Bounded Exponential Reward Shaping for Adaptive Reasoning Efficiency

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning models often generate lengthy responses even when additional computation offers limited benefit. Uniform length penalties encourage conciseness but do not account for differences in problem difficulty. We study DASE (Difficulty-Adaptive Shaping for Efficiency), a reinforcement learning approach that uses the current rollout group's solve rate to adapt a capped exponential penalty on reasoning length. The penalty encourages stronger compression on reliably solved prompts and relaxes it on lower-success prompts, without requiring a separate difficulty predictor or an inference-time controller. Experiments with Qwen3.5-122B-A10B across science, mathematics, and coding benchmarks show an approximately 37.9% reduction in average response length relative to an unpenalized reference and higher mean Overthinking-Adjusted Accuracy area than GFPO, fixed DLER, and the reference. Parameter ablations compare the shaping settings, and training trajectories show longer reasoning in groups with lower solve rates.

1 INTRODUCTION

Intermediate reasoning expands what a language model can accomplish before producing an answer. Chain-of-thought prompting, self-consistency, and explicit search make different uses of this additional computation (Wei et al., 2022; Wang et al., 2023; Yao et al., 2023). Reinforcement learning has further strengthened long-form reasoning in systems such as DeepSeek-R1, Kimi k1.5, and Qwen3 (Guo et al., 2025; Kimi Team et al., 2025; Yang et al., 2025). However, generating a longer response is not itself a desirable objective. An effective reasoning policy should use additional tokens when they improve its chance of solving a problem, not simply because long trajectories have become common during training.

This distinction matters because the value of computation depends on the problem and the current policy. Work on test-time scaling shows that compute allocation affects the attainable accuracy–cost trade-off (Snell et al., 2025; Muennighoff et al., 2025), while analyses of overthinking document redundant reasoning even on simple questions (Chen et al., 2025). A fixed output limit can terminate useful reasoning; an unrestricted correctness reward provides no explicit incentive to remove unnecessary computation. A length-sensitive training reward offers an intermediate option, but its scale determines whether the model learns concise solutions or merely sacrifices difficult problems.

Difficulty-aware length control uses online estimates of problem difficulty to adjust token costs or generation budgets. Adaptive Length Penalty (ALP) conditions token cost on a prompt's online solve rate (Xiang et al., 2025); LASER-D combines dynamic and difficulty-aware reward shaping (Liu et al., 2026); and Difficulty-Aware DLER adjusts truncation according to difficulty (Liu et al., 2025a). We examine a capped length penalty with an exponential dependence on current-group solve rate, focusing on its effect on per-token costs, group-centered advantages, and the accuracy–length trade-off.

We study DASE, which draws multiple responses for a prompt, measures their original binary correctness, and applies a shared, success-dependent coefficient to their individual reasoning lengths. The exponential mapping weakens compression on low-success groups, while a cap limits the maximum reward deduction. With binary task rewards and a cap below one, a correct response retains a larger shaped reward than any incorrect response. Length differences also affect group-centered advantages, even among correct responses.

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.

Our contributions are as follows:

- We specify DASE, combining first-K candidate selection with solve-rate-dependent, capped length shaping, and analyze its reward ordering and group-centered learning signal.
- We evaluate a single selected configuration against GFPO and DLER on five benchmarks, using accuracy, token consumption, and Overthinking-Adjusted Accuracy (Rakotonirina et al., 2026); parameter alternatives are studied separately as ablations.
- We analyze 6,144 training trajectories to measure how reasoning-length distributions vary with the rollout group's solve rate.

2 Related Work

2.1 Reasoning and computation allocation

Reasoning computation can be allocated within a response or across multiple candidates. Chain-of-thought and least-to-most prompting structure the steps within a response (Wei et al., 2022; Zhou et al., 2023); self-consistency and tree search use multiple candidates to improve answer selection (Wang et al., 2023; Yao et al., 2023). The gains from additional computation depend on how this budget is allocated (Snell et al., 2025; Muennighoff et al., 2025). DASE acts during training, combining a cost on each response's reasoning length with selection among sampled trajectories.

2.2 Policy optimization and trajectory selection

Group-relative optimization constructs advantages from rewards for responses to the same prompt (Shao et al., 2024). Candidate selection determines which responses enter this comparison: GFPO retains a subset of a larger rollout group based on length or reward per token (Shrivastava et al., 2026). Sampling, normalization, and update rules also affect reasoning training, as studied by DAPO, Dr. GRPO, and GSPO (Yu et al., 2025; Liu et al., 2025b; Zheng et al., 2025). For length-shaped rewards, group centering retains differences in response length, while standard-deviation division further rescales the signal. We use group centering without this division and retain all-correct groups, which provide a length signal when their responses have unequal deductions.

2.3 Length control and difficulty estimation

Token-budget-aware reasoning and L1 study explicit control over response length (Han et al., 2025; Aggarwal & Welleck, 2025). DLER studies truncation-based rewards (Liu et al., 2025a), and multi-stage adaptive reasoning restructures traces to discourage unnecessary continuation (Rakotonirina et al., 2026). Difficulty-aware methods additionally adjust the cost of computation across prompts. ALP uses the online solve rate to set token costs (Xiang et al., 2025), while LASER/LASER-D adapt length-based rewards during training and according to difficulty (Liu et al., 2026). Our approach builds on difficulty-aware shaping and the capped exponential penalty of DeepSeek-AI (2026). We use the retained rollout group's solve rate to set the coefficient and study this reward together with first-K selection.

3 DASE: Difficulty-Adaptive Shaping for Efficiency

3.1 Candidate oversampling and first-K selection

For a prompt q, the rollout stage launches M candidate trajectories and constructs a training group of size $K \leq M$. Let t_m be the time at which candidate m is returned, including a return caused by the generation limit. The first-K rule is

$$\begin{aligned} \widetilde{y}^{(m)} &\sim \pi_{\text{old}}(\cdot \mid q), \qquad m = 1, \ldots, M, \\ \sigma &= \text{argsort}_m(t_m), \qquad y^{(k)} = \widetilde{y}^{(\sigma(k))}, \quad k = 1, \ldots, K. \end{aligned} \tag{1}$$

Here σ orders candidates by return time, with candidate index breaking ties. We launch $M = 24$ candidates and retain the first $K = 16$ returns, giving an oversampling ratio $M/K = 1.5$. Selection

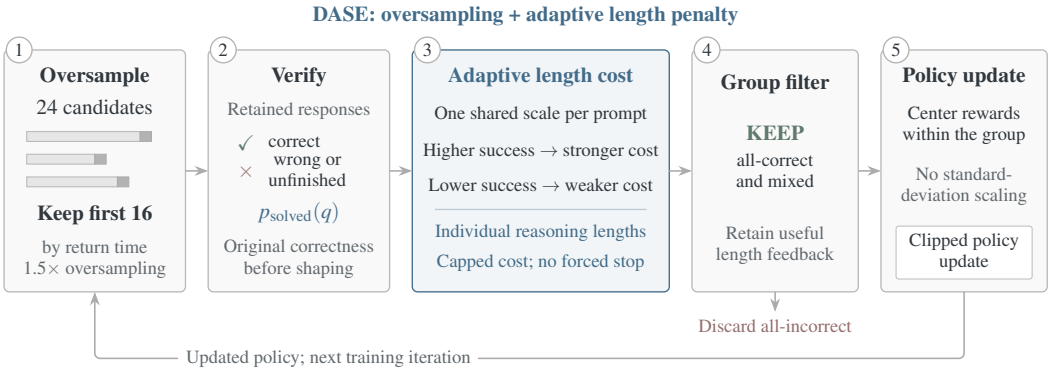

Figure 1: Main workflow of DASE. We launch 24 candidates and retain the first 16 returns (1.5× oversampling). Original correctness determines the shared length-cost scale; individual reasoning lengths determine capped deductions. All-correct and mixed groups enter the group-centered policy update, while all-incorrect groups are discarded. Response bars are schematic, not measured trajectories.

precedes correctness verification and is not a ranking by token count or correctness. The retained group supplies both the solve-rate estimate and the subsequent length-shaping update.

