih Ilhan, David Harrison, Ajay Jaiswal, Duc N. M. Hoang, Fartash Faghri, Yizhe Zhang, Minsik Cho, and Mehrdad Farajtabar. Unmasking on-policy distillation: Where it helps, where it hurts, and why, 2026. URL https://arxiv.org/abs/2605.10889.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. arXiv preprint arXiv:2511.21631, 2025. URL https://arxiv.org/abs/2511.21631.

Nicolas Boizard, Kevin El Haddad, Céline Hudelot, and Pierre Colombo. Towards cross-tokenizer distillation: the universal logit distillation loss for LLMs, 2024. URL https://arxiv.org/abs/2402.12030.

Walid Bousselham, Hilde Kuehne, and Cordelia Schmid. VOLD: Reasoning transfer from LLMs to vision-language models via on-policy distillation, 2025. URL https://arxiv.org/abs/2510.23497.

Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russell Webb. Distillation scaling laws. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 5977–6045. PMLR, 2025. URL https://proceedings.mlr.press/v267/busbridge25a.html.

Woohyeon Byeon, Jiwon Jeon, Jeonghye Kim, and Youngchul Sung. Be my tutor: On-policy co-distillation for mutual LLM improvement via peer feedback, 2026. URL https://arxiv.org/abs/2606.14368.

Yishuo Cai, Jiahui Liu, Yuanxin Liu, Haobo Deng, Linli Yao, Yuhao Zheng, Kun Ouyang, Zhimo Li, Ziyue Wang, Xu Sun, Haoli Bai, and Xiaohui Li. Thinking without images: Internalizing visual manipulation with on-policy self-distillation, 2026a. URL https://arxiv.org/abs/2606.08719.

Yuang Cai and Yuyu Yuan. $\mathcal{X}$-KD: General experiential knowledge distillation for large language models, 2026. URL https://arxiv.org/abs/2602.12674.

Yuchen Cai, Ding Cao, Liang Lin, Chunxi Luo, Xin Xu, Kai Yang, Weijie Liu, Saiyong Yang, Tianxiang Zhao, Guangzhong Sun, Guiquan Liu, and Junfeng Fang. Learning to foresee: Unveiling the unlocking efficiency of on-policy distillation, 2026b. URL https://arxiv.org/abs/2605.11739.

Di Cao, Dongjie Fu, Hai Yu, Siqi Zheng, Xu Tan, and Tao Jin. X-OPD: Cross-modal on-policy distillation for capability alignment in speech LLMs, 2026. URL https://arxiv.org/abs/2603.24596.

Tianlei Chen, Jiao Ou, Ziyuan Liu, Ruiming Tang, Jian Liang, and Han Li. Counteraction-aware multi-teacher on-policy distillation for general capability recovery with domain preservation, 2026a. URL https://arxiv.org/abs/2605.27115.

Xianwei Chen, Shimin Zhang, and Jibin Wu. $\boldsymbol{f}$-OPD: Stabilizing long-horizon on-policy distillation with freshness-aware control, 2026b. URL https://arxiv.org/abs/2605.17862.

Jiabin Fan, Guoqing Luo, Michael Bowling, and Lili Mou. KETCHUP: K-step return estimation for sequential knowledge distillation, 2025. URL https://arxiv.org/abs/2504.19024.

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. arXiv preprint arXiv:2406.09098, version 4, 2024. URL https://arxiv.org/abs/2406.09098v4. Version 4 revised in 2025; SciKnowEval V2 dataset: https://huggingface.co/datasets/hicai-zju/SciKnowEval.

Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Guanning-AI. DAPO14k. Hugging Face dataset, n.d. URL https://huggingface.co/datasets/guanning-ai/dapo14k. Accessed September 9, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

Huipeng Huang and Hongxin Wei. Tail-aware top-k on-policy distillation, 2026. URL https://arxiv.org/abs/2608.14728.

