

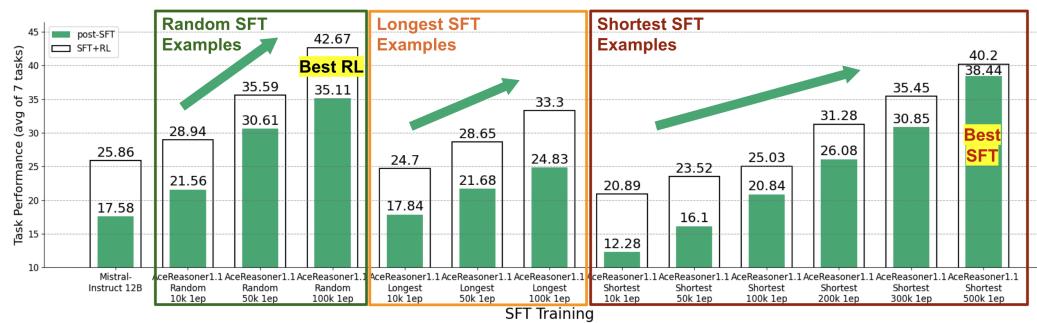
000 001 QUAGMIRES IN SFT-RL POST-TRAINING: WHEN HIGH 002 SFT SCORES MISLEAD AND WHAT TO USE INSTEAD 003 004

005 **Anonymous authors**
006 Paper under double-blind review

007 008 ABSTRACT 009

011 In post-training for reasoning Large Language Models (LLMs), the current state of
012 practice trains LLMs in two independent stages: Supervised Fine-Tuning (SFT)
013 and Reinforcement Learning with Verifiable Rewards (RLVR, shortened as “RL”
014 below). In this work, we challenge whether high SFT scores translate to improved
015 performance after RL. We provide extensive counter-examples where this is not
016 true. We find high SFT scores can be biased toward simpler or more homoge-
017 neous data and are not reliably predictive of subsequent RL gains or scaled-up
018 post-training effectiveness. In some cases, RL training on models with improved
019 SFT performance could lead to substantially worse outcome compared to RL on
020 the base model without SFT. We study alternative metrics and identify general-
021 ization loss on held-out reasoning examples and Pass@large k performance to
022 provide strong proxies for the RL outcome. We trained hundreds of models up
023 to 12B-parameter with SFT and RLVR via GRPO and ran extensive evaluations
024 on 7 math benchmarks with up to 256 repetitions, spending >1M GPU hours.
025 Experiments include models from Llama3, Mistral-Nemo, Qwen3 and multiple
026 state-of-the-art SFT/RL datasets. Compared to directly predicting from pre-RL
027 performance, prediction based on generalization loss and Pass@large k achieves
028 substantial higher precision, improving R^2 coefficient and Spearman’s rank corre-
029 lation coefficient by up to 0.5 (2x). This provides strong utility for broad use cases.
030 For example, in most experiments, we find SFT training on unique examples for
031 a one epoch underperforms training on half examples for two epochs, either after
032 SFT or SFT-then-RL; With the same SFT budget, training only on short examples
033 may lead to better SFT performance, though, it often leads to worse outcome after
034 RL compared to training on examples with varying lengths. This work develops an
035 enhanced evaluation tool that will be open-sourced.

036 1 INTRODUCTION 037



049 Figure 1: Mistral-NeMo-12B-Instruct undergone SFT-RL with SFT examples from AceReasoner1.1-SFT
050 dataset and RLVR via GRPO on DeepScaleR dataset. Reporting Pass@1 performance averaged over 7 math
051 benchmarks. When training on **Random/Longest/Shortest** SFT examples, *the final performance after RL*
052 *increases at different rates than the SFT performance*. Model with the best SFT performance is not the one with
053 the best final performance after RL. Post-SFT and SFT+RL performance correlate, though, optimizing post-SFT
054 performance might not optimize the final performance after RL.

The evolution of Large Language Models (LLMs) has seen a significant focus on enhancing their reasoning abilities, a process heavily reliant on post-training (Wen et al., 2025). This phase refines pre-trained models, adapting them for complex, multi-step tasks like mathematics, logic, and code generation, leading to the emergence of Large Reasoning Models (LRMs) (Kumar et al., 2025). The open-sourced DeepSeek R1 achieved phenomenal success in pushing forward the frontier of LLM’s reasoning capabilities (Guo et al., 2025). Its new post-training paradigm, Reinforcement Learning with Verifiable Rewards (RLVR) via Group Relative Policy Optimization (GRPO) (Liu et al., 2024; Shao et al., 2024), has shown substantial improvements on top of previous post-training methods. Following DeepSeek R1’s practice, current works typically conduct SFT before RL, assuming models with better performance after SFT will ultimately be better after RL (Liu et al., 2025b; Wen et al., 2025). In industrial practice, these post-training stages are often distributed among different teams, with SFT and RL handled by separate groups, each optimizing for their own performance metrics (Chen et al., 2025b; Meta, 2025). This process relies on the intuition that a model with stronger SFT performance will yield better outcomes after RLVR (Liu et al., 2025b). With efforts and resources being poured in improving post-training paradigms and data recipes, also escalating are the debates on whether SFT helps or hurts the subsequent RL training.

In this setup, post-training strategies and data are often designed either for SFT or RL, but not jointly. In practice, SFT and RL are often conducted sequentially (e.g., Rastogi et al. (2025)). SFT data is usually selected to maximize evaluation performance after SFT (Zhang et al., 2025; Ye et al., 2025), and the best-performing SFT models are believed to also yield stronger performance after subsequent RL. However, this assumption is often flawed. Over-training during SFT, for instance, can constrain the model’s behavior and limit the exploration crucial for effective RL (Chen et al., 2025a; Wang et al., 2025). For example, we find training on repeated examples for up to 8 epochs leads to better SFT performance than training on the same data for 2 epochs (4x compute) but yields visibly worse outcome after RL (Figure 4, left). On the contrary, Cen et al. (2025) shows SFT training on manually crafted “exploratory” examples, despite leading to a lower performance after SFT, helps achieve better final outcome after RL. This leads to a critical gap in the current practice:

An SFT-trained model with the best evaluation performance may not be the best candidate for subsequent training with RLVR (e.g., Figure 1).

When the final RLVR performance is unsatisfactory, it becomes challenging to attribute the failure to either the RL stage or a non-ideal SFT starting point. This misalignment can cause friction and overhead between teams. Furthermore, the high computational cost of RL training and long pipelines, especially in agentic use cases, makes end-to-end tuning across the SFT-RL stages prohibitively expensive (Toledo et al., 2025). Early stopping during RL is also generally ineffective, as the model with the fastest initial improvement may not achieve the highest final performance (Liu et al., 2025b). Even with identical post-training procedures, different models may respond vastly different (Figure 2). Consequently, a significant gap remains in our ability to reliably predict RLVR outcomes.

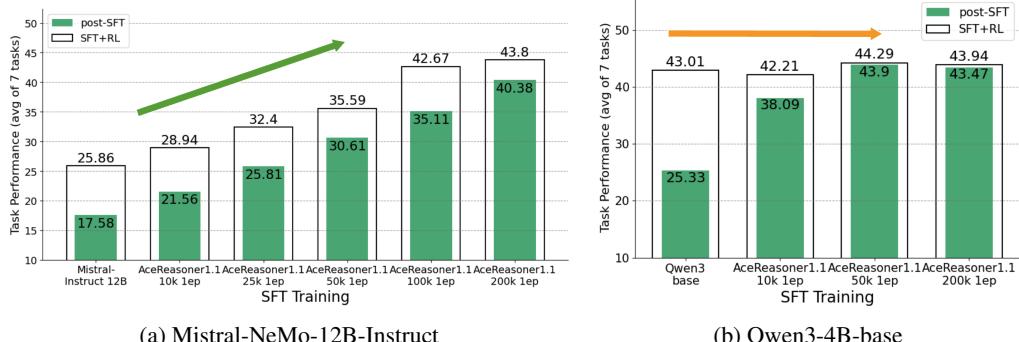


Figure 2: Both models underwent SFT-RL with SFT examples from AceReasoner1.1-SFT dataset and RLVR via GRPO on DeepScaleR dataset. Reporting Pass@1 performance averaged over 7 math benchmarks. *Even with identical post-training procedures, different models may respond vastly different.* With increasing SFT examples, Mistral’s (left) post-SFT performance and final performance after RL both increase. Yet, for Qwen3 models (right), the post-SFT performances appear uncorrelated with the final performance after RL, where the latter remains the same despite the substantially improved SFT performance.

This work centers on addressing this predictability problem. We ask the following research questions:

108 RQ1: *Do models with better pre-RL performance always lead to better outcomes after RLVR? If
109 not, what are the failure modes? (Section 3)*

110 RQ2: *What are effective SFT paradigms and data recipes when considering subsequent RLVR
111 training? Can we determine the suitability of an SFT model before committing to the
112 expensive RL stage? (Section 4)*

113 To tackle these questions, we first examine the relationship between pre-RL performance and post-RL
114 outcomes across various SFT training paradigms and data recipes. While we often observe some
115 extents of correlation between post-SFT performance and final outcome after RL, we identify cases
116 where the trends visibly diverge. For instance, training on the same dataset for more epochs may
117 significantly boost post-SFT performance but diminishes the potential for improvement during RL,
118 sometimes resulting in lower final performance (e.g., Figure 4). Similarly, training on simpler, shorter
119 reasoning examples improves pre-RL performance quickly, but these models gain much less from the
120 subsequent RL stage. These findings are particularly concerning given that many SFT data selection
121 methods favor simpler or more homogeneous examples (Zhang et al., 2025; Yu et al., 2025).

122 Next, we identify more reliable predictors for RL success. We find that as SFT proceeds, an eventual
123 increase in validation loss is strongly correlated with performance improvements in the later RL
124 stage. Furthermore, since the RL objective is to compress Pass@k performance into Pass@1 (i.e.,
125 maximize expected reward), we investigate using Pass@k at a large k as a predictor. We conduct
126 extensive empirical validation using Llama3-8B (Grattafiori et al., 2024), Mistral-Nemo-12B (team,
127 2024), and Qwen3-4B-base (Yang et al., 2025) models on state-of-the-art SFT datasets like Llama-
128 Nemotron (Singhal et al.) and AceReasoner1.1 (Liu et al., 2025b) and different RL datasets. Our
129 results demonstrate that these new metrics can reliably predict the outcome of RLVR, improving R^2
130 coefficient and Spearman’s rank correlation coefficient by up to 0.5 (2x), providing strong utility for
131 broad use cases. For example, in most experiments, we find SFT training on unique examples for a
132 one epoch underperforms training on half examples for two epochs, either after SFT or SFT-then-RL;
133 With the same SFT budget, training only on short examples may lead to better SFT performance,
134 though, it often leads to worse outcome after RL compared to training on examples with varying
135 lengths. These can be captured by the proposed predictors, but not from the post-SFT performance.

136 To address the limitations in available tools, we developed an enhanced tool for more convenient and
137 reliable evaluation of reasoning models, which will be open-sourced in contribution to the community.

138 2 RELATED WORKS

140 The research landscape for reasoning post-training and data strategies is fast evolving and in its
141 early days. In a typical setup, post-training for reasoning LMs conducts SFT and RL sequentially,
142 which has been reported to work better than only conducting SFT or RL (Rastogi et al., 2025).
143 Viewpoints in many impactful works can be inconsistent or even contradicting: “Initial ‘cold-start’
144 SFT is necessary for subsequent RL” (DeepSeek-R1 technical report, Guo et al. (2025)); “over-SFT
145 may constrain subsequent RL” (Llama-4 technical report, Meta (2025)); “SFT generalizes poorly
146 and RL without SFT does better” (Chen et al., 2025a), showing prominent gaps in characterizing
147 post-training dynamics and the role of each stage. The lack of predictability in the post-training
148 outcome poses a major blocker for optimizing training paradigms or data recipes.