Group-statistic formulas below assume all K retained trajectories are valid. Otherwise, they use only the K_q valid, distinct trajectories (Appendix B). For reference answer y^*, the empirical solve rate is

$$p_{\text{solved}}(q) = \frac{1}{K}\sum_{k=1}^{K} \mathbf{1}\left[\text{answer}(y^{(k)}) = y^*\right]. \tag{2}$$

Equality denotes task-verifier acceptance, including mathematical equivalence or passing the coding tests, rather than literal string identity. A valid retained unfinished response has no accepted final answer and contributes zero, as does a completed wrong answer. Both remain in the denominator; entirely masked trajectories do not. This is a policy- and selection-dependent estimate, not intrinsic difficulty. It uses original correctness, not shaped rewards, and requires no extra samples beyond the retained group.

Candidate selection and reward construction serve different purposes. The former determines which trajectories enter the group; the latter determines their relative training signal. In particular, selecting only correct returns would erase the failures used to estimate $p_{\text{solved}}(q)$. Cancelled or discarded candidates are not automatically members of the retained group, and are not counted as extra negative examples without a trainable returned trajectory.

3.2 Solve-rate-dependent length shaping

We adopt a capped length penalty whose coefficient depends exponentially on the group solve rate (DeepSeek-AI, 2026). The deduction grows linearly with reasoning length until it reaches the cap. We express it through the reasoning length at which its deduction saturates:

$$\mathcal{B}(q) = \frac{\delta L_{\text{ref}}}{\alpha}\exp\left(\frac{1 - p_{\text{solved}}(q)}{\beta}\right). \tag{3}$$

Here $\alpha, \beta, L_{\text{ref}} > 0$ control compression strength, sensitivity, and length units, respectively; $0 < \delta < 1$ bounds the deduction. Lower solve rates enlarge $\mathcal{B}(q)$, reducing the cost per reasoning token. This is a reward-saturation length, not a decoding limit. The selected values are $(\alpha, \beta, \delta, L_{\text{ref}}) = (0.8, 0.3, 0.5, 131{,}072)$; the same notation is used throughout the experimental comparisons.

For reasoning-token count $\ell^{(k)} = \ell(y^{(k)})$ and original task reward $r^{(k)}_{\text{task}}$, define

$$r^{(k)} = \begin{cases} r^{(k)}_{\text{task}} - \delta\,\ell^{(k)}/\mathcal{B}(q), & \ell^{(k)} < \mathcal{B}(q), \\ r^{(k)}_{\text{task}} - \delta, & \ell^{(k)} \geq \mathcal{B}(q). \end{cases} \tag{4}$$

The length penalty applies to both correct and incorrect retained responses. For a 32,768-token trace, the deductions are approximately 0.0071, 0.0378, and 0.20 at solve rates 0, 0.5, and 1, respectively.

Increasing α shortens the saturation length and raises the uncapped token cost. For solve rates below one, decreasing β enlarges that length, weakening compression on poorly solved groups. At perfect observed success, $\mathcal{B}(q) = \delta L_{\mathrm{ref}}/\alpha$ regardless of β. Responses can continue beyond this length with a constant deduction of δ, up to the rollout generation limit.

3.3 Interaction with the policy update

For the selected configuration, we express group centering as the mean pairwise reward difference:

$$a^{(k)} = \frac{1}{K}\sum_{j=1}^{K}\left(r^{(k)} - r^{(j)}\right). \tag{5}$$

This equals $r^{(k)} - \overline{r}$, without within-group standard-deviation division or batch advantage normalization. The actor uses token-level importance weighting and probability-difference masking at ± 0.3, with zero KL and entropy coefficients. Token losses are averaged within response rows, then across nonempty rows; Appendix B.1 specifies the full objective and multirow trajectory correction. When no deduction saturates, the length advantage is $-\delta(\ell^{(k)} - \overline{\ell})/\mathcal{B}(q)$, favoring shorter responses with equal task rewards. All-correct groups can supply a length signal; identical saturated deductions cancel.

The group filter retains all-correct and mixed groups but discards all-incorrect groups. Thus, unsuccessful trajectories within retained mixed groups participate in training, whereas an entirely unsuccessful group does not. This group-level choice is distinct from first-K candidate selection and from the per-trajectory length penalty.

3.4 Reward ordering and estimation limits

For binary task rewards, a correct response receives at least $1 - \delta > 0$, while an incorrect response receives at most zero. Group centering preserves this ordering, but a long correct response can still have negative advantage relative to the group mean. This bound is not a guarantee about optimized benchmark accuracy.

Under an independent Bernoulli approximation, the solve-rate estimate has standard deviation at most $1/(2\sqrt{K}) = 0.125$ for $K = 16$. The per-token cost $\delta/\mathcal{B}(q)$ is convex in that estimate, so sampling noise raises its expectation. Smaller β increases the relative sensitivity of the uncapped token cost to the estimated solve rate. First-K selection with $M > K$ can additionally change the sampled distribution. Appendix A gives the uncapped decomposition and finite-group calculation without assuming that selected trajectories provide an unbiased estimate of unrestricted policy success.

4 Experiments

4.1 Experiment Setup

Model and hardware. All methods use Qwen3.5-122B-A10B and an H200 GPU cluster. Runs start from the same model initialization on a mathematics–code training mixture. DASE uses 128 prompts per update, 24 candidates per prompt with 16 retained responses, a learning rate of 2×10^{-6}, and prompt/response limits of 16,384/131,072 tokens. Training decoding uses temperature 1, top-$p = 1$, and no top-k restriction. Our local verl/HybridFlow and SGLang stack builds on established training and serving infrastructure (Qwen Team, 2026; Sheng et al., 2025; Zheng et al., 2024); efficient serving systems such as vLLM are complementary to token reduction (Kwon et al., 2023).

Training data. We use 6,930 prompt records: 55.25% mathematics-category and 44.75% competitive programming. The merged corpus is shuffled without replacement within each pass. Appendix C provides source names, counts, and exact-duplicate checks.

Evaluation. We evaluate science, mathematics, and coding on five benchmarks. These are GPQA, AIME 2025, HMMT February 2026, AIME 2026, and LiveCodeBench-v6 Python (Rein et al., 2024; Jain et al., 2025), extending the mathematical reasoning settings represented by MATH and GSM8K (Hendrycks et al., 2021; Cobbe et al., 2021). Additional HMMT 2025 results appear in Appendix F. All methods use temperature 1, top-$p = 0.95$, top-$k = 20$, and a 131,072-token response limit. Repeated entries and mathematics variants retain their configured within-benchmark weights; the five benchmarks receive equal weight in aggregate results. Repeats are not treated as independent unique problems.

Accuracy and token consumption. We report sampled-response correctness and mean total response length. Accuracy comes from held-out evaluation records, not training rewards or best-of-n selection. Length is measured on the same benchmark responses and includes final-answer tokens, unlike the reasoning-only training penalty. AIME 2026 combines its configured general and part-I subsets; HMMT February 2026 is not pooled with 2025. Coding results use LiveCodeBench-v6 Python. Aggregation and weighting are described in Appendix D.

Overthinking-Adjusted Accuracy. OAA jointly measures correctness and completion within a token threshold. Following Rakotonirina et al. (2026), for benchmark b with n_b sampled responses, binary correctness $c_{b,i}$, and total response length $L_{b,i}$, define

$$\mathrm{OAA}_b(t) = \frac{1}{n_b} \sum_{i=1}^{n_b} c_{b,i}\, \mathbf{1}[L_{b,i} < t]. \tag{6}$$

This is the fraction of all evaluated responses that are both correct and shorter than threshold t; it is not accuracy conditional on being short. We score already generated responses without truncating or regenerating them. Primary results use total response tokens, matching the reference paper and our benchmark length tables. A reasoning-token variant is reported separately in Appendix I.

We summarize each curve by its normalized area:

$$\mathrm{AUC}_{\mathrm{OAA},b} = \frac{1}{T_b} \int_0^{T_b} \mathrm{OAA}_b(t)\, dt = \frac{1}{n_b} \sum_{i=1}^{n_b} c_{b,i} \max\left(1 - \frac{L_{b,i}}{T_b}, 0\right). \tag{7}$$

The second equality evaluates the empirical step-function integral exactly, avoiding numerical grid approximation. As in Rakotonirina et al. (2026), T_b is the mean length of the initial model on that benchmark, obtained here from the saved initial-model evaluation. These budgets remain fixed across all evaluated methods, and are distinct from the 131,072-token generation limit. We report $100\,\mathrm{AUC}_{\mathrm{OAA},b}$ and its equal-weight five-benchmark mean, with larger values indicating better correct-response efficiency. The previously selected parameters are unchanged; OAA is an additional evaluation metric, not a new selection criterion.