Gyeongman Kim, Doohyuk Jang, and Eunho Yang. PromptKD: Distilling student-friendly knowledge for generative language models via prompt tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6266–6282. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.364. URL https://aclanthology.org/2024.findings-emnlp.364/.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 24872–24895. PMLR, 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

Hojae Lee, Junho Kim, and SangKeun Lee. Mentor-KD: Making small language models better multi-step reasoners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17643–17658. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.977. URL https://aclanthology.org/2024.emnlp-main.977/.

Sicong Leng, Jing Wang, Jiaxi Li, Hao Zhang, Zhiqiang Hu, Boqiang Zhang, Hang Zhang, Yuming Jiang, Xin Li, Deli Zhao, Fan Wang, Yu Rong, Aixin Sun, and Shijian Lu. MMR1: Advancing the frontiers of multimodal reasoning. Open-source project, v0 release, 2025. URL https://github.com/LengSicong/MMR1/tree/mmr1_v0. Dataset: https://huggingface.co/datasets/MMR1/MMR1-Math-RL-Data-v0.

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

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing, 2026a. URL https://arxiv.org/abs/2604.02288.

Hao Li, Jingkun An, Zijun Song, Pengyu Zhu, Rui Li, Hao Wang, Wendi Feng, Yesheng Liu, Lijun Li, Jin-Ge Yao, and Lei Sha. SafeSteer: Localized on-policy distillation for efficient safety alignment, 2026b. URL https://arxiv.org/abs/2606.02530.

Jiaze Li, Hao Yin, Haoran Xu, Boshen Xu, Wenhui Tan, Zewen He, Jianzhong Ju, Zhenbo Luo, and Jian Luan. Video-OPD: Efficient post-training of multimodal large language models for temporal video grounding via on-policy distillation, 2026c. URL https://arxiv.org/abs/2602.02994.

Sijia Li, Yuchen Huang, Zifan Liu, Yanping Li, Jingjing Fu, Li Zhao, Jiang Bian, Ling Zhang, Jun Zhang, and Rui Wang. GEAR: Granularity-adaptive advantage reweighting for LLM agents via self-distillation, 2026d. URL https://arxiv.org/abs/2605.11853.

Xin Li, Hao Jiang, Annan Wang, Yichi Zhang, and Chau Yuen. The extrapolation cliff in on-policy distillation of near-deterministic structured outputs, 2026e. URL https://arxiv.org/abs/2605.08737.

Yang Li, Erik Nijkamp, Semih Yavuz, and Shafiq Joty. Learning from language feedback via variational policy distillation, 2026f. URL https://arxiv.org/abs/2605.15113.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe, 2026g. URL https://arxiv.org/abs/2604.13016.

Yuying Li, Leqi Zheng, Yongzi Yu, Wenrui Zhou, Xuchang Zhong, Xing Hu, Jing Jin, Hangjie Yuan, and Tao Feng. Filter, then reweight: Rethinking optimization granularity in on-policy distillation, 2026h. URL https://arxiv.org/abs/2606.02684.

Kun Liang, Clive Bai, Xin Xu, Chenming Tang, Sanwoo Lee, Weijie Liu, Saiyong Yang, and Yunfang Wu. ORBIT: On-policy exploration-exploitation for controllable multi-budget reasoning, 2026. URL https://arxiv.org/abs/2601.08310.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. arXiv preprint arXiv:2305.20050, 2023. URL https://arxiv.org/abs/2305.20050.

Honglin Lin, Zheng Liu, Yun Zhu, Chonghan Qin, Juekai Lin, Xiaoran Shang, Conghui He, Wentao Zhang, and Lijun Wu. MMFineReason: Closing the multimodal reasoning gap via open data-centric methods. arXiv preprint arXiv:2601.21821, 2026. URL https://arxiv.org/abs/2601.21821.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/663bce02a0050c4a11f1eb8a7f1429d3-Abstract-Conference.html.