149 2.1 POST-TRAINING FOR REASONING: SFT-THEN-RLVR PARADIGM

151 Post-training for reasoning LLMs typically consists of two or three stages: a) **Supervised Fine-**
152 **Tuning (SFT)**, b) an optional **Direct Preference Optimization (DPO)** stage, and c) **Reinforcement**
153 **Learning with Verifiable Rewards (RLVR)** (Lambert et al., 2024). SFT serves as the “cold-start”
154 phase, providing the model with a strong initial policy by exposing it to high-quality reasoning chains
155 (Guo et al., 2025). The model is trained on problems with high-quality solutions sourced from the
156 frontier models. DPO fixes/strengthens targeted behaviors (e.g., precise instruction following in
157 math/logic derivations) assessed important for effective reasoning, which is more subjective and
158 often optional (Lambert et al., 2024). RL further improves the model’s reasoning and problem-
159 solving capability. This allows the model to explore the solution space more broadly than SFT alone,
160 discovering novel and more robust reasoning paths.

161 While the sequential SFT-then-RL pipeline is dominant, researchers have explored alternative
paradigms to more tightly integrate or unify these learning stages. Efforts include iterate or in-

162 terleave SFT and RL (Meta, 2025), gradually shift from SFT to RL while increasing task difficulty
 163 (Yang et al., 2025), or directly unify the objectives of SFT and RL (Xu et al., 2025). Though these are
 164 promising research directions, they come with their own complexities and have not yet universally
 165 replaced the SFT-then-RL paradigm, which remains a robust and widely-adopted industry standard.
 166 Many important issues regarding the stability, data requirements, and effectiveness of these unified
 167 methods remain to be solved. Our work, therefore, focuses on improving the predictability and
 168 efficiency of the prevailing SFT-then-RL pipeline, providing practical tools that are immediately
 169 applicable to current state-of-the-art workflows.

170 2.2 RECENT ADVANCEMENTS AND CURRENT CHALLENGES

171 In post-training for reasoning, SFT data is usually selected to maximize evaluation performance
 172 after SFT (Li et al., 2025; Ye et al., 2025), and the best-performing SFT models are believed to also
 173 yield stronger performance after subsequent RL. Significant research effort is now focused on more
 174 sophisticated selection and curation strategies for SFT data. Techniques range from filtering for
 175 complexity and diversity to generating synthetic data that covers a wider range of reasoning structures
 176 (Rastogi et al., 2025; Yuan et al., 2025; Ye et al., 2025; Abdin et al., 2024). Some methods propose
 177 selecting data points based on their difficulty and influence, aiming to find a subset of examples that
 178 provides the strongest learning signal (Muennighoff et al., 2025). Current efforts prioritize scaling up
 179 SFT training on existing models, leading to new SOTA performance on reasoning tasks for those
 180 models (Guha et al., 2025). A significant challenge is that standard SFT performance metrics, such
 181 as average accuracy on benchmarks, are not always predictive of post-RL success. This creates a
 182 critical gap between the optimization target of the SFT stage and the final performance of the model.
 183

184 Several issues contribute to this gap. First, models can overfit to the specific patterns and artifacts
 185 present in the SFT dataset, leading to poor generalization during the exploration phase of RL (Chen
 186 et al., 2025a). Furthermore, naively collecting or generating data can lead to datasets that lack
 187 diversity in reasoning strategies or are skewed toward simpler problems, causing the SFT-trained
 188 model to develop biases that stifle exploration in the subsequent RL stage (Guha et al., 2025). The
 189 landscape is further fogged by the recently reported data contamination issues (Wu et al., 2025). The
 190 results from these models have served as the basis for many research findings.

191 The (lack of) predictability for final performance after RL from pre-RL models leads to quagmires
 192 for post-training. SFT teams may provide suboptimal RL learners. It creates frictions between post-
 193 training teams owning different SFT and RL stages and chaos in optimizing the training paradigm/data
 194 recipes, adding overheads on the model development and hindering productivity. It calls for new
 195 tools that better characterize the post-training dynamics and predictive of the RL outcome. This will
 196 have profound impact on broad downstream fields—research and applications alike—from improving
 197 SFT data curation, search for the next post-training paradigm, to RL for non-verifiable tasks, etc.

198 3 THE SFT METRIC TRAP

200 Previous works, from SFT data selection to RL training methodologies, have often operated under
 201 a common assumption. They *implicitly* assume or *explicitly* argue that models exhibiting better
 202 post-SFT performance will consistently yield superior final outcomes after subsequent reinforcement
 203 learning (Rastogi et al., 2025; Liu et al., 2025b). This assumption has justified the widespread practice
 204 of optimizing the SFT and RL stages in isolation, with teams or processes focusing on maximizing
 205 SFT evaluation metrics as a primary goal. However, the separation of SFT and RL optimization can
 206 lead to a widening gap in reasoning post-training, where improvements in the initial stage do not
 207 translate to the final stage. This motivates us to ask two fundamental questions:

- 208 • *Do models with better pre-RL performance always lead to better outcomes after RLVR?*
- 209 • *If not, what are the failure modes?*

211 To investigate these questions, we design experiments across two representative scenarios that reflect
 212 common practices and research directions in the field: a “dataset-level” analysis and an “instance-
 213 level” analysis. In **Dataset-Level Scenarios**, SFT examples are drawn from the same data distribution,
 214 but we vary the amount of unique samples and the training paradigm (e.g., learning rate, number of
 215 epochs); In **Instance-Level Scenarios**, we consider training on different datasets while keeping the
 training pipeline fixed (i.e., using the same model and training paradigm). This setup is primarily

216 concerned with SFT data selection and curation, examining whether strong SFT performance on a
 217 given dataset transfers to the final outcome after RL.
 218

219 3.1 DATASET-LEVEL SCENARIOS 220

221 In this scenario, we draw SFT examples from the same underlying data distribution but vary the
 222 training configuration, such as the number of unique samples/training epochs/learning rate. This
 223 setup is highly relevant to industrial practices where SFT and RL are often handled by different teams.
 224 In current practices, the number of training epochs is a design choice often determined by practical
 225 factors such as data availability or compute budget. Specifically, when the amount of training samples
 226 is a more prominent constraint (such as domains with limited high-quality examples), repeating for
 227 more epochs on the data may be preferred to improve post-SFT performance. On the contrary, if data
 228 is abundant relative to the allocated compute budget (for this domain/capability), current practices
 229 (such as [Singhal et al.](#)) may prefer to train for just a single epoch on unique examples.
 230

231 In these cases, the training paradigm is determined
 232 **heuristically** where the only optimizable target
 233 is the post-SFT performance. Surprisingly, we
 234 identified both practices to be **suboptimal**. We
 235 found that post-SFT performance often improves
 236 stably when training for more epochs—even with
 237 excessive overtraining. But models overtrained
 238 during SFT show decreasing potentials for the
 239 subsequent RL. Typically, the model with the best
 240 final performance after RL is not the one with
 241 the best post-SFT performance. Further, with the
 242 same compute budget for SFT, training on more
 243 data for one epoch typically leads to visibly lower
 244 post-SFT performance compared to training on
 245 less data for a few more epochs, and the final
 246 performance after RL remains underperforming.
 247 A concrete example is provided in Figure 4. High
 248 SFT scores can be biased toward *homogeneous* or
 249 *repeated examples* and are not reliably predictive
 250 of subsequent RL gains.
 251

252 This mismatch between post-SFT and post-RL
 253 performance is not directly visible from post-SFT
 254 models. As shown in Figure 3 where we fit a linear function between post-SFT and post-RL
 255 performance, these two performance correlates with $R^2 = 0.43$, indicating post-SFT performance
 256 only explains 43% of variation in the final outcome after RL whereas the gaps remain evident.
 257

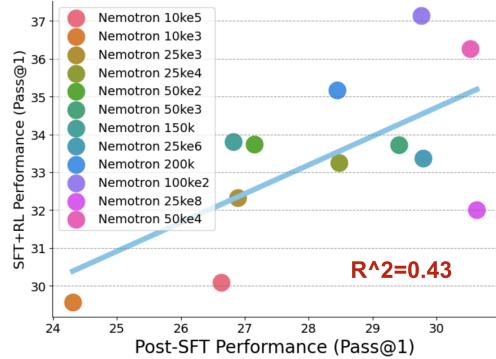


Figure 3: Llama3-8B-Instruct undergone SFT-RL with SFT examples from Llama-Nemotron-SFT dataset and RLVR via GRPO on MATH dataset (train-split). Reporting Pass@1 performance averaged over 7 math benchmarks. Linear fit between post-SFT performance and final outcome after RL. The two performance correlates with $R^2 = 0.43$, indicating *post-SFT performance explains only 43% of variation in the final outcome after RL and the remaining gaps are prominent*.

258

259

260

261

262

263

264

265

266

267

268

269

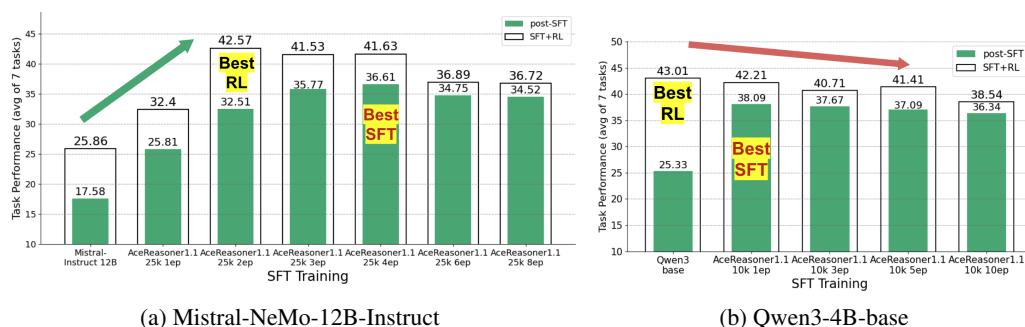


Figure 4: Both models undergone SFT-RL with SFT examples from AceReasoner1.1-SFT dataset and RLVR via GRPO on DeepScaleR dataset. Reporting Pass@1 performance averaged over 7 math benchmarks. When repeating SFT for more epochs on the same data, Mistral’s (left) SFT continues to improve with up to 4 epochs where the final performance after RL saturates after 2 epochs. Qwen3’s (right) final performance after RL degrades with SFT training, though, these models’ post-SFT performance is substantially higher than the base model. *Both cases show clear divergence between post-SFT performance and final performance after RL. Here, optimizing post-SFT performance will be suboptimal or ineffective for improving the final model.*

270 3.2 INSTANCE-LEVEL SCENARIOS
271

272 In this scenario, we fix the model and the training configurations but vary the SFT datasets. This
273 setup is primarily concerned with SFT data selection and curation, examining whether the strong
274 SFT performance promised by a particular dataset transfers to strong final performance after RL.
275 For instance, state-of-the-art data selection methods are often prone to selecting examples that are
276 more “natural” or easier for the model to learn (Zhang et al., 2025; Yu et al., 2025). While this
277 simpler data may allow the model to achieve high SFT metrics more quickly, we question whether
278 this comes at the cost of learning more difficult or advanced reasoning capabilities that are crucial for
279 downstream success. We identified similar gaps between post-SFT performance and final outcome
280 after RL. Visualizing representative examples in Figure 1, high SFT scores can be biased toward
281 *simpler examples* and are not reliably predictive of subsequent RL gains or scaled-up post-training
282 effectiveness. For example, *training on shortest examples led to faster performance improvements*
283 than training on randomly sampled examples during SFT. These shorter examples are closer to the
284 model’s original generations and easier to learn, though, these are not best examples for the model
285 to gain reasoning capabilities in preparation for RL. The final performance after RL is significantly
286 lower. These gaps are not directly captured in the post-SFT performance.