4.2 Baselines and parameter selection

We compare against GFPO (Shrivastava et al., 2026) and the DLER baseline as configured in our training stack (Liu et al., 2025a), as well as an unpenalized reference. GFPO samples 32 responses and retains 16 with stratified length selection, guess exclusion, and a 65,536-token target-length setting. The DLER reward-zeroing threshold exceeds its training response limit and is inactive under that limit; this comparison does not isolate an active truncation penalty. Appendix B specifies the trigger, normalization, and generation settings.

Only one DASE configuration enters the main comparison. We retain $(\alpha, \beta) = (0.80, 0.30)$, originally selected by the highest five-benchmark mean accuracy averaged over two validation evaluations common to the reference and all evaluated parameter settings. This validation-based selection uses the available benchmark histories rather than an independent selection set. The selection protocol is documented in Appendix D; parameter alternatives appear in the ablation study.

4.3 Main benchmark comparison

For accuracy and OAA, bold marks the best length-control method; † marks the best including the unpenalized reference. For length, bold marks the minimum. Ties are included at displayed precision; descriptive statistics are not ranked.

Table 1: Qwen3.5-122B-A10B results: accuracy (%) and mean response length (thousands of tokens). Bold marks the best length-control accuracy or shortest length; † marks the best accuracy including the unpenalized reference. Avg. equally weights the five benchmarks, using unrounded values. LCB is LiveCodeBench-v6 Python.

Method	GPQA	AIME25	HMMT26	AIME26	LCB	Avg.
			Accuracy ↑			
Unpenalized	86.26†	92.71†	82.58†	93.33†	80.71	87.12†
GFPO	**85.66**	90.42	74.24	92.08	**81.14**†	84.71
DLER	85.56	**91.46**	76.89	88.33	78.71	84.19
DASE	84.80	91.25	**77.65**	**93.33**†	79.71	**85.35**
			Response length ↓			
Unpenalized	30.1	40.1	52.4	40.3	50.3	42.6
GFPO	**17.6**	28.4	34.6	28.9	33.3	28.5
DLER	19.3	27.9	34.3	28.1	36.6	29.2
DASE	17.9	**25.2**	**33.2**	**25.1**	**31.0**	**26.5**

Table 2: Normalized response-token $\mathrm{AUC}_{\mathrm{OAA}}$, multiplied by 100. Each benchmark uses a common integration budget fixed from the initial model; Avg. is the five-benchmark mean. Bold marks the best length-control method; † marks the best result including the unpenalized reference.

Method	GPQA	AIME25	HMMT26	AIME26	LCB	Avg.
Unpenalized	21.88	14.05	15.66	14.34	17.92	16.77
GFPO	**38.51**†	24.53	25.05	24.87	26.78	27.95
DLER	35.25	27.19	28.13	28.87	22.48	28.38
DASE	38.25	**33.32**†	**31.21**†	**34.66**†	**32.25**†	**33.94**†

Table 1 shows that DASE reduces the benchmark-averaged response length from approximately 42.65k to 26.48k tokens, a 37.9% reduction, while mean accuracy decreases from 87.12% to 85.35%. Means, savings, and differences are computed before rounding. Compared with GFPO, the selected method is shorter by approximately 7.2% and has 0.64 percentage points higher mean accuracy. Compared with fixed DLER, it is shorter by approximately 9.5% and has 1.16 points higher mean accuracy. Thus, DASE achieves the shortest mean response length and highest mean accuracy among the length-control methods in the main comparison, while remaining less accurate than the unpenalized reference.

Performance varies across benchmarks. DASE preserves the reference's AIME 2026 accuracy of 93.33% while reducing its mean response length from approximately 40.3k to 25.1k tokens. On HMMT February 2026, its accuracy is 77.65%, compared with 74.24% for GFPO and 76.89% for fixed DLER, while its mean response is shorter than either. However, GFPO is more accurate on GPQA (85.66% versus 84.80%) and coding (81.14% versus 79.71%), while fixed DLER is more accurate on AIME 2025. This variation motivates reporting each benchmark separately rather than interpreting the mean as a uniform improvement.

The validation histories in Appendix E provide additional training context. We report generated-token savings, not measured inference speedups or reductions in GPU-hours. GFPO samples larger rollout groups, and DLER uses a different training generation limit, so these comparisons do not imply equal training compute.

4.4 Joint accuracy–efficiency evaluation

We apply the OAA protocol defined in Section 4.1 to the evaluated methods, without reselecting parameters for this metric.

Table 2 shows that DASE attains the highest mean $\mathrm{AUC}_{\mathrm{OAA}}$ in the main comparison, at 33.94, improving on the unpenalized reference by 17.17 points, GFPO by 5.99 points, and fixed DLER by 5.56 points. Unlike separate means, this measure rewards short responses only when they are

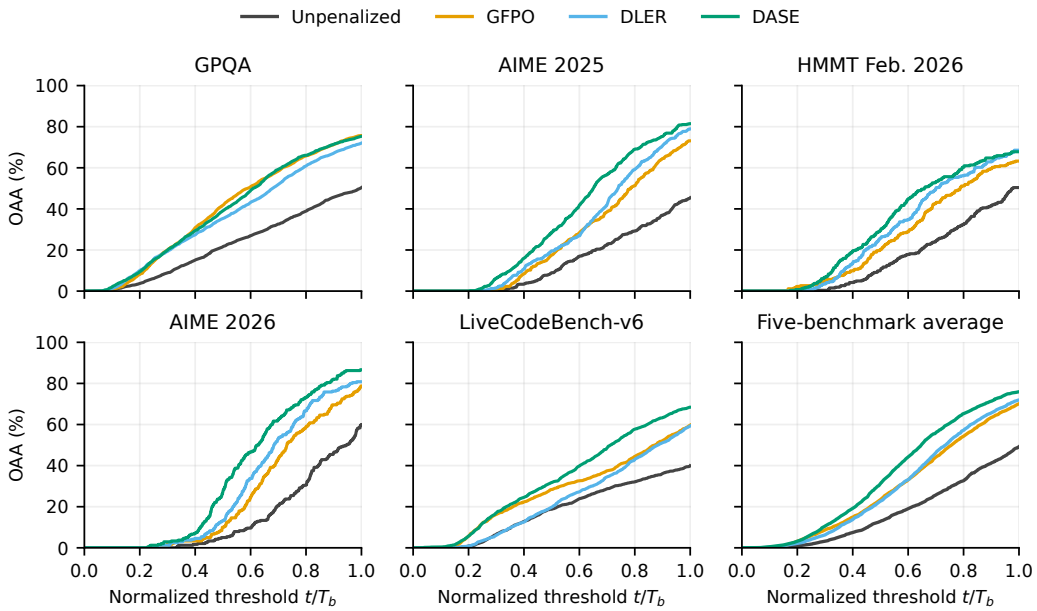

Figure 2: OAA curves for the unpenalized reference, GFPO, DLER, and DASE against normalized token threshold t/T_b for the five benchmarks and their equal-weight mean. A higher curve counts more correct responses within that token threshold.

Table 3: Mean response tokens conditional on correctness. Conditional means are computed within each benchmark and then averaged equally across the five benchmarks; they do not replace individual trajectories in the OAA calculation.

Method	Correct-response tokens	Incorrect-response tokens
Unpenalized	36,819	84,450
GFPO	25,702	44,679
DLER	25,436	49,957
DASE	22,712	49,419

correct: fixed DLER has lower mean accuracy and longer mean responses than GFPO, yet a slightly larger OAA area.

OAA curves appear in Figure 2. The reasoning-only variant gives DASE 34.66 versus 17.07 for the unpenalized reference, 28.86 for GFPO, and 29.05 for DLER. Response length and correctness are negatively correlated on all five benchmarks; for DASE, the correlations range from -0.697 to -0.373 (Appendix I).