Math-AI Team. AMC23: American mathematics competitions 2023. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/math-ai/amc23.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Yifan Zhang, Xiao Zong, Yida Xu, Muxi Diao, Zhimin Bao, Chen Li, and Honggang Zhang. We-Math: Does your large multimodal model achieve human-like mathematical reasoning? arXiv preprint arXiv:2407.01284, 2024. URL https://arxiv.org/abs/2407.01284.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. arXiv preprint arXiv:2311.12022, 2023. URL https://arxiv.org/abs/2311.12022.

Haolin Ren, Ziyang Huang, Chenhao Yuan, Jun Zhao, and Kang Liu. Trace-based on-policy distillation for masked diffusion language models, 2026. URL https://arxiv.org/abs/2607.16872.

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

Jie Sun, Mao Zheng, Mingyang Song, Qiyong Zhong, Gengsheng Li, Zhepei Hong, Chang Wu, Pengfei Liu, Junfeng Fang, and Xiang Wang. EasyOPD: An easy-to-use on-policy distillation framework for large language models, 2026. URL https://arxiv.org/abs/2607.11012.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with MATH-Vision dataset. arXiv preprint arXiv:2402.14804, 2024. URL https://arxiv.org/abs/2402.14804.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation, 2026. URL https://arxiv.org/abs/2605.26844.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking kullback–leibler divergence in knowledge distillation for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5737–5755. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-main.383/.

Yecheng Wu, Song Han, and Hai Cai. Lightning OPD: Efficient post-training for large reasoning models with offline on-policy distillation, 2026. URL https://arxiv.org/abs/2604.13010.

Haoran Xin, Anhao Zhao, Ying Sun, Jin Li, Xiaoyu Shen, and Hui Xiong. Escaping the KL agreement trap in on-policy distillation, 2026. URL https://arxiv.org/abs/2606.09471.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust region on-policy distillation, 2026. URL https://arxiv.org/abs/2606.01249.

Jing Xiong, Hui Shen, Shansan Gong, Yuxin Cheng, Jianghan Shen, Chaofan Tao, Haochen Tan, Haoli Bai, Lifeng Shang, and Ngai Wong. OVD: On-policy verbal distillation, 2026. URL https://arxiv.org/abs/2601.21968.

Haoran Xu, Hongyu Wang, Yifei Gao, Jiaze Li, Xiaofeng Zhang, and Xiaosong Yuan. SG-OPD: Sign-gated on-policy distillation via sign-consistency gating and phased teacher sampling, 2026. URL https://arxiv.org/abs/2606.09304.

Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. KDRL: Post-training reasoning LLMs via unified knowledge distillation and reinforcement learning, 2025a. URL https://arxiv.org/abs/2506.02208.

Shicheng Xu, Liang Pang, Yunchang Zhu, Jia Gu, Zihao Wei, Jingcheng Deng, Feiyang Pan, Huawei Shen, and Xueqi Cheng. RLKD: Distilling LLMs' reasoning via reinforcement learning, 2025b. URL https://arxiv.org/abs/2505.16142.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025. URL https://arxiv.org/abs/2505.09388.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models, 2025. URL https://arxiv.org/abs/2511.10643.

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

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality check on the evaluation of large multimodal models. arXiv preprint arXiv:2407.12772, 2024a. URL https://arxiv.org/abs/2407.12772.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? arXiv preprint arXiv:2403.14624, 2024b. URL https://arxiv.org/abs/2403.14624.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. Dual-space knowledge distillation for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18164–18181. Association for Computational Linguistics, 2024c. doi: 10.18653/v1/2024.emnlp-main.1010. URL https://aclanthology.org/2024.emnlp-main.1010/.

Songming Zhang, Xue Zhang, Tong Zhang, Bojie Hu, Yufeng Chen, and Jinan Xu. AlignDistil: Token-level language model alignment as adaptive policy distillation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19791–19807. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.972. URL https://aclanthology.org/2025.acl-long.972/.

Yifan Zhang and Math-AI Team. American invitational mathematics examination (AIME) 2025. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/math-ai/aime25.

Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. Revisiting knowledge distillation for autoregressive language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10900–10913. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.587. URL https://aclanthology.org/2024.acl-long.587/.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. DistillSpec: Improving speculative decoding via knowledge distillation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8766fbc68e1ed1cdef712ce273e0a363-Abstract-Conference.html.

A More Related Work

On-policy language model distillation. Knowledge distillation for generative language models (Zhong et al., 2024; Kim et al., 2024; Zhang et al., 2024c; Lee et al., 2024; Gemma Team, 2024; Boizard et al., 2024; Busbridge et al., 2025; Wu et al., 2025; Zhang et al., 2025) has increasingly moved from matching teacher outputs on fixed data toward training on the student's own generations (Ko et al., 2024; Zhou et al., 2024; Ye et al., 2025; Fan et al., 2025; Xu et al., 2025a;b; Bousselham et al., 2025). MiniLLM (Gu et al., 2024) studies reverse KL as a distillation objective for autoregressive language models, while GKD (Agarwal et al., 2024) develops on-policy distillation by applying teacher feedback to student-generated trajectories, reducing the mismatch between training and inference distributions. Recent work has further extended on-policy distillation to settings such as context distillation and diffusion language models (Ye et al., 2026; Ren et al., 2026; Song & Zheng, 2026; Sun et al., 2026; Agrawal et al., 2026; Byeon et al., 2026; Cai et al., 2026a; Cai & Yuan, 2026; Cai et al., 2026b; Cao et al., 2026; Li et al., 2026c; Liang et al., 2026; Xiong et al., 2026). These methods primarily study how teacher supervision should be obtained or applied on on-policy trajectories (Armandpour et al., 2026; Chen et al., 2026a;b; Li et al., 2026a;d;e;f;g; Wu et al., 2026; Xin et al., 2026; Li et al., 2026h; Wang et al., 2026; Xing et al., 2026; Xu et al., 2026). Our work instead focuses on the computational realization of token-level full-vocabulary supervision: we ask whether the complete reverse-KL correction can be preserved without keeping the complete vocabulary differentiable.

Sparse vocabulary approximation in distillation. A complementary line of work reduces the cost of distribution-level distillation by restricting or sampling the vocabulary used for supervision (Anshumann et al., 2025; Huang & Wei, 2026; Li et al., 2026b). Sparse Logit Sampling (Anshumann et al., 2025) shows that naively retaining only a small set of teacher logits can introduce bias, and uses importance sampling to recover the dense distillation gradient in expectation. Recent TopK OPD methods further show that renormalizing teacher and student distributions on a restricted support can discard useful tail information, and augment the retained support to better represent that missing mass (Huang & Wei, 2026). SparseOPD takes a different route: it does not approximate the teacher–student comparison before the full correction is formed. Instead, it first constructs the deterministic full-vocabulary correction from the complete distributions, and only then compresses the correction for sparse differentiation. This separation lets support selection depend directly on correction magnitude and preserves aggregate positive and negative correction mass, while limiting backpropagation to a small set of output-head entries.

B Additional Experimental Details

This section specifies the data splits, student adaptation, optimization, and evaluation protocols used for the main accuracy experiments, followed by a separate efficiency study.

B.1 Training Data and Model Adaptation

Training splits. Mathematics uses the complete training split of DAPO14k. For chemistry, we use the released seed-42 split of the Chemistry L3 multiple-choice questions in SciKnowEval V2: a fixed 90%/10% partition into training and held-out questions. Multimodal training uses the original MMR1-Math-RL-Data-v0 release. Its validation split is excluded from training. Table 7 gives the data usage and task-specific settings.

Student adaptation. Each method starts from the same student checkpoint within a model–task pair. We train low-rank adaptation (LoRA) parameters (Hu et al., 2021). For multimodal students, the vision tower remains frozen, and no bias parameters are trained. Training hardware, per-run training time, and shared hyperparameters are listed in Table 6.