287 4 PROPOSED METRICS TOWARDS MORE RELIABLE PREDICTIONS
288289 4.1 GENERALIZATION LOSS ON VALIDATION EXAMPLES
290

291 During the investigation above, we identified a counterintuitive pattern in which post-SFT perfor-
292 mance improves stably when training for more epochs whereas the overtrained models show decreased
293 potentials during the subsequent RL. The best final performance after RL is not usually achieved on
294 models with the best post-SFT performance. To be able to optimize the final outcome on the given
295 training examples, one needs to optimize the SFT training paradigm based on the predicted final
296 outcome after RL. We materialize this insight and identify generalization loss after SFT to be a viable
297 indicator of the model’s potential during the subsequent RL. While repeating training for more epochs,
298 together with the improving post-SFT performance, we observe the generalization loss on validation
299 examples to elevate and eventually flare up, indicating strong over-fitting. This generalization loss
300 shows strong correlation with further performance gains during subsequent RL, allowing prediction
301 for the final outcome after RL (Figure 5). When using it in practice, after conducting SFT training
302 with different numbers of examples and epochs, we can immediately rule out post-SFT models with
303 both lower performance and higher generalization loss as they will likely remain underperforming
304 after the subsequent RL, facilitating determination of the best SFT training paradigm.

305 4.2 PASS@K ACCURACY EVALUATED AT LARGE K
306

307 The objective of RLVR via GRPO is to maximize expected reward, which explicitly optimizes the
308 Pass@1 accuracy on the RL tasks. GRPO only progresses when at least one of the responses for the
309 RL task is correct. Recent works argue that GRPO compresses Pass@k accuracy into Pass@1 (Yue
310 et al., 2025), and empirical evidence appears to support the argument showing GRPO mostly improves
311 average Pass@1 accuracy on tasks where the original model achieves an above-zero accuracy (Liu
312 et al., 2025b). Though it remains debatable whether GRPO discovers new solution traces beyond the
313 capabilities of the original model (Liu et al., 2025a), all these analyses and findings suggest RLVR
314 dynamics during GRPO training to be strongly coupled with the original models Pass@k accuracy.
315 Hu et al. (2023) pioneers in using the Pass@high metric to study the scaling of task performance.
316 The authors argue that Pass@k provides finer resolution to the Pass@1 metric and better captures
317 the underlying dynamics. Acting on this intuition, we consider Pass@k performance of the post-
318 SFT model, especially with large k, as a candidate metric for predicting its final outcome after the
319 subsequent RL. When using it in practice, after SFT training, we evaluate Pass@k performance on the
320 post-SFT models with different values of k. For efficient implementation, we leverage the following
321 formula which provides unbiased estimations for Pass@k accuracies for all $k \leq n$ (Brown et al.,
322 2024), $\text{Pass}@k = \mathbb{E} \left(1 - \frac{\binom{n-c}{k}}{\binom{n}{k}} \right)$ where integer n denotes the total number of responses generated
323 for the task, integer k denotes the target value for k Pass@k, and integer c denotes the number of
correct responses for the task, respectively.

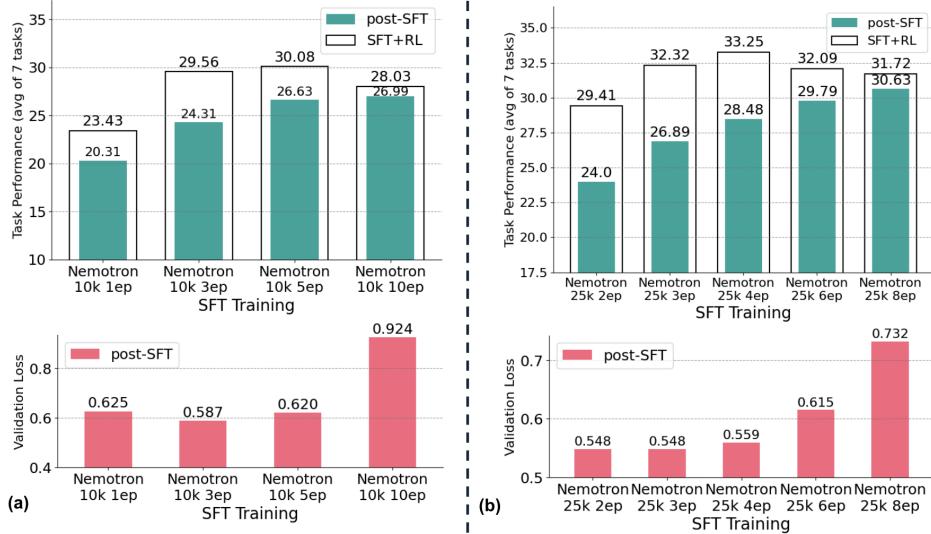


Figure 5: Llama3-8B-Instruct undergone SFT-RL with SFT examples from Llama-Nemotron-SFT dataset and RLVR via GRPO on MATH dataset (train-split). Reporting Pass@1 performance averaged over 7 math benchmarks and generalization loss on the validation set of SFT data. We identify generalization loss after SFT to be a viable indicator for the model’s RL potential. While repeating training for more epochs, together with the improving post-SFT performance, we observe the generalization loss on validation examples to elevate and eventually flare up, indicating strong over-fitting. *This generalization loss shows strong correlation with the further performance gain during the subsequent RL, allowing prediction for the final outcome after RL.*

We consider the Pass@large k performance as the indicator for the final outcome after RL and deem the post-SFT model with the best Pass@large k performance to have the best Pass@1 performance after RL. The best post-SFT model can be determined without needing to conduct any actual RL run.

5 EMPIRICAL EVALUATIONS

5.1 SETUP

We conduct three sets of experiments with SFT-RL post-training. On **Llama3-8B-Instruct** models, we conduct SFT training with examples from Llama-Nemotron dataset (where we only select math samples with responses generated by QwQ-32B (Team, 2025) or DeepSeek R1 (Guo et al., 2025), hereinafter the same) and RL training on MATH dataset (train-split) (Hendrycks et al., 2021); on **Mistral-Nemo-12B-Instruct** and Qwen3-4B-base, we conduct SFT training with examples from AceReasoner1.1-SFT dataset and RL training on DeepScaleR dataset (Luo et al., 2025). For all models, we conduct RL training for 3 epochs where each run takes up to 5 days. We repeat RL training for 4+ runs on each data recipe and training paradigm, conduct 4+ evaluations on different checkpoints across RL training run, and report the best performance for the model. We evaluate task performance on 7 math benchmarks, **MATH-500** (Hendrycks et al., 2021), **AIME 1983-2024** (Veeraboina, 2023), **GSM8k** (Cobbe et al., 2021), **AIME 2025** (of America, 2025), **AMC** (Competitions, 2025), **Olympiad** (He et al., 2024), **Minerva** (Lewkowycz et al., 2022), and report model performance as Pass@1 accuracy averaged over 64 repetitions and across 7 tasks. For the proposed predictors, we evaluate the generalization loss on the validation set of the SFT data and Pass@64 accuracy averaged over 256 repetitions. Experiments spent >1M GPU hours on NVIDIA A100. Please refer to Appendix B for additional details. Shown in Figure 2 (right), in this setup, Qwen3-series models do not appear to benefit from state-of-the-art SFT datasets, and models undergone different SFT training achieve considerably close final performance after RL. Since this work focuses on studying the impact of different SFT training on the subsequent RL, we present these results as qualitative examples instead (deferred to Appendix A).

Following the categorization above, we organize experiments in two major scenarios: **dataset-level** prediction, and **instance-level** prediction. In dataset-level prediction experiments: we conduct SFT training for the base model on samples from math reasoning datasets with different training paradigms (varying number of examples and epochs). In instance-level prediction experiments: we

378 first create diverse different curated SFT datasets by selecting the shortest/longest subsets, random
 379 samples, or their different mixtures (samples are shown in Appendix E). Then, we conduct SFT
 380 training for the base model on samples from each curated dataset with the same training paradigms
 381 (one epoch). We consider two primary metrics measure prediction performance, **Coefficient of**
 382 **determination (R^2)** (Pearson, 1909), and **Spearman’s rank correlation coefficient (Spearman)**
 383 (Zar, 1972). Specifically, R^2 measures the proportion of variation in the prediction variable (final
 384 performance) that is unexplained by the predictor, examining the accuracy of prediction on the final
 385 performance after RL. Spearman yields a number ranging from -1 to 1 that indicates how strongly
 386 two sets of ranks are correlated, which we use to examine the effectiveness in identifying post-SFT
 387 models that lead to the best final performance. Additional results can be found in Appendix C.
 388

389 5.2 USE CASE 1: DATASET-LEVEL PREDICTION

390 This use case focuses on optimizing the SFT training paradigm, a common dataset-level challenge.
 391 Given a fixed compute budget, practitioners must decide on the optimal trade-off between the volume
 392 of unique data and the number of training epochs, navigating the risks of under- and over-training.
 393 We test the predictive power of our proposed metrics against the baseline of using post-SFT Pass@1
 394 accuracy. To examine the accuracy of prediction with R^2 , we randomly select 50% SFT models and
 395 fit a linear function between their post-SFT performance and final performance after RL. The fitted
 396 function is then used to predict the final performance of the other 50% SFT models. We compare the
 397 predictions to their actual post-RL outcomes to compute R^2 . We repeat the random sampling for 100
 398 times and report the standard error.

399 Table 1: Spearman’s rank correlation between performance predicted from post-SFT models and the actual
 400 performance after RL. Both generalization loss and Pass@64 achieve notable margins over prediction from
 401 Pass@1, whereas averging the two prediction may or may not lead to better results.

403 Spearman’s Rank 404 Correlation / Models	405 Prediction based on SFT Pass@1 (avg. of 64) baseline	406 Prediction based on SFT Generalization Loss	407 Prediction based on SFT Pass@Large k (k=64)	408 Avg. Prediction from SFT Gen. Loss + Pass@Large k (64)
409 Llama3-8B-Instruct	410 0.75	411 0.94	412 0.95	413 0.97 (+0.22)
414 Mistral-NeMo-12B-Instruct	415 0.78	416 0.90	417 0.92 (+0.14)	418 0.90

419 Table 2: Measuring prediction accuracy with coefficient of determination (R^2). We randomly select 50% SFT
 420 models and fit a linear function between their post-SFT performance and performance after RL, and use it to
 421 predict for the other 50% SFT models. We repeat random sampling for 100 times and report standard errors.

422 Coefficient of 423 determination (R^2) / Models	424 Prediction based on SFT Pass@1 (avg. of 64) baseline	425 Prediction based on SFT Generalization Loss	426 Prediction based on SFT Pass@Large k (k=64)	427 Avg. Prediction from SFT Gen. Loss + Pass@Large k (64)
428 Llama3-8B-Instruct	429 0.57 ± 0.29	430 0.88 ± 0.09	431 0.87 ± 0.10	432 $0.94 \pm 0.04 (+0.37)$
433 Mistral-NeMo-12B-Instruct	434 0.29 ± 0.38	435 $0.79 \pm 0.26 (+0.50)$	436 0.57 ± 0.32	437 0.72 ± 0.24

438 Takeaway 1: Dataset-level Prediction

- 439 Both generalization loss and Pass@large k are effective predictors for post-RL performance
 440 when optimizing SFT training configurations on a single dataset, providing higher-accuracy
 441 estimates that help guide decisions and save significant compute.
- 442 Both predictors excel at identifying correct rankings for post-RL performance, achieving \geq
 443 0.90 Spearman correlation (30% improvements); generalization loss provides advantageous
 444 prediction accuracy (R^2) for post-RL performance with up to 2x improvements.

453 5.3 USE CASE 2: INSTANCE-LEVEL PREDICTION

454 This use case addresses the challenge of SFT data selection, an instance-level optimization problem.
 455 Here, the training pipeline is fixed, but we aim to select the optimal SFT dataset from a pool of
 456 candidates curated with different strategies (e.g., selecting for shortest/longest solutions, diversity, etc.
 457 Ye et al. (2025)). This scenario tests whether strong SFT performance on a given dataset translates to
 458 a good final outcome after RL.

432 Table 3: Spearman’s rank correlation between performance predicted from post-SFT models and the actual
 433 performance after RL. Pass@64 achieve notable margins over prediction from Pass@1.