Table 3 separates token consumption by correctness. For DASE, correct responses average 22,712 tokens and incorrect responses average 49,419 tokens, with conditional means computed within benchmarks and then averaged equally. Both are shorter than the corresponding unpenalized responses. These comparisons cover different solved subsets because each method can succeed on different questions. The much longer incorrect responses show how failed attempts contribute to total token consumption.

The integration budget determines which successful completions receive credit. A correct response of length $T_b/2$ receives half credit, whereas one at or above T_b contributes zero area. A larger AUC therefore need not imply higher unrestricted accuracy. Fixed initialization-based budgets prevent method-dependent normalization, but the metric remains budget-dependent. We report full curves alongside accuracy and length rather than treating the scalar area as a universal ranking.

Table 4: Parameter ablation with $\delta = 0.5$ and $L_{\text{ref}} = 131{,}072$. Eval.: benchmark results; Select.: accuracy averaged over two selection evaluations; Rollout: training means (Appendix D). Lengths count response tokens, with evaluation means computed before rounding. Bold marks column-wise optima.

α	β	Eval. acc.	Eval. length	Select. acc.	Rollout acc.	Rollout length
1.00	0.25	85.11	24,250	84.80	**53.31**	38,194
0.80	0.30	**85.35**	26,477	**85.60**	50.26	41,533
1.10	0.20	83.46	**22,809**	83.78	50.08	**32,926**

4.5 Parameter ablation

Table 4 separates the choice of penalty parameters from the baseline comparison. The selected $(\alpha, \beta) = (0.80, 0.30)$ setting attains the highest validation selection score. The $(1.00, 0.25)$ setting gives shorter responses and slightly lower mean accuracy, whereas $(1.10, 0.20)$ compresses more aggressively with a larger accuracy decrease. Since compression strength and sensitivity both vary, this is a joint parameter ablation rather than an isolated estimate of either parameter's effect.

Over the training interval summarized in Appendix G, the selected setting averages 41,533 rollout tokens versus 77,805 for the unpenalized reference, with raw rollout correctness of 50.26% versus 53.04%. These training metrics are distinct from the benchmark accuracies used for selection. Across the four training configurations documented in that appendix, only approximately 0.005%–0.023% of trajectories reach the cap on average in this interval, so most updates remain in the linear-in-length part of Equation 4. The appendix also provides the coefficient curves and parameter-specific training dynamics.

4.6 Reasoning-length distributions across difficulty

To examine difficulty-dependent token allocation, we extract the original correctness labels and generated token IDs from the selected configuration's saved training batches at steps 90, 100, and 110. After deduplicating trajectory identifiers within each query group, this yields 384 groups of 16 responses, or 6,144 trajectories. Difficulty is measured by the current group's original solve rate: easy groups have $p_{\text{solved}}(q) \geq 0.75$, medium groups have $0.25 \leq p_{\text{solved}}(q) < 0.75$, and hard groups have $p_{\text{solved}}(q) < 0.25$. Reasoning length counts generated tokens before the last end-of-thinking marker, consistent with the reward definition. This analysis includes both correct and incorrect responses.

Figure 3 shows that easy groups concentrate more of their probability mass at shorter reasoning lengths. Their mean reasoning length is 37,836 tokens, compared with 43,728 for medium groups and 44,432 for hard groups. Hard groups therefore use 17.4% more reasoning tokens than easy groups on average. The corresponding medians are 35,976, 41,316, and 42,066 tokens, so the difference is not driven only by a few extreme long trajectories. The three bins contain 2,048, 2,848, and 1,248 trajectories, respectively.

The medium and hard distributions overlap substantially, with similar means and medians. Both bins cover a broad range of reasoning lengths despite their different solve rates.

4.7 Scope of the evidence

The difficulty analysis measures success and length in the same rollout groups, so their association does not establish that shorter reasoning causes higher accuracy. OAA scores completed responses and does not predict the outcome of forcibly stopping generation or identify an optimal per-question budget. Baseline sampling, filtering, and normalization choices also differ, and parameter selection uses the reporting benchmarks. The comparisons therefore describe the evaluated training configurations; small score differences are not statistically established rankings.

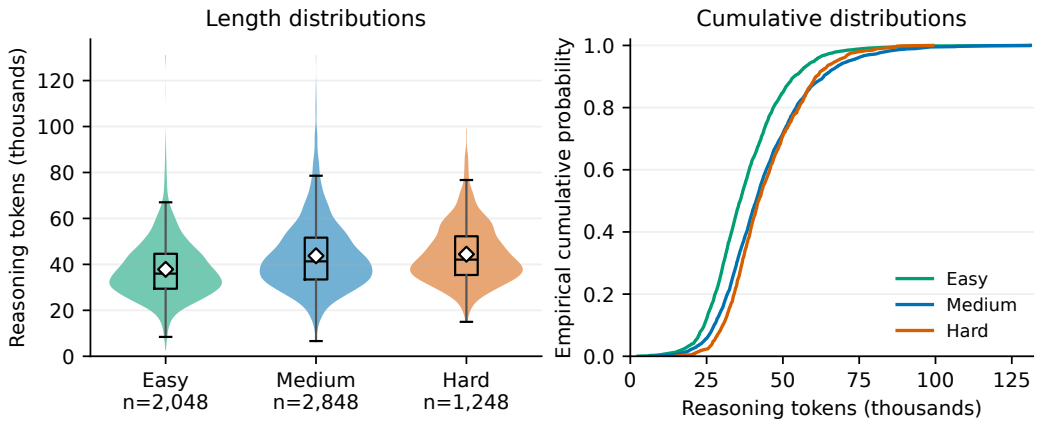

Figure 3: Reasoning-token distributions for the selected DASE configuration across observed difficulty levels. Left: violin distributions with interquartile boxes, median lines, 1.5-IQR whiskers, and diamonds for means. Right: empirical cumulative distributions. Counts denote trajectories, not independent prompts. Easy, medium, and hard correspond to group solve rates $[0.75, 1]$, $[0.25, 0.75)$, and $[0, 0.25)$, respectively.

5 Conclusion

DASE uses the rollout group's solve rate to scale a capped length penalty. On five benchmarks, the selected configuration reduces mean response length by 37.9% relative to the unpenalized reference and achieves a mean response-token OAA area of 33.94, compared with 27.95 for GFPO and 28.38 for the DLER baseline as configured in our training stack. The parameter setting with the highest validation accuracy differs from the one with the highest OAA area. Responses in lower-success training groups are longer on average, although the length distributions overlap substantially.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing, grammar checking, and template verification. We have reviewed all AI-assisted work.

REPRODUCIBILITY AND CODE AVAILABILITY STATEMENT

Section 3 specifies the reward and optimization procedure. Section 4 and the appendices document the model, hardware configuration, hyperparameters, decoding settings, benchmark aggregation, parameter selection, and OAA computation. The experimental implementation is currently confidential and cannot be provided with this submission. We intend to release the code after the internal review and approval process is completed. Until then, this paper provides the experimental configuration and evaluation protocol rather than a public code artifact.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling How Long A Reasoning Model Thinks With Reinforcement Learning. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=4jdIxXBNve.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT Think That Much for 2+3=? On the Overthinking of Long Reasoning Models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 9487–9499. PMLR, 2025. URL https://proceedings.mlr.press/v267/chen25bx.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the Limits of KV Cache Compression. *arXiv preprint arXiv:2609.19969*, 2026. URL https://arxiv.org/abs/2609.19969.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-Budget-Aware LLM Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24842–24855. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1274. URL https://aclanthology.org/2025.findings-acl.1274/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. In *International Conference on Learning Representations*, pp. 58791–58831, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/94074dd5a072d28ff75a76dabed43767-Paper-Conference.pdf.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, et al. Kimi k1.5: Scaling Reinforcement Learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025. URL https://arxiv.org/abs/2501.12599.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626. Association for Computing Machinery, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. DLER: Doing Length pEnalty Right - Incentivizing More Intelligence per Token via Reinforcement Learning. *arXiv preprint arXiv:2510.15110*, 2025a. URL https://arxiv.org/abs/2510.15110.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to Reason Efficiently with Adaptive Length-based Reward Shaping. In *International Conference on Learning Representations*, pp. 43279–43308, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/47795c4ae2f7d07ea2fb0d11fa2c3c90-Paper-Conference.pdf.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-Like Training: A Critical Perspective. In *Conference on Language Modeling*, 2025b. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Qwen Team. Qwen3.5-122B-A10B Model Card, 2026. URL https://huggingface.co/Qwen/Qwen3.5-122B-A10B. Official model card.