Prompting. All tasks use the student model's native Qwen chat template with thinking mode disabled. Mathematics prompts request a step-by-step solution and a boxed final answer. Chemistry uses a chemistry-expert instruction, with the same prompt construction at training and evaluation.

Table 6: Shared settings for the main accuracy experiments.

Setting	Value
Optimizer	AdamW (Loshchilov & Hutter, 2019)
Initial learning rate / weight decay	10^{-5} / 0
Maximum gradient norm	1.0
Training hardware	8 NVIDIA H20 GPUs
Global batch size	64 trajectories
Training time per run	1.8–4 hours
Training completions per prompt	1
Precision / gradient checkpointing	BF16 / enabled
LoRA rank / scaling parameter	64 / 128
Maximum prompt length	1,024 tokens
Rollout temperature / top-p	1.0 / 1.0
Rollout top-k restriction	Disabled
Training seed	42
Rollout engine / tensor parallelism	Colocated vLLM / 1 per rank
Attention implementation	FlashAttention-2

Table 7: Task-specific training settings.

Setting	Mathematics	Chemistry	Multimodal
Student sizes	1.7B, 4B	1.7B, 4B	2B, 4B
Training questions	Full DAPO14k train split	1,890	5,782
Training epochs	1	3	1
Maximum response tokens	2,048	512	2,048
Distillation temperature	1.0	1.0	1.0

B.2 Optimization and Distillation

Training implementation. Text experiments use Accelerate with DeepSpeed ZeRO-2 and CPU optimizer offloading. We use the MiniLLM framework to implement on-policy distillation (OPD) training.

Distillation protocol. For every method, the student generates one on-policy completion and the frozen teacher scores that same completion by teacher forcing. Full Vocabulary computes the full-vocabulary reverse Kullback–Leibler divergence from student to teacher. TopK uses the student's Top-20 support and renormalizes both distributions within that support. Sampled Token uses the difference between the sampled action's student and teacher log-probabilities as a score-function coefficient, without a baseline. SparseOPD uses a batch-global budget of $20N$, where N is the number of valid response positions, with at most 32 retained coordinates per sign at each position. Its sparse output-head row block size is 32, and its full-vocabulary planning chunk size is 4,096.

B.3 Evaluation Protocol

Table 8 gives the decoding settings shared by all methods within each task. For mathematics and chemistry, Mean@8 averages the binary correctness of eight independently generated answers for each question, then averages over questions. We keep answer extraction and grading consistent across methods within each benchmark. For multimodal benchmarks, we follow the LMMs-Eval evaluation protocol (Zhang et al., 2024a), extracting answers and grading them by exact matching, with one generation per question. Each evaluation split is evaluated in full.

Table 8: Evaluation decoding parameters. All tasks disable the model's thinking mode and impose no top-k sampling restriction.

Setting	Mathematics	Chemistry	Multimodal
Generations per question	8	8	1
Temperature	0.6	0.6	1.0
Top-p	0.9	0.9	0.9
Maximum prompt tokens	1,024	1,024	1,024
Maximum output tokens	4,096	512	2,048

B.4 Efficiency Measurements

Experimental design. We profile Qwen3-4B-Base with Qwen3-4B as the frozen teacher. This study uses full-parameter updates, separately from the LoRA accuracy experiments. Each method processes the same data. Table 9 lists the protocol.

Table 9: Protocol for the efficiency measurements.

Setting	Value
Hardware	Two H200 GPUs
Teacher / student	Qwen3-4B / Qwen3-4B-Base
Updated parameters	All student parameters
Measured completion lengths	2,048 / 4,096 / 8,192 / 16,384 tokens
Global batch / microbatch per GPU	64 / 1 sequence
Gradient accumulation	32 microbatches per GPU
Measured updates per configuration	1
TopK / average SparseOPD support budget	20 / 20 coordinates per valid position

Timing boundaries. Forward time covers the loss-producing forward calls accumulated over the update; backward time covers the corresponding autograd calls, including recomputation and communication but excluding the optimizer. Each stage is summed within each GPU, and we report the larger of the two GPU totals. Full Vocabulary evaluates both student and teacher in the forward stage. SparseOPD evaluates the teacher, computes full-vocabulary statistics, and constructs its sparse plan beforehand.