434 Spearman’s Rank/ 435 Correlation / Models	436 Prediction based on SFT Pass@1 (avg. of 64) baseline	437 Prediction based on SFT Pass@Large k (k=64)
Llama3-8B-Instruct	0.69	0.94 (+0.25)
Mistral-NeMo-12B-Instruct	0.70	0.98 (+0.28)

438 Table 4: Measuring prediction accuracy with coefficient of determination (R^2). We randomly select 50% SFT
 439 models and fit a linear function between their post-SFT performance and performance after RL, and use it to
 440 predict for the other 50% SFT models. We repeat random sampling for 100 times and report standard errors.

441 Coefficient of determination (R^2) / Models	442 Prediction based on SFT Pass@1 (avg. of 64) baseline	443 Prediction based on SFT Pass@Large k (k=64)
Llama3-8B-Instruct	0.58 ± 0.20	0.92 ± 0.05 (+0.34)
Mistral-NeMo-12B-Instruct	0.73 ± 0.16	0.98 ± 0.01 (+0.25)

444 *In this scenario, the generalization loss predictor is not applicable.* Since each SFT dataset comes
 445 from a different distribution, the validation loss includes a distributional gap component in addition
 446 to generalization error. Without a common, representative validation set, it is difficult to make a fair
 447 comparison. Pass@large k metric proves to be exceptionally robust. Since it measures the model’s
 448 inherent capability to produce correct solutions, it is less sensitive to distributional shifts in the
 449 training data. It can be used to effectively rank different SFT datasets and select the one with the
 450 highest potential for RL, without needing to run any RL experiments for calibration.

451 **Takeaway 2: Instance-level Prediction**

452

- 453 Pass@large k turns out highly accurate and robust in instance-level predictions, improving
 454 Spearman correlation by up to 36% and prediction accuracy (R^2) by up to 59%. It effectively
 455 identifies datasets for strong post-RL performance and predicts RL outcomes.
- 456 Generalization loss is not applicable for instance-level selection due to distributional gaps
 457 between different datasets.

458 **How to use them in practice?** Our metrics support two primary workflows. If the goal is simply
 459 to *rank* SFT candidates, one can use generalization loss to quickly filter out clearly suboptimal
 460 models (i.e., those with both low performance and high loss). Then, Pass@large k can be used to
 461 reliably rank the remaining candidates to identify the most promising one. If the goal is to *predict*
 462 the final performance value—for instance, to inform trade-offs between SFT costs and expected
 463 gains—practitioners can run RL on a small number of SFT models to gather calibration data. A linear
 464 predictor can then be fitted using our proposed metrics, allowing for accurate performance estimation
 465 across all SFT candidates without the need for exhaustive RL runs.

466 **6 CONCLUSIONS**

467 This work confronts a critical quagmire in reasoning post-training: the common assumption that high
 468 SFT scores guarantee strong performance after subsequent RL. Through extensive experimentation
 469 with Llama3/Mistral-Nemo/Qwen3 models spending >1M GPU hours, we provide broad counter-
 470 examples where SFT performance is often misleading or biased toward simpler/repeated data. Our
 471 primary contribution is the identification and validation of two more reliable predictors for post-RL
 472 success: generalization loss on held-out reasoning examples and Pass@large k accuracy, improving
 473 prediction accuracy (R^2) and Spearman’s rank correlation by up to 0.5 (2x) over prediction from
 474 post-SFT performance. By allowing practitioners to better predict the final outcome, our work helps
 475 de-risk the expensive RL stage and streamline the entire post-training pipeline. We will open-source
 476 our enhanced evaluation tool to facilitate broader adoption and further research. This work focuses
 477 on mathematical reasoning. A natural **next step** is to study the topic in a wider range of reasoning
 478 tasks (e.g., coding, science) and agentic use cases; Our study is limited to the prevailing paradigm
 479 of online RL with GRPO. The relationship between SFT characteristics and post-RL performance
 480 with other methods such as offline RL/DPO or other RL algorithms may worth further explorations;
 481 Directly evaluating Pass@large k requires repeating evaluation for at least k times, which becomes
 482 computational expensive with long sequence lengths. Estimating Pass@k accuracy from that of
 483 smaller k holds the promise for more efficient evaluations (Schaeffer et al., 2025).

486 REFERENCES
487

488 Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar,
489 Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical
490 report. *arXiv preprint arXiv:2412.08905*, 2024.

491 Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and
492 Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling.
493 *arXiv preprint arXiv:2407.21787*, 2024.

494 Zhepeng Cen, Yihang Yao, William Han, Zuxin Liu, and Ding Zhao. Behavior injection: Preparing
495 language models for reinforcement learning. *arXiv preprint arXiv:2505.18917*, 2025.

496

497 Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang
498 Xie. Sft or r1? an early investigation into training r1-like reasoning large vision-language models.
499 *arXiv preprint arXiv:2504.11468*, 2025a.

500 Yang Chen, Zhuolin Yang, Zihan Liu, Chankyu Lee, Peng Xu, Mohammad Shoeybi, Bryan Catanzaro,
501 and Wei Ping. Acereason-nemotron: Advancing math and code reasoning through reinforcement
502 learning. *arXiv preprint arXiv:2505.16400*, 2025b.

503

504 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
505 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John
506 Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*,
507 2021.

508 American Mathematics Competitions. 2021/2022 amc problems and solutions, 2025. URL <https://huggingface.co/datasets/AI-MO/aimo-validation-amc>.

509

510 Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman,
511 Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, et al. Language models scale reliably
512 with over-training and on downstream tasks. *arXiv preprint arXiv:2403.08540*, 2024.

513

514 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad
515 Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of
516 models. *arXiv preprint arXiv:2407.21783*, 2024.

517

518 Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna
519 Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning
520 models. *arXiv preprint arXiv:2506.04178*, 2025.

521

522 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
523 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
524 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

525

526 Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu,
527 Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for
528 promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint
529 arXiv:2402.14008*, 2024.

530

531 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,
532 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv
533 preprint arXiv:2103.03874*, 2021.

534

535 Shengding Hu, Xin Liu, Xu Han, Xinrong Zhang, Chaoqun He, Weilin Zhao, Yankai Lin, Ning Ding,
536 Zebin Ou, Guoyang Zeng, et al. Predicting emergent abilities with infinite resolution evaluation.
537 *arXiv preprint arXiv:2310.03262*, 2023.

538

539 Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL <https://github.com/huggingface/open-r1>.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal,
540 Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. Llm
541 post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321*,
542 2025.

540 Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E.
 541 Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model
 542 serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating
 543 Systems Principles*, 2023.

544 Hynek Kydlíček. Math-verify: Math verification library. [https://github.com/
 545 huggingface/math-verify](https://github.com/huggingface/math-verify), accessed on 2025-09-25, 2025.

546 Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman,
 547 Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in
 548 open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

549 Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ra-
 550 masesh, Ambrose Sloane, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative
 551 reasoning problems with language models. *Advances in neural information processing systems*,
 552 35:3843–3857, 2022.

553 Yang Li, Youssef Emad, Karthik Padthe, Jack Lanchantin, Weizhe Yuan, Thao Nguyen, Jason Weston,
 554 Shang-Wen Li, Dong Wang, Ilia Kulikov, et al. Naturalthoughts: Selecting and distilling reasoning
 555 traces for general reasoning tasks. *arXiv preprint arXiv:2507.01921*, 2025.

556 Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao,
 557 Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint
 558 arXiv:2412.19437*, 2024.

559 Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong.
 560 Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models.
 561 *arXiv preprint arXiv:2505.24864*, 2025a.

562 Zihan Liu, Zhuolin Yang, Yang Chen, Chankyu Lee, Mohammad Shoeybi, Bryan Catanzaro, and Wei
 563 Ping. Acereason-nemotron 1.1: Advancing math and code reasoning through sft and rl synergy.
 564 *arXiv preprint arXiv:2506.13284*, 2025b.

565 Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai,
 566 Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview
 567 with a 1.5b model by scaling rl, 2025. Notion Blog.

568 Sajee Mathew and J Varia. Overview of amazon web services. *Amazon Whitepapers*, 105(1):22,
 569 2014.

570 AI Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation.
 571 <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>, checked on, 4(7):2025, 2025.

572 Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke
 573 Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time
 574 scaling. *arXiv preprint arXiv:2501.19393*, 2025.

575 Mathematical Association of America. 2025 aime, 2025. URL [https://huggingface.co/
 576 datasets/yentinglin/aime_2025](https://huggingface.co/datasets/yentinglin/aime_2025).

577 Karl Pearson. Determination of the coefficient of correlation. *Science*, 30(757):23–25, 1909.

578 Abhinav Rastogi, Albert Q Jiang, Andy Lo, Gabrielle Berrada, Guillaume Lample, Jason Rute, Joep
 579 Barmentlo, Karmesh Yadav, Kartik Khandelwal, Khyathi Raghavi Chandu, et al. Magistral. *arXiv
 580 preprint arXiv:2506.10910*, 2025.

581 Rylan Schaeffer, Joshua Kazdan, John Hughes, Jordan Juravsky, Sara Price, Aengus Lynch, Erik
 582 Jones, Robert Kirk, Azalia Mirhoseini, and Sanmi Koyejo. How do large language monkeys get
 583 their power (laws)? *arXiv preprint arXiv:2502.17578*, 2025.

584 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
 585 Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical
 586 reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

594 Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng,
 595 Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint*
 596 *arXiv*: 2409.19256, 2024.

597 Soumye Singhal, Jiaqi Zeng, Alexander Bukharin, Yian Zhang, Gerald Shen, Ameya Sunil Maha-
 598 baleshwarkar, Bilal Kartal, Yoshi Suhara, Akhiad Bercovich, Itay Levy, et al. Llama-nemotron:
 599 Efficient reasoning models. In *The Exploration in AI Today Workshop at ICML 2025*.

600 601 Mistral AI team. Mistral nemo. <https://mistral.ai/news/mistral-nemo>, accessed on 2025-09-25, 2024.

602 603 Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL
 604 <https://qwenlm.github.io/blog/qwq-32b/>.

605 606 Edan Toledo, Karen Hambardzumyan, Martin Josifoski, Rishi Hazra, Nicolas Baldwin, Alexis
 607 Audran-Reiss, Michael Kuchnik, Despoina Magka, Minqi Jiang, Alisia Maria Lupidi, et al. Ai
 608 research agents for machine learning: Search, exploration, and generalization in mle-bench. *arXiv*
preprint arXiv:2507.02554, 2025.

609 610 Hemish Veeraboina. Aime problem set 1983-2024, 2023. URL <https://www.kaggle.com/datasets/hemishveeraboina/aime-problem-set-1983-2024>.

611 612 Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen,
 613 Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive
 614 effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

615 616 Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu
 617 Tang, Xiaowei Lv, et al. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond.
arXiv preprint arXiv:2503.10460, 2025.

618 619 Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhan Zhou,
 620 Huijie Lv, Ming Zhang, et al. Reasoning or memorization? unreliable results of reinforcement
 621 learning due to data contamination. *arXiv preprint arXiv:2507.10532*, 2025.

622 623 Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Rui Feng Xu,
 624 and Fei Mi. Kdrl: Post-training reasoning llms via unified knowledge distillation and reinforcement
 625 learning. *arXiv preprint arXiv:2506.02208*, 2025.

626 627 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang
 628 Gao, Chengan Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*,
 2025.

629 630 Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for
 631 reasoning. *arXiv preprint arXiv:2502.03387*, 2025.

632 633 Ping Yu, Weizhe Yuan, Olga Golovneva, Tianhao Wu, Sainbayar Sukhbaatar, Jason Weston, and Jing
 634 Xu. Rip: Better models by survival of the fittest prompts. *arXiv preprint arXiv:2501.18578*, 2025.

635 636 Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Ilia Kulikov, Kyunghyun Cho, Dong
 637 Wang, Yuandong Tian, Jason E Weston, et al. Naturalreasoning: Reasoning in the wild with 2.8 m
 638 challenging questions. *arXiv preprint arXiv:2502.13124*, 2025.