Nathanaël Carraz Rakotonirina, Ren Pang, Neha Anna John, Michael Bohlke-Schneider, and Momchil Hardalov. Correct, Concise and Complete: Multi-stage Training For Adaptive Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 12798–12810. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.622. URL https://aclanthology.org/2026.findings-acl.622/.

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. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297. Association for Computing Machinery, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Vaishnavi Shrivastava, Ahmed H Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. Sample More to Think Less: Group Filtered Policy Optimization for Concise Reasoning. In *International Conference on Learning Representations*, pp. 98439–98462,

2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a00ef870c140f09263ba2288855be50b-Paper-Conference.pdf.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning. In *International Conference on Learning Representations*, pp. 10131–10165, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1b623663fd9b874366f3ce019fdfdd44-Paper-Conference.pdf.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. Just Enough Thinking: Efficient Reasoning with Adaptive Length Penalties Reinforcement Learning. *arXiv preprint arXiv:2506.05256*, 2025. URL https://arxiv.org/abs/2506.05256.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822. Curran Associates, Inc., 2023. doi: 10.52202/075280-0517. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4277440d50f1f15d2cb4c14f7e0c0d2-Paper-Conference.pdf.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group Sequence Policy Optimization. *arXiv preprint arXiv:2507.18071*, 2025. URL https://arxiv.org/abs/2507.18071.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient Execution of Structured Language Model Programs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 62557–62583. Curran Associates, Inc., 2024. doi: 10.52202/079017-2000. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/724be4472168f31ba1c9ac630f15dec8-Paper-Conference.pdf.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-Most Prompting Enables

Complex Reasoning in Large Language Models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.

A Analytical Properties

When every response in a group is below the cap, subtracting the group mean in Equation 5 gives

$$a^{(k)} = \left(r_{\text{task}}^{(k)} - \overline{r}_{\text{task}}\right) - \frac{\delta}{\mathcal{B}(q)}\left(\ell^{(k)} - \overline{\ell}\right). \tag{8}$$

The bars denote arithmetic means over the same valid, distinct retained trajectories for q: all K trajectories when valid, or the K_q trajectories defined in Appendix B otherwise. This identity explains why a query-shared length scale remains effective: the length term is centered, rather than removed. A constant length deduction would cancel, but unequal response lengths do not. When only some responses are capped, the actual capped deductions must instead be centered.

For binary rewards, consider a correct response and an incorrect response from one group. Writing the nonnegative deduction as $d = r_{\text{task}} - r \in [0, \delta]$, their reward gap obeys

$$r_{\text{correct}} - r_{\text{incorrect}} = 1 - d_{\text{correct}} + d_{\text{incorrect}} \geq 1 - \delta. \tag{9}$$

The lower bound is strictly positive when $\delta < 1$. It relies on the task rewards being exactly one and zero; additional task penalties or differently scaled rewards require a bound based on the smallest task-reward gap. It does not imply that every correct response has positive centered advantage.

To quantify estimation sensitivity, suppose the K retained correctness labels are independent Bernoulli draws with probability $s(q)$. Define the per-token cost $w(q) = \delta/\mathcal{B}(q)$. The binomial moment-generating function yields

$$\mathbb{E}[w(q) \mid s(q)] = \frac{\alpha e^{-1/\beta}}{L_{\text{ref}}}\left(1 - s(q) + s(q)e^{1/(K\beta)}\right)^K \geq \frac{\alpha}{L_{\text{ref}}} e^{-(1-s(q))/\beta}. \tag{10}$$

The left side is the expected per-token cost after observing a finite rollout group. The right side evaluates it at the population solve probability. Jensen's inequality gives the direction of the gap. With fewer valid trajectories, the same calculation replaces K by K_q, conditional on $K_q > 0$ and the independent Bernoulli assumption for those valid labels. The corresponding standard-deviation bound is $1/(2\sqrt{K_q})$; the main-text value 0.125 applies when $K_q = 16$. Correlated responses, validity filtering, first-K selection, and the dependence between a response's correctness and its group's estimate limit this approximation. In particular, with $M > K$, the retained distribution need not equal the original policy distribution. This is not an unbiased-policy-gradient claim.

The relative sensitivity of the uncapped token cost is $\partial \log w(q)/\partial p_{\text{solved}}(q) = 1/\beta$. Its absolute sensitivity is $\partial w(q)/\partial p_{\text{solved}}(q) = w(q)/\beta$, while $\partial \mathcal{B}(q)/\partial p_{\text{solved}}(q) = -\mathcal{B}(q)/\beta$. Thus, smaller β increases relative sensitivity, but need not increase absolute sensitivity because $w(q)$ also depends on β. For the selected parameters, an all-correct group has $\mathcal{B}(q) = 81{,}920$ reasoning tokens, while a group with solve rate 0.5 has approximately 433,725 tokens, beyond the configured generation limit. This explains why saturation is restricted to long, high-success responses even though the per-token cost remains nonzero on all groups.

B Implementation and Baseline Configurations

The implementation computes reasoning length from generated, unmasked response tokens before the last unmasked end-of-thinking marker. If that marker is absent, it counts all generated response tokens as unfinished reasoning. Prompt tokens, tool text outside the response mask, and padding do not count toward the length. This operational convention measures generated reasoning tokens, not the semantic amount of useful reasoning.

Trajectories are identified by query and trajectory identifiers. The reward implementation counts a trajectory's correctness once when estimating group success, sums generated lengths over distinct real rows, and excludes distributed padding copies through row identifiers. Let $\mathcal{V}_q$ index the distinct retained trajectories with at least one unmasked generated response token, and let $K_q = |\mathcal{V}_q| \leq K$. Entirely masked trajectories do not enter $\mathcal{V}_q$, whereas valid wrong or unfinished responses do. For

$K_q > 0$, the solve rate and centered advantage are

$$p_{\text{solved}}(q) = \frac{1}{K_q} \sum_{k \in \mathcal{V}_q} r_{\text{task}}^{(k)},$$
$$a^{(k)} = r^{(k)} - \frac{1}{K_q} \sum_{j \in \mathcal{V}_q} r^{(j)}, \qquad k \in \mathcal{V}_q. \tag{11}$$

Here $r_{\text{task}}^{(k)}$ is binary verifier correctness. These reduce to Equations 2 and 5 when all $K = 16$ retained trajectories are valid. A group with $K_q = 0$ has no response tokens contributing to the actor loss; these group-statistic formulas are not evaluated for an empty set. The nominal first-K selection size and oversampling ratio remain $K = 16$ and $M/K = 1.5$. The trajectory-level reward adjustment is broadcast to its rows and added at the final unmasked response-token position in each row, consistent with the local trajectory-aware convention.

The rollout stage launches 24 candidates per prompt and retains 16 using first-K selection, giving an oversampling ratio of 1.5. First-K denotes return order, not an explicit sort by token length. A retained unfinished response is a negative example; a cancelled, discarded, or entirely masked candidate is not automatically an additional training example. The empirical solve rate $p_{\text{solved}}(q)$ and the parameters $(\alpha, \beta, \delta, L_{\text{ref}})$ use the notation in Section 3 throughout the tables and difficulty analysis.

B.1 Policy objective and loss aggregation

The actor uses the `dppo` policy-gradient objective. Let i index a response row, t a token position, $h_{i,t}$ its conditioning history, and $m_{i,t} \in \{0, 1\}$ the generated-response mask. Define $n_i = \sum_t m_{i,t}$ and let $\mathcal{R}$ contain the nonempty response rows. Prompt, padding, and other masked tokens contribute neither loss nor token count. The loss mask covers generated response tokens, including final-answer tokens, whereas the shaping penalty uses reasoning tokens only.

For a row from trajectory k of prompt q, let $\mathcal{R}(q, k)$ be the rows belonging to that trajectory. The group-centered advantage in Equation 5 is distributed across rows as

$$n_{q,k} = \sum_{i \in \mathcal{R}(q,k)} n_i, \qquad \widetilde{a}_i = \frac{n_i}{n_{q,k} + \varepsilon_{\text{num}}} a^{(k)}, \qquad \varepsilon_{\text{num}} = 10^{-15}. \tag{12}$$

The same $\widetilde{a}_i$ is used at every unmasked token in row i. This correction combines row-wise token averaging with trajectory-level length normalization when a trajectory spans multiple rows.