Memory definition. Let M_r^{start} be GPU r's Torch allocated memory at step start, and $M_r(t)$ its allocation during a stage ϕ. The main table reports

$$M_{\text{extra}}^{\phi} = \max_r \left\{ \max_{t \in \phi} M_r(t) - M_r^{\text{start}} \right\}.$$

We use the same pre-step baseline for forward and backward, retaining the memory of tensors saved by the forward pass for backward. One GiB equals 2^{30} bytes. This metric measures a net increase above existing allocations.

Planning cost and implementation. At 8K sequence length, SparseOPD reduces peak extra backward memory by 70.5% relative to Full Vocabulary with only a 14.5% increase in complete-update time. The time outside forward and backward also includes input preparation and optimizer work, so it cannot all be attributed to token selection. Within backward, selected hidden and weight rows must still be materialized, dense parameter gradients accumulated, and the full Transformer backbone backpropagated through, which limits the corresponding time reduction.

B.5 Parameter Sensitivity Analysis

Multiple seeds. Table 11 evaluates seed and budget sensitivity on 1.7B mathematical reasoning under a fixed evaluation protocol. At the default budget, SparseOPD consistently achieves strong

Table 10: **Efficiency results for all four methods.** Memory columns are total peak Torch allocations during each stage, before baseline subtraction; the main table instead reports extra memory. Step time covers the complete update window, including vocabulary sparsification planning. Bold marks the lowest measured value at each length, including ties.

Length	Method	Forward		Backward		Step (s)
		Time (s)	Peak (GiB)	Time (s)	Peak (GiB)	
2K	Full Vocabulary	6.25	84.82	20.49	83.46	33.31
	Sampled Token	6.51	82.48	20.18	**81.55**	32.86
	TopK	5.82	81.27	**18.64**	**81.55**	**30.14**
	SparseOPD(Ours)	**3.30**	**79.42**	19.07	81.57	37.45
4K	Full Vocabulary	10.64	91.23	28.07	88.50	45.20
	Sampled Token	9.83	86.42	26.11	83.97	42.23
	TopK	9.39	83.99	**24.25**	**81.95**	**40.27**
	SparseOPD(Ours)	**4.96**	**80.35**	24.56	82.00	50.97
8K	Full Vocabulary	19.91	103.91	45.61	98.46	72.04
	Sampled Token	18.79	94.28	42.60	89.40	68.05
	TopK	18.48	89.43	**37.58**	84.76	**61.95**
	SparseOPD(Ours)	**9.14**	**82.29**	39.18	**84.32**	82.50
16K	Full Vocabulary	39.1	129.27	83.9	118.37	129.5
	Sampled Token	39.8	110.02	81.8	100.25	128.9
	TopK	39.3	100.31	**67.1**	90.37	112.5
	SparseOPD(Ours)	**18.6**	**86.18**	73.3	**88.96**	142.7

Table 11: **Parameter sensitivity on 1.7B mathematical reasoning.** Scores are Mean@8 accuracy (%) under the same evaluation protocol; Avg. is the unweighted mean over benchmarks. SparseOPD(Ours) reproduces the main result ($K = 20$, seed 42). The shaded summary reports mean $\pm$ sample standard deviation across seeds 42, 43, and 44, computed from the reported scores. Budget variants use seed 42.

Setting	MATH500	Minerva	AMC23	AIME25	**Avg.**
SparseOPD(Ours)	64.5	26.1	35.9	5.8	33.1
Seed 43 ($K = 20$)	65.10	25.28	34.69	5.42	32.62
Seed 44 ($K = 20$)	65.70	25.60	37.19	3.75	33.06
Mean $\pm$ std	65.10$\pm$0.60	25.66$\pm$0.41	35.93$\pm$1.25	4.99$\pm$1.09	32.93$\pm$0.27
$K = 16$	64.25	26.15	31.25	5.42	31.77
$K = 24$	65.60	26.42	36.25	5.83	33.53

average accuracy across training seeds, with a small standard deviation. This consistency suggests that the main result is not driven by a particularly favorable seed.