639 640 Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does
 641 reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv*
preprint arXiv:2504.13837, 2025.

642 643 Jerrold H Zar. Significance testing of the spearman rank correlation coefficient. *Journal of the*
American Statistical Association, 67(339):578–580, 1972.

644 645 Dylan Zhang, Qirun Dai, and Hao Peng. The best instruction-tuning data are those that fit. *arXiv*
preprint arXiv:2502.04194, 2025.

646 647 Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyuan Luo, Zhangchi Feng, and
 648 Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv*
preprint arXiv:2403.13372, 2024.

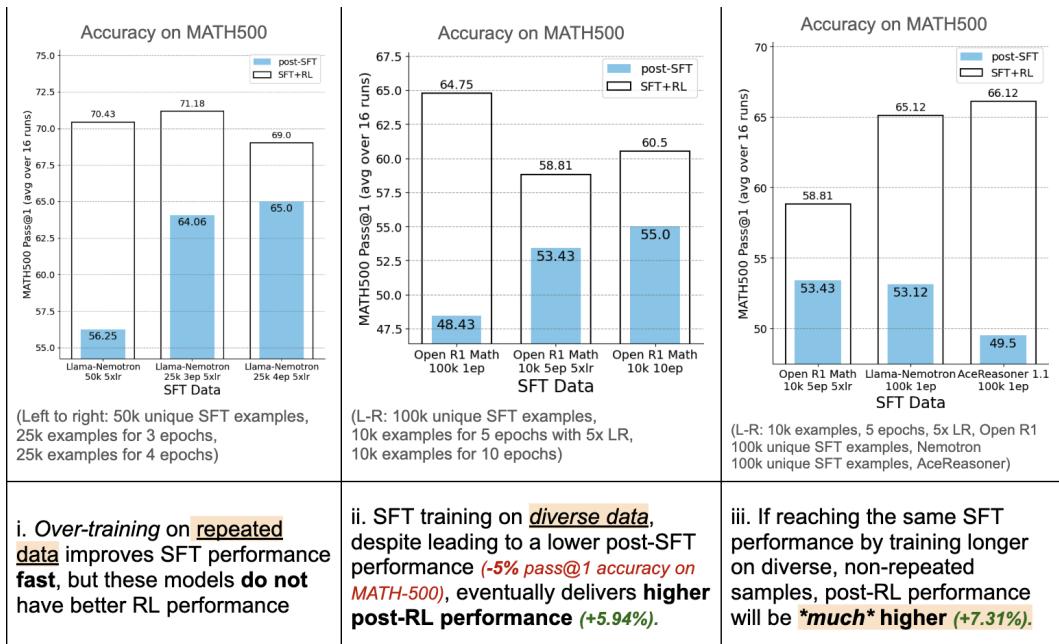
Appendices

648	A Additional SFT-RL Examples and Visualizations	14
649	A.1 Llama3-8B-Instruct	14
650	A.2 Mistral-NeMo-12B-Instruct	15
651	A.3 Qwen3-4B-base	15
652	B Implementation Details	16
653	B.1 Models and Datasets	17
654	B.2 Training	17
655	B.3 Evaluation	17
656	C Additional Experimental Results	18
657	C.1 Dataset-level	18
658	C.2 Instance-level	18
659	D Broader Investigations and Supplementary Examples	21
660	D.1 Fine-grained comparisons: Best SFT model vs. Best SFT+RL model	21
661	D.2 Ablation Studies: Pass@k, Entropy	23
662	E Sample SFT Examples	26
663	E.1 Shortest Examples	26
664	E.2 Longest Examples	29
665		
666		
667		
668		
669		
670		
671		
672		
673		
674		
675		
676		
677		
678		
679		
680		
681		
682		
683		
684		
685		
686		
687		
688		
689		
690		
691		
692		
693		
694		
695		
696		
697		
698		
699		
700		
701		

702 A ADDITIONAL SFT-RL EXAMPLES AND VISUALIZATIONS

703 A.1 LLAMA3-8B-INSTRUCT

707 Figure 6 shows results on Llama3-8B-Instruct undergone SFT-RL with SFT examples from Llama-
 708 Nemotron-SFT/AceReasoner1.1-SFT/OpenR1-Math (Hugging Face, 2025) dataset and RLVR via
 709 GRPO on MATH dataset (train-split). Reporting average Pass@1 performance on MATH-500 (test-
 710 split). *High SFT scores can be biased toward simpler or more homogeneous data and are not reliably*
 711 *predictive of subsequent RL gains or post-training effectiveness.* SFT on fewer unique examples
 712 repeated for more training epochs (ep) or/and with a larger learning rate (LR) leads to higher accuracy
 713 on reasoning benchmarks such as MATH-500 (+8.75% vs. non-repeated data, left figure). However,
 714 models trained this way show smaller improvements during RL (-1.43% vs. non-repeated). In
 715 contrast, SFT on more diverse, non-repeated data—despite yielding lower initial SFT performance
 716 (-5% vs. repeated data, middle/right figure)—results in significantly better post-RL performance
 717 (+5.94%).



733
 734 i. Over-training on repeated
 735 data improves SFT performance
 736 **fast**, but these models **do not**
 737 have better RL performance

ii. SFT training on diverse data,
 738 despite leading to a lower post-SFT
 739 performance (**-5% pass@1 accuracy on**
 740 **MATH-500**), eventually delivers **higher**
 741 **post-RL performance (+5.94%).**

iii. If reaching the same SFT
 742 performance by training longer
 743 on diverse, non-repeated
 744 samples, post-RL performance
 745 will be ***much* higher (+7.31%).**

738 Figure 6: Llama3-8B-Instruct undergone SFT-RL with SFT examples from Llama-Nemotron-
 739 SFT/AceReasoner1.1-SFT/OpenR1-Math dataset and RLVR via GRPO on MATH dataset (train-split). Reporting
 740 average Pass@1 performance on MATH-500 (test-split). *High SFT scores can be biased toward simpler or more*
 741 *homogeneous data and are not reliably predictive of subsequent RL gains or post-training effectiveness.* SFT
 742 on fewer unique examples repeated for more training epochs (ep) or/and with a larger learning rate (LR) leads to higher accuracy
 743 on reasoning benchmarks such as MATH-500 (+8.75% vs. non-repeated data, left figure). However,
 744 models trained this way show smaller improvements during RL (-1.43% vs. non-repeated). In contrast,
 745 SFT on more diverse, non-repeated data—despite yielding lower initial SFT performance (-5% vs. repeated data,
 746 middle/right figure)—results in significantly better post-RL performance (+5.94%).

748 Figure 7 shows results on Llama3-8B-Instruct undergone SFT-RL with SFT examples from Llama-
 749 Nemotron-SFT dataset and RLVR via GRPO on MATH dataset (train-split). Reporting Pass@1
 750 performance averaged over 7 math benchmarks. High SFT scores can be biased toward *simpler*
 751 *examples* and are not reliably predictive of subsequent RL gains or scaled-up post-training effec-
 752 *tiveness*. For example, training on shortest examples (e.g., s10k, s500k) led to faster performance
 753 improvements than training on randomly sampled examples (e.g., 10k, 200k) during SFT (lower
 754 smaller dots). These shorter examples are closer to the model’s original generations and easier to
 755 learn, though, these are not best examples for the model to gain reasoning capabilities in preparation
 for RL. The final performance after RL (upper larger dots) is significantly lower.

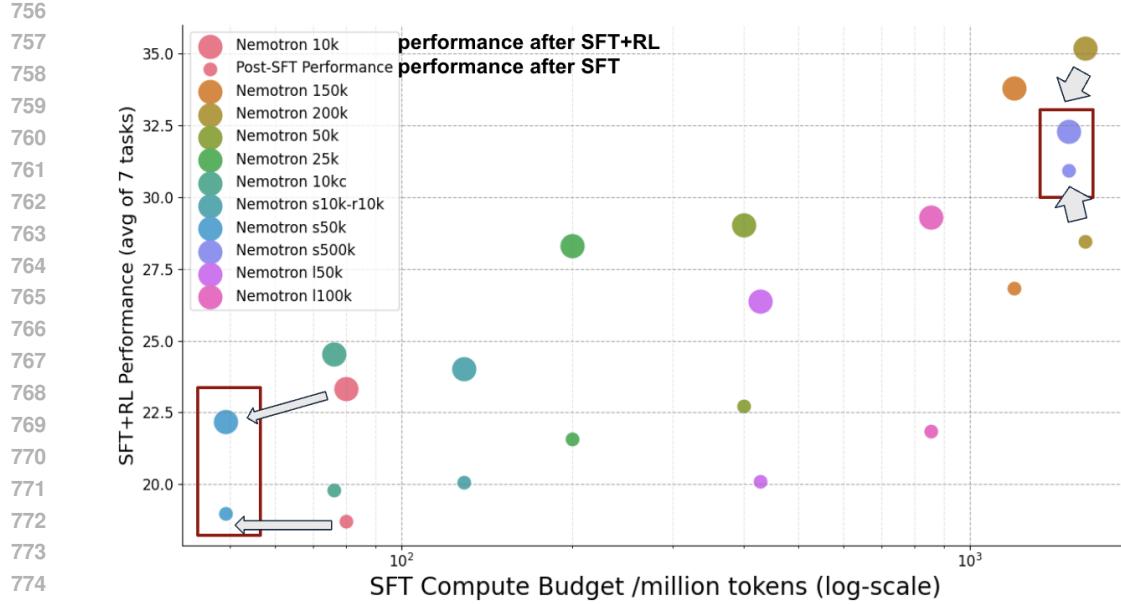


Figure 7: Llama3-8B-Instruct undergone SFT-RL with SFT examples from Llama-Nemotron-SFT dataset and RLVR via GRPO on MATH dataset (train-split). Reporting Pass@1 performance averaged over 7 math benchmarks. High SFT scores can be biased toward *simpler examples* and are not reliably predictive of subsequent RL gains or scaled-up post-training effectiveness. For example, training on shortest examples (e.g., s10k, s500k) led to faster performance improvements than training on randomly sampled examples (e.g., 10k, 200k) during SFT (lower smaller dots). These shorter examples are closer to the model’s original generations and easier to learn, though, these are not best examples for the model to gain reasoning capabilities in preparation for RL. The final performance after RL (upper larger dots) is significantly lower.

A.2 MISTRAL-NEMO-12B-INSTRUCT

Figure 8 shows results on Mistral-NeMo-12B-Instruct undergone SFT-RL with **shortest** SFT examples from AceReasoner1.1-SFT dataset and RLVR via GRPO on DeepScaleR dataset. Reporting Pass@1 performance averaged over 7 math benchmarks. With increasing SFT examples, Mistral’s post-SFT performance first dips and then gradually recovers and improves to performance better than before SFT training. Compared to the base model, the final performance after RL also first dips and then gradually goes up and improves to a better level. *Notably, post-RL performance recovers to the same level as the base model slower than the post-SFT performance.* The post-SFT and post-RL performance trends are not identical.

A.3 QWEN3-4B-BASE

Figure 9 shows results on Qwen3-4B-base undergone SFT-RL with **shortest** SFT examples from AceReasoner1.1-SFT dataset and RLVR via GRPO on DeepScaleR dataset. Reporting Pass@1 performance averaged over 7 math benchmarks. With increasing SFT examples, Qwen3’s post-SFT performances appear uncorrelated with the final performance after RL, where the latter remains the same despite the substantially improved SFT performance.

Figure 12 shows results on Qwen3-4B-base undergone SFT-RL with **Shortest/Longest-/Longest+Shortest** SFT examples from AceReasoner1.1-SFT dataset and RLVR via GRPO on DeepScaleR dataset. Reporting Pass@1 performance averaged over 7 math benchmarks. All SFT training substantially improves Qwen3’s post-SFT performance, but the final performance after RL is mixed. Training on **Longest** and 10k **Longest+10k Shortest** SFT examples lead to visibly improved final performance after RL where the latter achieves the best final performance for Qwen3 models in this work. Other SFT training lead to significantly degraded final performance after RL.