Write π_θ for the actor being optimized and π_{roll} for the policy that generated the response. Rollout token log probabilities are retained, and pre-update actor log probabilities are not recomputed. The importance ratio and sampled-token probability difference are

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid h_{i,t})}{\pi_{\text{roll}}(y_{i,t} \mid h_{i,t})}, \qquad \overline{\rho}_{i,t} = \text{sg}[\text{clip}(\rho_{i,t}, 0, 5)],$$
$$\Delta_{i,t}(\theta) = \pi_\theta(y_{i,t} \mid h_{i,t}) - \pi_{\text{roll}}(y_{i,t} \mid h_{i,t}), \tag{13}$$

where sg stops gradients through the importance weight. The probability-difference gate is

$$g_{i,t}(\theta) = 1 - \mathbf{1}[(\widetilde{a}_i > 0 \ \wedge \ \Delta_{i,t} > 0.3) \ \vee \ (\widetilde{a}_i < 0 \ \wedge \ \Delta_{i,t} < -0.3)]. \tag{14}$$

Thus, clipping acts on the sampled token's probability difference, rather than on a PPO probability-ratio surrogate. The actor minimizes

$$u_{i,t}(\theta) = -\widetilde{a}_i \, \overline{\rho}_{i,t} \, g_{i,t}(\theta) \log \pi_\theta(y_{i,t} \mid h_{i,t}),$$
$$\mathcal{L}_{\text{actor}}(\theta) = \frac{1}{|\mathcal{R}|} \sum_{i \in \mathcal{R}} \frac{1}{n_i} \sum_t m_{i,t} \min(u_{i,t}(\theta), 100). \tag{15}$$

The upper bound of 100 is applied to each token loss before aggregation. Advantages, importance weights, and the discrete gate are held fixed for differentiation. There is no additional out-of-range importance-ratio gradient mask, entropy bonus, or reference-policy KL term. Within each row, gated-off tokens remain in the denominator n_i. Nonempty rows receive equal weight after their

Table 5: Shared adaptive-training settings and the selected main-comparison parameters.

Setting	Value
Model	Qwen3.5-122B-A10B
Candidate / retained responses per prompt	24 / 16
Prompts per update / minibatch	128 / 128
Learning rate	2×10^{-6}
Prompt / response limit	16,384 / 131,072 tokens
Training temperature / top-p	1 / 1
Evaluation temperature / top-p / top-k	1 / 0.95 / 20
Selected α / β	0.80 / 0.30
Penalty cap / length normalization	0.5 / 131,072
Within-group standard-deviation division	Disabled
Batch advantage normalization	Disabled
Positive / negative clipping range	0.3 / 0.3
Token importance-weight range	$[0, 5]$ (stop-gradient)
Per-token loss upper bound	100
Optimization epochs per update	1
Loss aggregation	Token mean, then response-row mean
KL coefficient	0
Entropy coefficient	0
All-correct / all-incorrect group retention	1 / 0

token means are computed; the objective is not a global mean over all tokens. Dynamic micro-batches are scaled by their response-row counts before gradient accumulation. Each update uses one optimization epoch over a minibatch corresponding to 128 prompts.

The adaptive configurations retain all-correct groups and discard all-incorrect groups, while the unpenalized reference disables the group filter. An additional standalone length-reward module is disabled, avoiding an intentionally configured second length cost. The comparison therefore concerns the full recorded training recipes, including their filtering choices, rather than an isolated reward-function intervention.

The GFPO baseline samples 32 responses and retains 16 using length-based stratified selection, with guess exclusion and a target-length setting of 65,536 tokens. Its evaluation history spans checkpoints 10–120. Training uses sample-averaged loss.

The fixed DLER configuration uses a 65,536-token training response limit, group filtering, and batch advantage normalization with standard-deviation division. The no-batch-normalization variant disables the corresponding normalization stage. Both use a fixed reward-zeroing threshold of $B_{\mathrm{DLER}} = 81{,}919$ tokens. The threshold is compared with the trajectory's generated-response token count, obtained by summing its response mask; it includes final-answer tokens but excludes prompt tokens and masked positions. It is not compared with prompt-plus-response context length or reasoning-only length.

At training-group completion, before group filtering, the rule is

$$r_{\mathrm{DLER}}^{(k)} = \begin{cases} 0, & L_{\mathrm{response}}^{(k)} > B_{\mathrm{DLER}}, \\ r_{\mathrm{task}}^{(k)}, & L_{\mathrm{response}}^{(k)} \le B_{\mathrm{DLER}}. \end{cases} \tag{16}$$

The comparison is strict: a response at the threshold keeps its reward. An over-budget response has its training reward set to zero and is marked as over budget, while its original correctness label remains unchanged. This operation neither stops decoding nor deletes generated tokens; generation has already returned. The modified rewards enter group filtering and subsequent advantage computation. The rule is disabled for evaluation, where all methods use the same 131,072-token response limit.

Under the configured training response limit, $L_{\mathrm{response}}^{(k)} \le 65{,}536 < 81{,}919$, so this reward-zeroing branch cannot trigger. The 81,920-token context limit does not change this conclusion because prompt tokens are excluded from the comparison. Thus, the reported DLER runs compare local

training configurations with a shorter generation limit, group filtering, and their respective normalization choices, not an active DLER truncation penalty. Their results do not establish the effect of reward zeroing at a reachable threshold.

Table 6: DLER normalization variants using the five-benchmark means. Accuracy and response length are computed from unrounded evaluation statistics at checkpoint 100.

Variant	Mean accuracy (%)	Mean response tokens
DLER	**84.19**	29,248
DLER w/o batch norm.	81.80	**25,781**

C Training Data Composition and Sampling

The training corpus is a merged JSONL collection of 6,930 prompt records. Its mathematics-category tasks contain 3,761 standard mathematical-reasoning records, 66 out-of-distribution IQ records, and 2 additional IQ records. Competitive programming contributes 3,101 records. Table 7 reports the resulting task proportions and source-label counts.

Table 7: Composition of the training corpus. Task proportions describe records in the merged corpus, not fixed per-batch quotas. Source labels identify constituent collections.

Task category	Records	Share (%)
Mathematics, including IQ reasoning	3,829	55.25
Competitive programming	3,101	44.75
Total	6,930	100.00

Source label	Records
`open_math_reasoning`	2,644
`xcoder_rl`	1,317
`hardtests_problems`	895
`skywork`	320
Other named sources	609
Unspecified source	1,145
Total	6,930

The other named sources comprise `aopeng` (144), `klear_codetest` (79), `rstar_coder_seed_rl` (77), `am_thinking_v1_rl_dataset` (71), `nvidia_AceReason-1.1-SFT` (61), `nemotron_rl_coding` (35), `deepmath` (34), `areal` (28), `synthetic_2_rl` (21), `open-r1_DAPO-Math-17k-Processed` (19), `aops_instruct` (17), `Kwai-Klear_KlearReasoner-MathSub-30K` (12), `klear_reasoner` (8), `POLARIS-Project_Polaris-Dataset-53K` (2), and `nvidia_AceReason_Math` (1).

The loader reads the `messages` field from this single merged data source. Training uses shuffled sampling without replacement within each pass, with batches of 128 prompts and no adaptive source reweighting. Consequently, the mathematics–programming balance is determined by corpus composition rather than separate task quotas. The prompt limit is 16,384 tokens with right truncation. Rollout group filtering is applied after generation, separately from corpus sampling.

Exact-duplicate checks compare the full `prompt` and `messages` fields separately. Each field is serialized with sorted object keys, preserved Unicode text, and compact JSON separators, then identified by a SHA-256 hash; array order and text content are preserved. Both checks yield 6,930 distinct records and zero exact duplicates. The 5,785 nonmissing question fields and 5,785 nonmissing UUIDs are also individually unique. These corpus-level checks are distinct from the query–trajectory and row-identifier deduplication used during reward computation.

D Validation Metrics and Parameter Selection

We compute benchmark results from checkpoint-level evaluations using original response correctness and total response lengths. AIME 2026 pools the general and part-I subsets. The following aggregation rules apply to every evaluated method.