Support budget. We vary the average support budget $K \in \{16, 20, 24\}$ while fixing the training seed. SparseOPD maintains strong performance across the tested settings, with larger budgets improving average accuracy and $K = 24$ achieving the best result. MATH500 and Minerva remain relatively stable, whereas AMC23 benefits more from increased support, indicating that budget sensitivity varies across benchmarks while overall performance remains robust.

B.6 CASE STUDY

B.6.1 MATHEMATICS

Question M1

For which n is $n^4 + 6n^3 + 11n^2 + 3n + 31$ a perfect square?

Standard Answer

10.

Question M2

Determine the smallest prime p such that $2018!$ is divisible by p^3, but not divisible by p^4.

Standard Answer

509.

Question M3

Let $ABCD$ be a unit square in the plane. Points X and Y are chosen independently and uniformly at random on the perimeter of $ABCD$. If the expected value of the area of triangle $\triangle AXY$ can be expressed as $\frac{m}{n}$ for relatively prime positive integers m and n, compute $m + n$.

Standard Answer

113.

Question M4

Let a, b, c be distinct numbers such that the equations $x^2 + ax + 1 = 0$ and $x^2 + bx + c = 0$ have a common real root, and the equations $x^2 + x + a = 0$ and $x^2 + cx + b = 0$ also have a common real root. Compute the sum $a + b + c$.

Standard Answer

-3.

Question M5

Four positive integers a, b, c, d satisfy the condition $a < b < c < d$. Determine the smallest possible value of d such that the arithmetic mean of the numbers a, b, c is half of the arithmetic mean of the numbers a, b, c, d.

Standard Answer

10.

B.6.2 CHEMISTRY

Question C1

Determine the pH of the solution of 500 mL of 0.1 M CH_3COOH, 400 mL of 0.2 M HCl, and 300 mL of 0.3 M $Ba(OH)_2$.

A. 8.92 B. 8.68 C. 1.38 D. 12.62.

Standard Answer

D. 12.62.

Question C2

An unknown compound has a very broad Fourier-transform infrared (FTIR) absorption near $3000\,cm^{-1}$ and a strong absorption near $1700\,cm^{-1}$. Its proton nuclear magnetic resonance (^1H NMR) spectrum contains no vinyl-hydrogen signals, but includes a doublet of triplets of quartets and a doublet of triplets of triplets. Identify the compound.

A. $CH_3C(H)(C_2H_5)C(H)(C_2H_5)CH_2COOH$
B. $CH_3CH_2C(H)(C_2H_5)C(H)(C_2H_5)COOH$
C. $CH_3C(H)(CH_3)C(H)(CH_3)CH_2COOH$
D. $CH_3CH_2C(H)(CH_3)C(H)(CH_3)COOH$

Standard Answer

A. $CH_3C(H)(C_2H_5)C(H)(C_2H_5)CH_2COOH$.

Question C3

acetic acid is treated with bromine, pyridine, and acetic anhydride with heating, forming product 1.
1 is heated with ethanol and a small amount of sulfuric acid, forming product 2.
2 is treated with sodium cyanide, forming product 3.
3 is then treated with excess sodium hydride and 1,5-dibromopentane, forming final product 4.
how many distinct hydrogen signals will be observable in the ^1H NMR spectrum of 4? (some of them maybe very close in chemical shift and thus not practically distinguishable, but the desired answer is the number of chemically distinct hydrogens)
A. 10
B. 12
C. 5
D. 8

Standard Answer

D. 8

CHEMISTRY (CONTINUED)