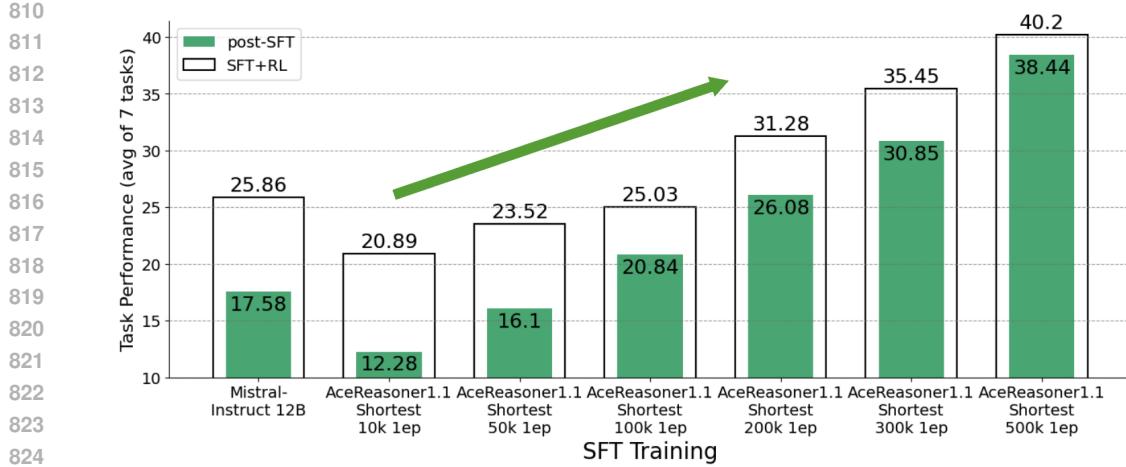


Figure 8: Mistral-NeMo-12B-Instruct undergone SFT-RL with **shortest** SFT examples from AceReasoner1.1-SFT dataset and RLVR via GRPO on DeepScaleR dataset. Reporting Pass@1 performance averaged over 7 math benchmarks. With increasing SFT examples, Mistral’s post-SFT performance first dips and then gradually recovers and improves to performance better than before SFT training. Compared to the base model, the final performance after RL also first dips and then gradually goes up and improves to a better level. *Notably, post-RL performance recovers to the same level as the base model slower than the post-SFT performance.* The post-SFT and post-RL performance trends are not identical.

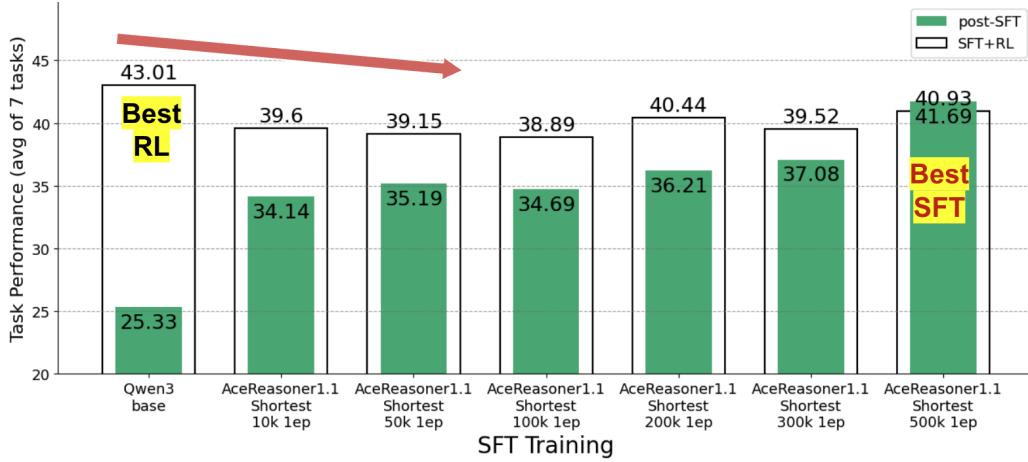


Figure 9: Qwen3-4B-base undergone SFT-RL with **shortest** SFT examples from AceReasoner1.1-SFT dataset and RLVR via GRPO on DeepScaleR dataset. Reporting Pass@1 performance averaged over 7 math benchmarks. With increasing SFT examples, Qwen3’s post-SFT performances appear uncorrelated with the final performance after RL, where the latter remains the same despite the substantially improved SFT performance.

B IMPLEMENTATION DETAILS

All experiments (SFT, RL, evaluation) are conducted on individual AWS (Mathew & Varia, 2014) node with 8x NVIDIA A100 80GB GPU. Experiments spent >1M GPU hours on NVIDIA A100 80GB. We repeat RL training for 4+ runs on each data recipe and training paradigm (each run takes up to 5 days), conduct 4+ evaluations on different checkpoints across RL training run, and report the best performance for the model. We set the max sequence length to 8k tokens throughout SFT, RL, and evaluation.

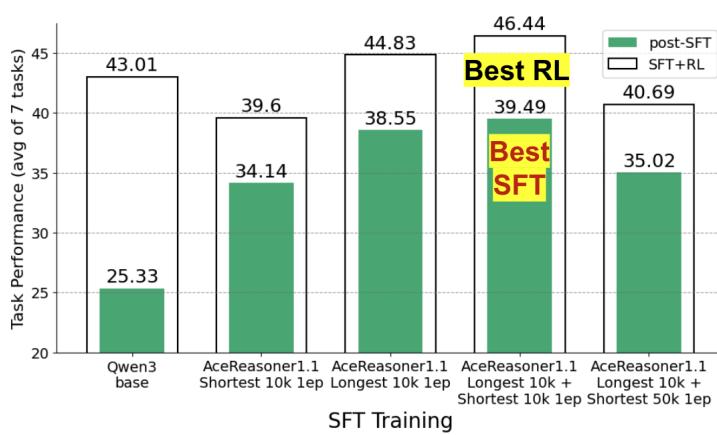


Figure 10: Qwen3-4B-base undergone SFT-RL with **shortest/Longest/Longest+Shortest** SFT examples from AceReasoner1.1-SFT dataset and RLVR via GRPO on DeepScaleR dataset. Reporting Pass@1 performance averaged over 7 math benchmarks. All SFT training substantially improves Qwen3’s post-SFT performance, but the final performance after RL is mixed. Training on **Longest** and 10k **Longest+10k Shortest** SFT examples lead to visibly improved final performance after RL where the latter achieves the best final performance for Qwen3 models in this work. Other SFT training lead to significantly degraded final performance after RL.

B.1 MODELS AND DATASETS

We conduct three sets of experiments with SFT-RL post-training. On **Llama3-8B-Instruct** models, we conduct SFT training with examples from Llama-Nemotron dataset (where we only select math samples with responses generated by QwQ-32B (Team, 2025) or DeepSeek R1 (Guo et al., 2025), hereinafter the same) and RL training on MATH dataset (train-split) (Hendrycks et al., 2021); on **Mistral-Nemo-12B-Instruct** and Qwen3-4B-base, we conduct SFT training with examples from AceReasoner1.1-SFT dataset and RL training on DeepScaleR dataset (Luo et al., 2025). For all models, we conduct RL training for 3 epochs where each run takes up to 5 days.

B.2 TRAINING

We conduct SFT training with LLaMA-Factory (Zheng et al., 2024) using learning rates lr=1e-5 and global batch size = 128, and RL training via GRPO with verl (Sheng et al., 2024) using learning rates lr=1e-6 and global batch size = 128. We sample 16 rollouts for each question with temperature=1.0. We set KL loss coefficient=0 and entropy coefficient=0.001.

B.3 EVALUATION

Evaluations are conducted with pipelines originally developed in this work based on vllm (Kwon et al., 2023) and HuggingFace’s math-verify (Kydlíček, 2025), enabling efficient inference with performant and accurate verification. We ran evaluations with the same template and generation configuration as in RL, using decoding temperature t=1.0 and the standard reasoning prompt (“Let’s think step by step and output the final answer within \boxed{}.”).

We evaluate task performance on 7 math benchmarks, including **MATH-500** (Hendrycks et al., 2021), **AIME 1983-2024** (Veeraboina, 2023), **GSM8k** (Cobbe et al., 2021), **AIME 2025** (of America, 2025), **AMC** (Competitions, 2025), **Olympiad** (He et al., 2024), **Minerva** (Lewkowycz et al., 2022), and report model performance as Pass@1 averaged over 64 repetitions and across 7 tasks. For the proposed predictors, we evaluate the generalization loss on the validation set of the SFT data and Pass@64 accuracy averaged over 256 repetitions.

918 C ADDITIONAL EXPERIMENTAL RESULTS
919920 C.1 DATASET-LEVEL
921

922 Table 5 shows results on Llama3-8B-Instruct undergone SFT-RL with SFT examples from Llama-
923 Nemotron-SFT dataset and RLVR via GRPO on MATH dataset (train-split). Reporting Pass@1
924 performance averaged over 7 math benchmarks. Measuring prediction accuracy with coefficient of
925 determination (R^2) varying the ratio of fit-validation datapoints. We randomly select x SFT models
926 and fit a linear function between their post-SFT performance and performance after RL, and use it to
927 predict for the rest SFT models. We repeat random sampling for 100 times and report standard errors.

928 Table 6 shows results on Mistral-NeMo-12B-Instruct undergone SFT-RL with SFT examples from
929 AceReasoner1.1-SFT dataset and RLVR via GRPO on DeepScaleR dataset. Measuring prediction ac-
930 curacy with coefficient of determination (R^2) varying the ratio of fit-validation datapoints. Reporting
931 Pass@1 performance averaged over 7 math benchmarks. We randomly select x SFT models and fit a
932 linear function between their post-SFT performance and performance after RL, and use it to predict
933 for the rest SFT models. We repeat random sampling for 100 times and report standard errors.

934 Table 5: Llama3-8B-Instruct undergone SFT-RL with SFT examples from Llama-Nemotron-SFT
935 dataset and RLVR via GRPO on MATH dataset (train-split). Reporting Pass@1 performance averaged
936 over 7 math benchmarks. Measuring prediction accuracy with coefficient of determination (R^2)
937 varying the ratio of fit-validation datapoints. We randomly select x SFT models and fit a linear
938 function between their post-SFT performance and performance after RL, and use it to predict for the
939 rest SFT models. We repeat random sampling for 100 times and report standard errors.

No. of Fitting-Validation Datapoints / Coefficient of determination (R^2)	Prediction based on SFT Pass@1 (avg. of 64) baseline	Prediction based on SFT Generalization Loss	Prediction based on SFT Pass@Large k (k=64)	Avg. Prediction from SFT Gen. Loss + Pass@Large k (64)
Fitting: 3; Validation: 13	0.48 \pm 0.40	0.80 \pm 0.33	0.80 \pm 0.23	0.86 \pm 0.22 (+0.38)
Fitting: 4; Validation: 12	0.57 \pm 0.29	0.82 \pm 0.21	0.84 \pm 0.15	0.92 \pm 0.08 (+0.35)
Fitting: 5; Validation: 11	0.57 \pm 0.29	0.88 \pm 0.09	0.87 \pm 0.10	0.94 \pm 0.04 (+0.37)
Fitting: 6; Validation: 10	0.57 \pm 0.26	0.89 \pm 0.07	0.87 \pm 0.10	0.95 \pm 0.03 (+0.38)
Fitting: 7; Validation: 9	0.64 \pm 0.19	0.89 \pm 0.07	0.90 \pm 0.06	0.95 \pm 0.05 (+0.31)
Fitting: 8; Validation: 8	0.64 \pm 0.20	0.88 \pm 0.08	0.88 \pm 0.08	0.93 \pm 0.05 (+0.29)
Fitting: 10; Validation: 6	0.59 \pm 0.33	0.85 \pm 0.17	0.85 \pm 0.15	0.94 \pm 0.05 (+0.35)
Fitting: 12; Validation: 4	0.54 \pm 0.43	0.86 \pm 0.18	0.81 \pm 0.23	0.91 \pm 0.12 (+0.37)

950
951 Table 6: Mistral-NeMo-12B-Instruct undergone SFT-RL with SFT examples from AceReasoner1.1-
952 SFT dataset and RLVR via GRPO on DeepScaleR dataset. Measuring prediction accuracy with
953 coefficient of determination (R^2) varying the ratio of fit-validation datapoints. Reporting Pass@1
954 performance averaged over 7 math benchmarks. We randomly select x SFT models and fit a linear
955 function between their post-SFT performance and performance after RL, and use it to predict for the
956 rest SFT models. We repeat random sampling for 100 times and report standard errors.