For benchmark b with n_b evaluated responses, accuracy is $a_b = 100\, n_b^{-1} \sum_{i=1}^{n_b} c_{b,i}$, where $c_{b,i} \in \{0, 1\}$ is original correctness. Each response has equal weight within its benchmark, including repeated entries; main results do not average across checkpoints. For each constituent task label s, let n_s be its response count and t_s its mean response length. We compute

$$t_b = \frac{\sum_{s \in b} n_s t_s}{\sum_{s \in b} n_s}, \qquad \overline{a} = \frac{1}{5} \sum_{b=1}^{5} a_b, \qquad \overline{t} = \frac{1}{5} \sum_{b=1}^{5} t_b. \tag{17}$$

The first expression matches within-benchmark response lengths to the accuracy aggregation. The last two give benchmarks equal weight regardless of their numbers of repeated entries. Token savings are $100(1 - \overline{t}/\overline{t}_{\text{reference}})$ percent. All means, differences, and savings use unrounded values. This avoids letting the largest evaluation file dominate the overall average. The AIME 2025 labels include both text/graph variants and exam parts, and the local coding subset is identified as LiveCodeBench-v6 Python.

Table 8: Evaluation file weighting. Repeats are retained within each benchmark but do not increase that benchmark's weight in the five-benchmark average.

Family	Entries	Reporting location
GPQA	1,980	Main table
AIME 2025	480	Main table
HMMT February 2026	264	Main table
AIME 2026	240	Main table
LiveCodeBench-v6 Python	1,400	Main table
HMMT February 2025	240	Additional benchmark table
HMMT November 2025	240	Additional benchmark table
All configured entries	4,844	Includes repeated mathematics file

Parameter selection averages the five-benchmark accuracy at checkpoints 100 and 110, common to all three adaptive configurations with evaluation results and to the selected reference evaluation stream. The maximum determines one parameter configuration, $(0.80, 0.30)$, used throughout the main baseline comparison and difficulty analysis. Checkpoint 100 is then used for every row of the main tables. This separates parameter selection from per-benchmark or per-run checkpoint selection, but the selection and reporting sets still overlap; it is an exploratory validation comparison rather than an independently selected test result.

Training summaries give equal weight to each of the 30 steps from 81 to 110. Parameter-specific training curves use trailing five-step means, while tables use unsmoothed data. Resumed runs are treated as continuations of the same training run. The results come from single training runs per configuration, so small differences do not establish statistically significant rankings.

E Validation Histories

Figure 4 shows the accuracy and response length histories of the three length-controlled methods. The figure omits the unpenalized reference because its evaluation history is incomplete; the main tables report its checkpoint-100 results. All main comparisons use checkpoint 100; parameter selection uses checkpoints 100 and 110 as described in Appendix D.

F Additional Benchmarks

DASE is less accurate than GFPO and DLER on both HMMT 2025 benchmarks. These two benchmarks are excluded from parameter selection and the five-benchmark averages.

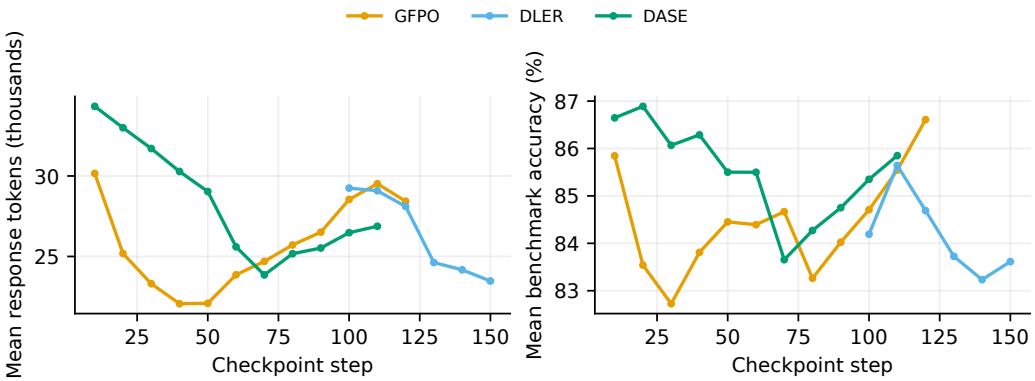

Figure 4: Validation histories for GFPO, DLER, and the selected DASE configuration. Accuracy and response length are five-benchmark means. Lines join available checkpoints; no missing evaluation is imputed.

Table 9: Additional HMMT benchmark accuracies at checkpoint 100 (%). These two benchmarks are not part of the five-benchmark parameter-selection average.

Method	HMMT February 2025	HMMT November 2025
Unpenalized	**87.50**	87.50
GFPO	82.50	88.75
DLER	83.75	**90.42**
DASE	77.50	85.42

G Additional Parameter-Ablation Results

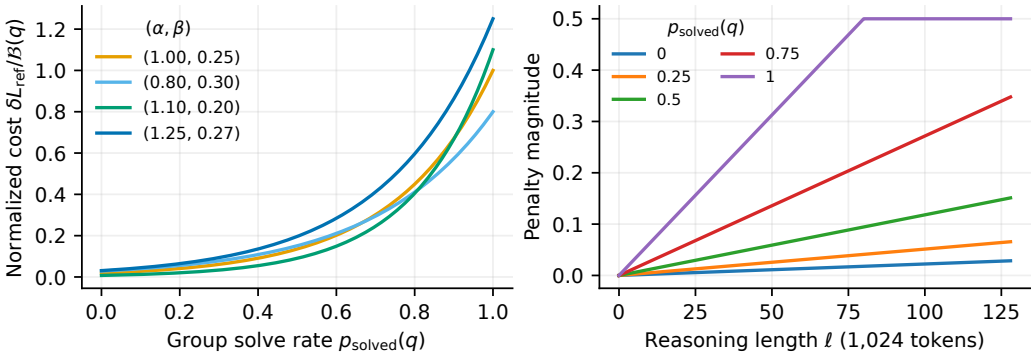

Figure 5: Analytical parameter behavior, not experimental outcomes. Left: normalized per-token cost $L_{\text{ref}} w(q) = \delta L_{\text{ref}}/\mathcal{B}(q)$ for the four (α, β) settings. Right: deductions under the selected $(0.80, 0.30)$ setting for different observed success rates.

The training analysis covers all four parameter settings. Their cost curves depend jointly on α and β, so the scale parameter alone does not determine the resulting response length. The $(1.25, 0.27)$ setting has no usable benchmark evaluation and appears only in the analytical curves and training summaries.

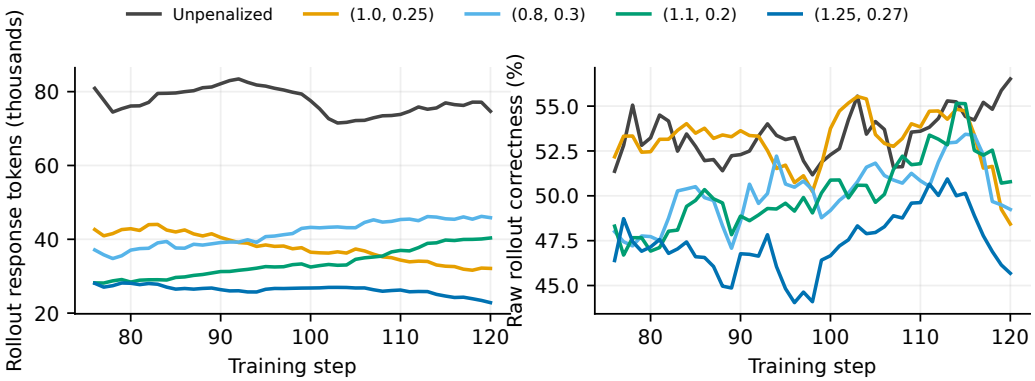

Figure 6: Parameter-ablation training curves over steps 76–120. Legend pairs denote (α, β); the unpenalized reference supplies context. Curves use a trailing five-step mean. Correctness is original rollout correctness, not shaped reward or validation accuracy.

Table 10: Unsmoothed means of per-step training metrics over steps 81–110. Savings are relative to the unpenalized reference in the same interval.