Question C4

A chemical reaction for the synthesis of a product containing H^+ ion was proceeding at room temperature and pH 1. Accidentally an unknown substance was fallen into the running reaction making the rate of the reaction slower for the product formation and the container got hot due to an exothermic reaction. The pH value of the solution changed to 4 after this accidental addition. What can be the possible reason for changing the rate of reaction?
A. The increased volume of the solution
B. The increased pressure of the solution
C. The increased pH of the solution
D. The increased temperature of the solution

Standard Answer

C. The increased pH of the solution

Question C5

You are analyzing a small peptidic compound that has been chemically synthesized. The ^{1}H NMR spectrum of the crude compound appears consistent with the expected molecule, with the exception of two peaks that both correspond to the same alpha-proton. These two peaks have similar chemical shifts and roughly equal integrals (together they integrate to 1H - the expected integral); based on the coupling pattern, you can rule out spin-spin coupling as an explanation for the duplicate peaks. LC-MS analysis of the crude compound at elevated temperature shows two clearly defined peaks of equal intensities. Both peaks have the same mass spectrum, which is consistent with the expected molecule. What is the most likely explanation for these observations?
A. The crude compound exists as a mixture of enantiomers
B. 'Double coupling' has occurred during an amide-bond forming reaction
C. The crude compound exists as a mixture of diastereoisomers
D. The compound is contaminated with a precursor

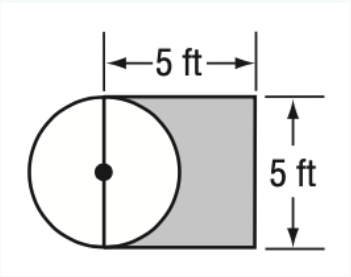
Standard Answer

C. The crude compound exists as a mixture of diastereoisomers

B.6.3 MULTIMODAL GEOMETRY

Question G1

One side of a square is a diameter of a circle. The length of one side of the square is 5 feet. To the nearest hundredth, what is the probability that a point chosen at random is in the shaded region?

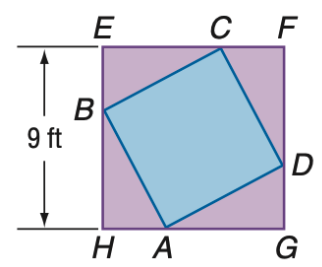


Standard Answer

0.44.

Question G2

In the figure, the vertices of quadrilateral ABCD intersect square EFGH and divide its sides into segments with measures that have a ratio of 1:2. Find the ratio between the areas of ABCD and EFGH.

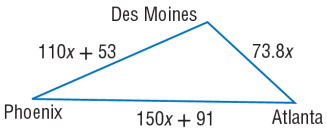


Standard Answer

5 : 9.

Question G3

A plane travels from Des Moines to Phoenix, on to Atlanta, and back to Des Moines, as shown below. Find the distance in miles from Des Moines to Phoenix if the total trip was 3482 miles.



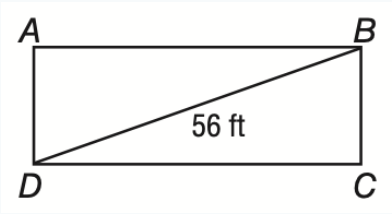
Standard Answer

1153.

MULTIMODAL GEOMETRY (CONTINUED)

Question G4

The diagonals of rectangle $ABCD$ each have a length of 56 feet. If $m\angle BAC = 42°$, what is the length of $\overline{AB}$ to the nearest tenth of a foot?

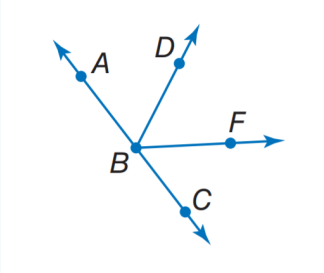


Standard Answer

41.6.

Question G5

$\overrightarrow{BA}$ and $\overrightarrow{BC}$ are opposite rays and $\overrightarrow{BD}$ bisects $\angle ABF$. If $m\angle ABF = 3x - 8$ and $m\angle ABD = x + 14$, find $m\angle ABD$.



Standard Answer

50.