No. of Fitting-Validation Datapoints / Coefficient of determination (R^2)	Prediction based on SFT Pass@1 (avg. of 64) baseline	Prediction based on SFT Generalization Loss	Prediction based on SFT Pass@Large k (k=64)	Avg. Prediction from SFT Gen. Loss + Pass@Large k (64)
Fitting: 3; Validation: 7	0.32 \pm 0.39	0.73 \pm 0.41 (+0.41)	0.52 \pm 0.31	0.61 \pm 0.38
Fitting: 4; Validation: 6	0.27 \pm 0.36	0.75 \pm 0.34 (+0.48)	0.51 \pm 0.37	0.69 \pm 0.26
Fitting: 5; Validation: 5	0.29 \pm 0.38	0.79 \pm 0.26 (+0.50)	0.57 \pm 0.32	0.72 \pm 0.24
Fitting: 6; Validation: 4	0.37 \pm 0.37	0.78 \pm 0.25 (+0.41)	0.57 \pm 0.37	0.67 \pm 0.35
Fitting: 7; Validation: 3	0.36 \pm 0.36	0.77 \pm 0.30 (+0.41)	0.57 \pm 0.35	0.66 \pm 0.37
Fitting: 8; Validation: 2	0.31 \pm 0.46	0.68 \pm 0.36 (+0.37)	0.47 \pm 0.54	0.64 \pm 0.37

966 C.2 INSTANCE-LEVEL
967

968 Table 7 shows results on Llama3-8B-Instruct undergone SFT-RL with SFT examples from Llama-
969 Nemotron-SFT dataset and RLVR via GRPO on MATH dataset (train-split). Reporting Pass@1
970 performance averaged over 7 math benchmarks. Measuring prediction accuracy with coefficient of
971 determination (R^2) varying the ratio of fit-validation datapoints. We randomly select x SFT models

972 and fit a linear function between their post-SFT performance and performance after RL, and use it to
 973 predict for the rest SFT models. We repeat random sampling for 100 times and report standard errors.
 974

975 Table 8 shows results on Llama3-8B-Instruct undergone SFT-RL with SFT examples from Llama-
 976 Nemotron-SFT dataset and RLVR via GRPO on MATH dataset (train-split). Reporting Pass@1
 977 performance averaged over 7 math benchmarks. Spearman’s rank correlation between performance
 978 predicted from post-SFT models and the actual performance after RL, grouped by different SFT
 979 training budget.

980 Table 7: Llama3-8B-Instruct undergone SFT-RL with SFT examples from Llama-Nemotron-SFT
 981 dataset and RLVR via GRPO on MATH dataset (train-split). Reporting Pass@1 performance averaged
 982 over 7 math benchmarks. Measuring prediction accuracy with coefficient of determination (R^2)
 983 varying the ratio of fit-validation datapoints. We randomly select x SFT models and fit a linear
 984 function between their post-SFT performance and performance after RL, and use it to predict for the
 985 rest SFT models. We repeat random sampling for 100 times and report standard errors.

No. of Fitting-Validation Datapoints/ Coefficient of determination (R^2)	Prediction based on SFT Pass@1 (avg. of 64) baseline	Prediction based on SFT Pass@Large k (k=64)
Fitting: 3; Validation: 14	0.40 ± 0.31	0.89 ± 0.10 (+0.49)
Fitting: 4; Validation: 13	0.49 ± 0.30	0.89 ± 0.17 (+0.40)
Fitting: 5; Validation: 12	0.55 ± 0.22	0.91 ± 0.05 (+0.36)
Fitting: 6; Validation: 11	0.54 ± 0.30	0.92 ± 0.04 (+0.38)
Fitting: 7; Validation: 10	0.55 ± 0.24	0.92 ± 0.04 (+0.37)
Fitting: 8; Validation: 9	0.58 ± 0.20	0.92 ± 0.05 (+0.34)
Fitting: 10; Validation: 7	0.56 ± 0.25	0.92 ± 0.05 (+0.36)
Fitting: 12; Validation: 5	0.57 ± 0.28	0.92 ± 0.05 (+0.35)

996 Table 8: Llama3-8B-Instruct undergone SFT-RL with SFT examples from Llama-Nemotron-SFT
 997 dataset and RLVR via GRPO on MATH dataset (train-split). Reporting Pass@1 performance averaged
 998 over 7 math benchmarks. Spearman’s rank correlation between performance predicted from post-SFT
 999 models and the actual performance after RL, grouped by different SFT training budget.

SFT Compute Budget/ Spearman’s Rank Correlation	Prediction based on SFT Pass@1 (avg. of 64) baseline	Prediction based on SFT Pass@Large k (k=64)
Low Budget (< 2B tokens)	0.77	0.99 (+0.22)
Medium Budget (2 ~ 5B tokens)	0.60	0.90 (+0.30)
High Budget (5 ~ 20B tokens)	0.70	0.94 (+0.24)
Average	0.69	0.94 (+0.25)

1000 Table 9 shows results on Mistral-NeMo-12B-Instruct undergone SFT-RL with SFT examples from
 1001 AceReasoner1.1-SFT dataset and RLVR via GRPO on DeepScaleR dataset. Measuring prediction
 1002 accuracy with coefficient of determination (R^2) varying the ratio of fit-validation datapoints. Mistral-
 1003 NeMo-12B-Instruct undergone SFT-RL with SFT examples from AceReasoner1.1-SFT dataset and
 1004 RLVR via GRPO on DeepScaleR dataset. Reporting Pass@1 performance averaged over 7 math
 1005 benchmarks. We randomly select x SFT models and fit a linear function between their post-SFT
 1006 performance and performance after RL, and use it to predict for the rest SFT models. We repeat
 1007 random sampling for 100 times and report standard errors.

1008 Table 10 shows results on Mistral-NeMo-12B-Instruct undergone SFT-RL with SFT examples from
 1009 AceReasoner1.1-SFT dataset and RLVR via GRPO on DeepScaleR dataset. Reporting Pass@1
 1010 performance averaged over 7 math benchmarks. Spearman’s rank correlation between performance
 1011 predicted from post-SFT models and the actual performance after RL, grouped by different SFT
 1012 training budget.

1020
 1021
 1022
 1023
 1024
 1025

1026

1027

1028

1029

1030

1031

1032

1033 Table 9: Mistral-NeMo-12B-Instruct undergone SFT-RL with SFT examples from AceReasoner1.1-
 1034 SFT dataset and RLVR via GRPO on DeepScaleR dataset. Measuring prediction accuracy with
 1035 coefficient of determination (R^2) varying the ratio of fit-validation datapoints. Mistral-NeMo-12B-
 1036 Instruct undergone SFT-RL with SFT examples from AceReasoner1.1-SFT dataset and RLVR via
 1037 GRPO on DeepScaleR dataset. Reporting Pass@1 performance averaged over 7 math benchmarks.
 1038 We randomly select x SFT models and fit a linear function between their post-SFT performance and
 1039 performance after RL, and use it to predict for the rest SFT models. We repeat random sampling for
 1040 100 times and report standard errors.

1041

No. of Fitting-Validation Datapoints/ Coefficient of determination (R^2)	Prediction based on SFT Pass@1 (avg. of 64) baseline	Prediction based on SFT Pass@Large k (k=64)
Fitting: 2; Validation: 10	0.55 ± 0.42	0.87 ± 0.29 (+0.32)
Fitting: 3; Validation: 9	0.71 ± 0.15	0.94 ± 0.18 (+0.23)
Fitting: 4; Validation: 8	0.69 ± 0.22	0.98 ± 0.03 (+0.29)
Fitting: 5; Validation: 7	0.75 ± 0.10	0.98 ± 0.01 (+0.23)
Fitting: 6; Validation: 6	0.73 ± 0.16	0.98 ± 0.01 (+0.25)
Fitting: 8; Validation: 4	0.69 ± 0.35	0.97 ± 0.03 (+0.28)
Fitting: 10; Validation: 2	0.68 ± 0.42	0.91 ± 0.17 (+0.23)

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

Table 10: Mistral-NeMo-12B-Instruct undergone SFT-RL with SFT examples from AceReasoner1.1-
 SFT dataset and RLVR via GRPO on DeepScaleR dataset. Reporting Pass@1 performance averaged
 over 7 math benchmarks. Spearman’s rank correlation between performance predicted from post-SFT
 models and the actual performance after RL, grouped by different SFT training budget.

1064

SFT Compute Budget/ Spearman’s Rank Correlation	Prediction based on SFT Pass@1 (avg. of 64) baseline	Prediction based on SFT Pass@Large k (k=64)
Low Budget (< 2B tokens)	0.80	0.95 (+0.25)
Medium Budget (2 ~ 5B tokens)	0.80	1.00 (+0.20)
High Budget (5 ~ 20B tokens)	0.50	1.00 (+0.50)
Average	0.70	0.98 (+0.28)