Configuration	Correctness (%)	Response tokens	Saving (%)
Unpenalized	53.04	77,805	0.0
DASE (1.00, 0.25)	**53.31**	38,194	50.9
DASE (0.80, 0.30)	50.26	41,533	46.6
DASE (1.10, 0.20)	50.08	32,926	57.7
DASE (1.25, 0.27)	47.07	**26,525**	**65.9**

H Difficulty-Conditioned Trajectory Analysis

The distribution figure uses saved training batches at steps 90, 100, and 110 from the selected $(0.80, 0.30)$ configuration. Each batch contains 128 query groups of 16 distinct trajectories. We read the non-tensor metadata containing original correctness and generated token IDs, skipping the large serialized tensor payload. The extractor restricts the deserialization types, checks that each token-ID list length equals its recorded response-token count, deduplicates query–trajectory pairs, and verifies a group size of 16. No raw prompt, response text, or original query identifier is exported; a hash identifies each query group in the analysis table.

The tokenizer's end-of-thinking marker has ID 248069. Reasoning length is the number of generated tokens before its last occurrence. If the marker is absent, all generated tokens count as reasoning; seven of the 6,144 trajectories follow this rule. Group difficulty is computed from original binary correctness before applying the length penalty. The bins are easy for $p_{\text{solved}}(q) \geq 0.75$, medium for $0.25 \leq p_{\text{solved}}(q) < 0.75$, and hard for $p_{\text{solved}}(q) < 0.25$. All-incorrect groups are absent from these saved filtered training batches, so the observed hard groups have one to three correct responses out of 16.

Table 11: Difficulty-conditioned reasoning lengths for the selected DASE configuration. Counts refer to query-group occurrences and trajectories across three training steps.

Difficulty	Groups	Trajectories	Mean tokens	Median tokens
Easy	128	2,048	37,836	35,976
Medium	178	2,848	43,728	41,316
Hard	78	1,248	44,432	42,066

The violin plots and empirical cumulative distributions use all trajectories in each difficulty bin. Boxplot whiskers follow the 1.5-interquartile-range rule; the density plots and cumulative curves

retain the full observed range. Counts refer to query-group occurrences because the same training query can appear at different steps. Solve rates and lengths are measured from the same groups.

I OAA Protocol and Additional Results

We follow the response-token OAA definition and benchmark-specific budget convention in Section 3 of Rakotonirina et al. (2026). OAA counts a response only if its final answer is correct and its generated length is strictly less than the threshold. The denominator is the full number of sampled responses for that benchmark. Consequently, shortening an incorrect response yields no direct OAA credit, and a correct response whose length exceeds the integration budget contributes zero area. Increasing the budget changes the metric; as the budget tends to infinity, the normalized area tends to ordinary accuracy. Reporting the common budgets is therefore necessary for interpreting the scores.

I.1 Calibration and trajectory matching

For each benchmark, we set the integration budget to the mean length in the saved step-0 evaluation from the common initialization. This calibration model is distinct from the unpenalized checkpoint-100 reference in the main tables. All methods then use exactly the same benchmark-specific budgets. The step-0 response-token AUC itself is 21.53, compared with 16.77 for the unpenalized reference in Table 2. Parameters are not reselected for OAA.

Table 12: Fixed OAA integration budgets derived from the initial model's step-0 evaluation. Unrounded means are used for computation. Response and reasoning variants each calibrate their own length unit.

Benchmark	Responses	Response budget	Reasoning budget
GPQA	1,980	27,054.24	26,375.29
AIME25	480	34,573.97	33,714.50
HMMT26	264	44,465.88	43,615.85
AIME26	240	35,765.67	34,895.87
LCB	1,400	38,281.49	37,444.26

The five main benchmarks contain 4,364 evaluated responses per model, retaining repeated entries and within-benchmark task weighting. We extract original binary correctness, recorded response-token counts, and generated token IDs from each saved evaluation batch. The multiset of original prompts and task labels is checked across models, including the step-0 calibration batch. For every checkpoint-100 model, the extracted per-benchmark correctness and mean response length match the corresponding aggregate evaluation statistics. No OAA result is estimated from aggregate accuracy and mean length alone.

Equation 7 integrates the empirical curve exactly. For an individual correct response, integrating $\mathbb{I}[L < t]$ from zero to T gives $\max(T - L, 0)$; an incorrect response contributes zero. This proves the closed-form expression after division by nT. The strict boundary has zero measure in this continuous integral. Our verification independently sums the constant curve heights between sorted observed length breakpoints. The plotted grid is only for visualization and does not determine the reported AUC.

To average curves across benchmarks with different budgets, we use normalized threshold $u = t/T_b$ and define

$$\overline{\mathrm{OAA}}(u) = \frac{1}{5}\sum_{b=1}^{5} \mathrm{OAA}_b(uT_b), \qquad \int_0^1 \overline{\mathrm{OAA}}(u)\,du = \frac{1}{5}\sum_{b=1}^{5} \mathrm{AUC}_{\mathrm{OAA},b}. \tag{18}$$

Thus the area of the macro-average normalized curve equals the mean of the benchmark AUC values. The endpoint at $u = 1$ need not reach ordinary accuracy, because some correct responses exceed the initial model's mean length. The curves do not represent new inference runs under forced token budgets, and they do not count an intermediate correct answer before the model's final response completes.

I.2 REASONING-TOKEN VARIANT

Primary OAA uses total response tokens, following the cited paper. To directly examine reasoning length, we additionally replace L with the number of generated tokens before the final end-of-thinking marker and calibrate the corresponding budget from the initial model's mean reasoning length. Missing end-of-thinking markers are treated as unfinished reasoning, so all generated tokens count. This is a separately labeled variant with different length units, not a replacement of response-token OAA.

Table 13: Reasoning-token $\mathrm{AUC}_{\mathrm{OAA}}$ at checkpoint 100, multiplied by 100. Integration budgets are the common initial-model reasoning lengths in Table 12; the main Table 2 uses response lengths instead.

Method	GPQA	AIME25	HMMT26	AIME26	LCB	Avg.
Unpenalized	22.49	14.31	15.96	14.64	17.94	17.07
GFPO	**39.95**	25.52	25.79	25.92	27.15	28.86
DLER	36.40	27.75	28.61	29.51	22.99	29.05
DASE	39.26	**34.11**	**31.81**	**35.50**	**32.60**	**34.66**

The main comparison uses responses whose serialized token-ID counts agree with the recorded response lengths, supporting the reasoning-token calculation. In the additional $(1.10, 0.20)$ ablation run, two coding responses have one fewer serialized token ID than the recorded response count. We retain their recorded response lengths and correctness for primary OAA, which still matches the validation statistics, but do not impute a reasoning length or report a partial-benchmark reasoning macro-average for that configuration.

I.3 LENGTH–CORRECTNESS ASSOCIATION

For each benchmark, we compute the Pearson correlation between total response length and binary correctness, which is equivalent to the point-biserial correlation. Negative values indicate longer incorrect responses. We report the correlations separately by benchmark without significance tests, since repeated responses to the same questions are not independent.

Table 14: Within-benchmark point-biserial correlation between total response length and binary correctness at checkpoint 100. These are descriptive associations, not causal estimates.

Method	GPQA	AIME25	HMMT26	AIME26	LCB
Unpenalized	-0.351	-0.680	-0.636	-0.689	-0.607
GFPO	-0.376	-0.523	-0.471	-0.520	-0.525
DLER	-0.377	-0.648	-0.599	-0.706	-0.501
DASE	-0.373	-0.670	-0.590	-0.697	-0.601

Table 3 in the main text reports the complementary response lengths conditional on correctness, computed within benchmarks and then averaged equally across the five benchmarks.

I.4 OAA UNDER PARAMETER ABLATION

Table 15: Additional OAA parameter ablation. Accuracy and response-token AUC are five-benchmark means in percent-scale units. The selected $(0.80, 0.30)$ row repeats the main results. The originally selected setting is not reselected using AUC. Bold marks the best value in each metric column.

α	β	Mean accuracy	Mean $\mathrm{AUC}_{\mathrm{OAA}}$
1.00	0.25	85.11	36.90
0.80	0.30	**85.35**	33.94
1.10	0.20	83.46	**37.42**

Both alternative parameter settings achieve higher OAA than the selected configuration, while their checkpoint-100 accuracies are lower. The selected configuration follows the validation-accuracy criterion in Appendix D.