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080 **D BROADER INVESTIGATIONS AND SUPPLEMENTARY EXAMPLES**
1081
1082
10831084 **D.1 FINE-GRAINED COMPARISONS: BEST SFT MODEL VS. BEST SFT+RL MODEL**
1085
10861087 Individual task performance largely correlates with the average performance, but the relative im-
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
18010
18011
18012
18013
18014
18015
18016
18017
18018
18019
18020
18021
18022
18023
18024
18025
18026
18027
18028
18029
18030
18031
18032
18033
18034
18035
18036
18037
18038
18039
18040
18041
18042
18043
18044
18045
18046
18047
18048
18049
18050
18051
18052
18053
18054
18055
18056
18057
18058
18059
18060
18061
18062
18063
18064
18065
18066
18067
18068
18069
18070
18071
18072
18073
18074
18075
18076
18077
18078
18079
18080
18081
18082
18083
18084
18085
18086
18087
18088
18089
18090
18091
18092
18093
18094
18095
18096
18097
18098
18099
180100
180101
180102
180103
180104
180105
180106
180107
180108
180109
180110
180111
180112
180113
180114
180115
180116
180117
180118
180119
180120
180121
180122
180123
180124
180125
180126
180127
180128
180129
180130
180131
180132
180133
180134
180135
180136
180137
180138
180139
180140
180141
180142
180143
180144
180145
180146
180147
180148
180149
180150
180151
180152
180153
180154
180155
180156
180157
180158
180159
180160
180161
180162
180163
180164
180165
180166
180167
180168
180169
180170
180171
180172
180173
180174
180175
180176
180177
180178
180179
180180
180181
180182
180183
180184
180185
180186
180187
180188
180189
180190
180191
180192
180193
180194
180195
180196
180197
180198
180199
180200
180201
180202
180203
180204
180205
180206
180207
180208
180209
180210
180211
180212
180213
180214
180215
180216
180217
180218
180219
180220
180221
180222
180223
180224
180225
180226
180227
180228
180229
180230
180231
180232
180233
180234
180235
180236
180237
180238
180239
180240
180241
180242
180243
180244
180245
180246
180247
180248
180249
180250
180251
180252
180253
180254
180255
180256
180257
180258
180259
180260
180261
180262
180263
180264
180265
180266
180267
180268
180269
180270
180271
180272
180273
180274
180275
180276
180277
180278
180279
180280
180281
180282
180283
180284
180285
180286
180287
180288
180289
180290
180291
180292
180293
180294
180295
180296
180297
180298
180299
180300
180301
180302
180303
180304
180305
180306
180307
180308
180309
180310
180311
180312
180313
180314
180315
180316
180317
180318
180319
180320
180321
180322
180323
180324
180325
180326
180327
180328
180329
180330
180331
180332
180333
180334
180335
180336
180337
180338
180339
180340
180341
180342
180343
180344
180345
180346
180347
180348
180349
180350
180351
180352
180353
180354
180355
180356
180357
180358
180359
180360
180361
180362
180363
180364
180365
180366
180367
180368
180369
180370
180371
180372
180373
180374
180375
180376
180377
180378
180379
180380
180381
180382
180383
180384
180385
180386
180387
180388
180389
180390
180391
180392
180393
180394
180395
180396
180397
180398
180399
180400
180401
180402
180403
180404
180405
180406
180407
180408
180409
180410
180411
180412
180413
180414
180415
180416
180417
180418
180419
180420
180421
180422
180423
180424
180425
180426
180427
180428
180429
180430
180431
180432
180433
180434
180435
180436
180437
180438
180439
180440
180441
180442
180443
180444
180445
180446
180447
180448
180449
180450
180451
180452
180453
180454
180455
180456
180457
180458
180459
180460
180461
180462
180463
180464
180465
180466
180467
180468
180469
180470
180471
180472
180473
180474
180475
180476
180477
180478
180479
180480
180481
180482
180483
180484
180485
180486
180487
180488
180489
180490
180491
180492
180493
180494
180495
180496
180497
180498
180499
180500
180501
180502
180503
180504
180505
180506
180507
180508
180509
180510
180511
180512
180513
180514
180515
180516
180517
180518
180519
180520
180521
180522
180523
180524
180525
180526
180527
180528
180529
180530
180531
180532
180533
180534
180535
180536
180537
180538
180539
180540
180541
180542
180543
180544
180545
180546
180547
180548
180549
180550
180551
180552
180553
180554
180555
180556
180557
180558
180559
180560
180561
180562
180563
180564
180565
180566
180567
180568
180569
180570
180571
180572
180573
180574
180575
180576
180577
180578
180579
180580
180581
180582
180583
180584
180585
180586
180587
180588
180589
180590
180591
180592
180593
180594
180595
180596
180597
180598
180599
180600
180601
180602
180603
180604
180605
180606
180607
180608
180609
180610
180611
180612
180613
180614
180615
180616
180617
180618
180619
180620
180621
180622
180623
180624
180625
180626
180627
180628
180629
180630
180631
180632
180633
180634
180635
180636
180637
180638
180639
180640
180641
180642
180643
180644
180645
180646
180647
180648
180649
180650
180651
180652
180653
180654
180655
180656
180657
180658
180659
180660
180661
180662
180663
180664
180665
180666
180667
180668
180669
180670
180671
180672
180673
180674
180675
180676
180677
180678
180679
180680
180681
180682
180683
180684
180685
180686
180687
180688
180689
180690
180691
180692
180693
180694
180695
180696
180697
180698
180699
180700
180701
180702
180703
180704
180705
180706
180707
180708
180709
180710
180711
180712
180713
180714
180715
180716
180717
180718
180719
180720
180721
180722
180723
180724
180725
180726
180727
180728
180729
180730
180731
180732
180733
180734
180735
180736
180737
180738
180739
180740
180741
180742
180743
180744
180745
180746
180747
180748
180749
180750
180751
180752
180753
180754
180755
180756
180757
180758
180759
180760
180761
180762
180763
180764
180765
180766
180767
180768
180769
180770
180771
180772
180773
180774
180775
180776
180777
180778
180779
180780
180781
180782
180783
180784
180785
180786
180787
180788
180789
180790
180791
180792
180793
180794
180795
180796
180797
180798
180799
180800
180801
180802
180803
180804
180805
180806
180807
180808
180809
180810
180811
180812
180813
180814
180815
180816
180817
180818
180819
180820
180821
180822
180823
180824
180825
180826
180827
180828
180829
180830
180831
180832
180833
180834
180835
180836
180837
180838
180839
180840
180841
180842
180843
180844
180845
180846
180847
180848
180849
180850
180851
180852
180853
180854
180855
180856
180857
180858
180859
180860
180861
180862
180863
180864
180865
180866
180867
180868
180869
180870
180871
180872
180873<br

1134 Table 13: Detailed comparison of post-SFT and SFT+RL performance (Pass@1 accuracy) between
 1135 the model with the best SFT performance (M2=Llama3-8B-Instruct with SFT on 10k Nemotron
 1136 examples for 10 epochs) and the model with the best SFT+RL performance (M1=Llama3-8B-Instruct
 1137 with SFT on 10k Nemotron examples for 5 epochs) from Figure 5(a) across various benchmarks.
 1138 *Despite M1 trailing M2 by 1.33% after SFT, in the final SFT+RL results, M1 leads M2 by a relative
 1139 margin of +7.31% with up to +76.52% on individual task.*

Model / Benchmark	Avg	MATH-500	AIME 1983-2024	GSM8k	AIME 2025	AMC	Olympiad	Minerva
<i>Supervised Fine-Tuning (SFT)</i>								
M1 SFT	26.63	53.12	2.75	84.59	3.18	21.71	7.75	13.37
M2 SFT	26.99	54.43	2.69	85.87	3.18	21.59	7.88	13.34
(M1-M2)/M2 (%)	-1.33	-2.41	+2.34	-1.49	0.00	+0.56	-1.59	+0.22
<i>Supervised Fine-Tuning + Reinforcement Learning (SFT+RL)</i>								
M1 SFT+RL	30.08	59.96	5.06	87.59	4.74	26.81	11.12	15.28
M2 SFT+RL	28.03	59.93	3.83	85.40	4.32	24.93	9.16	8.66
(M1-M2)/M2 (%)	+7.31	+0.05	+32.03	+2.56	+9.65	+7.54	+21.45	+76.52

1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1150 Table 14: Detailed comparison of post-SFT and SFT+RL performance (Pass@1 accuracy) between
 1151 the model with the best SFT performance (M2=Llama3-8B-Instruct with SFT on 25k Nemotron
 1152 examples for 8 epochs) and the model with the best SFT+RL performance (M1=Llama3-8B-Instruct
 1153 with SFT on 25k Nemotron examples for 4 epochs) from Figure 5(b) across various benchmarks.
 1154 *Despite M1 trailing M2 by 7.02% after SFT, in the final SFT+RL results, M1 leads M2 by a relative
 1155 margin of +4.82%, with up to +45.76% on individual task.*

Model / Benchmark	Avg	MATH-500	AIME 1983-2024	GSM8k	AIME 2025	AMC	Olympiad	Minerva
<i>Supervised Fine-Tuning (SFT)</i>								
M1 SFT	28.48	57.25	5.00	83.21	5.99	26.03	9.25	12.68
M2 SFT	30.63	62.12	6.78	84.21	8.23	30.71	10.00	12.37
(M1-M2)/M2 (%)	-7.02	-7.84	-26.26	-1.19	-27.22	-15.24	-7.50	+2.51
<i>Supervised Fine-Tuning + Reinforcement Learning (SFT+RL)</i>								
M1 SFT+RL	33.25	67.40	9.90	87.84	9.79	32.06	15.34	11.25
M2 SFT+RL	31.72	65.12	9.84	84.93	9.27	34.12	13.12	7.72
(M1-M2)/M2 (%)	+4.82	+3.50	+0.61	+3.43	+5.62	-6.04	+16.92	+45.76

1164
 1165
 1166
 1167
 1168
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187

1188 D.2 ABLATION STUDIES: PASS@K, ENTROPY
1189

1190 To facilitate pursuit towards underlying mechanism behind the intriguing SFT-RL dynamics, we
1191 investigate the aggregated entropy at the response level.

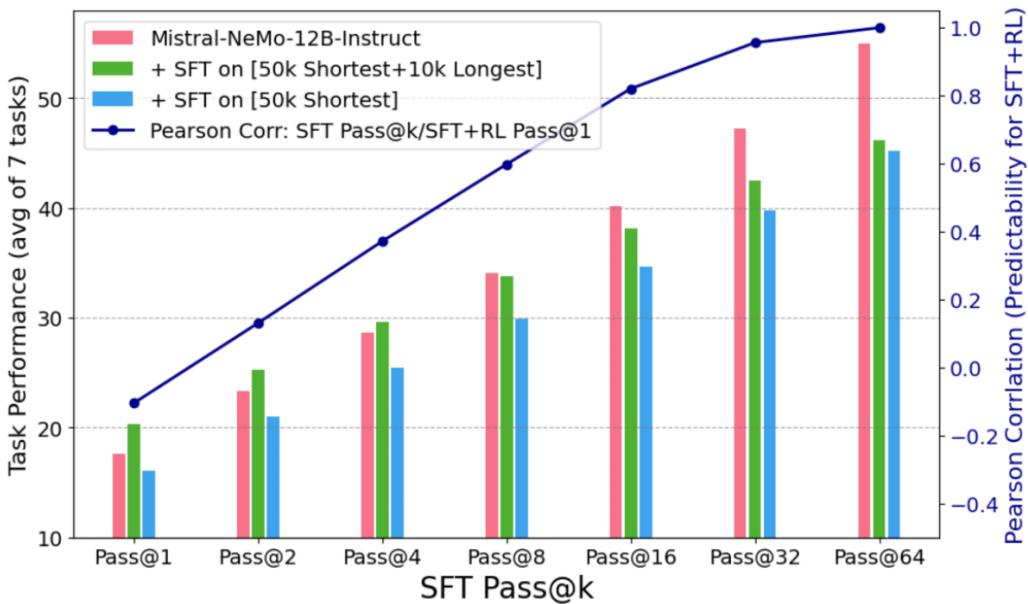
1192 As shown in Tables 15, 16, and 17, we witnessed that correct responses generally show a lower
1193 entropy compared to incorrect responses. However,

1194

- 1195 1. A lower entropy after SFT training does not always suggest worse post-RL outcomes, and
1196 vice versa.
- 1197 2. A lower entropy after SFT training does not indicate better post-SFT performance, nor does
1198 the entropy gap between correct and incorrect responses.

1200 One hypothesis is that the loss of tokens diverges. The loss is often dominated by a few high-loss
1201 tokens (Gadre et al., 2024). During SFT over-training, despite validation loss on some tokens
1202 elevating due to overfitting, the loss on some other tokens may still be decreasing, and these tokens
1203 may be more crucial for the task performance. On the other end, Wang et al. (2025) suggests that
1204 certain tokens are crucial for RL. These seem to be different tokens from those crucial for SFT.
1205 Overfitting/higher loss on these tokens could cause degradation in the subsequent RL. We found that
1206 during decoding, math/symbolic tokens generally show a near-zero loss—i.e., the model is almost
1207 always certain on these math tokens. The highest loss tokens are human languages that are naturally
1208 ambiguous or interchangeable, such as whether to start the sentence with "Thus" or "Therefore". As
1209 a result, the (per-token) entropy for the generations mostly measures *verbosity*, and does not show
1210 consistent patterns with accuracy.

1211 In general, we did not observe a consistent pattern to explain the SFT-RL dynamic, but it confirms
1212 that *entropy alone is an inadequate explanation*.



1233 Figure 11: Detailed comparison between Pass@k performance for Mistral-NeMo-12B-Instruct/
1234 Mistral-NeMo-12B-Instruct with SFT on 50k *shortest* and 10k *longest* AceReasoner1.1-SFT examples
1235 for 1 epoch/ Mistral-NeMo-12B-Instruct with SFT on 50k *shortest* AceReasoner1.1-SFT examples
1236 for 1 epoch under k=1,2,4,8,16,32,64. We show Pearson Correlation scores between Pass@k of
1237 these models and their Pass@1 performance after subsequent RL. Predicting post-RL performance
1238 from pre-RL Pass@1 leads to a negative Pearson Corr. score, which indicates ineffective predictions.
1239 Prediction accuracy (indicated by Pearson Corr.) steadily improves with larger k values in Pass@k,
1240 reaching over 0.80 at Pass@16 and close to 1 at Pass@64.

1241

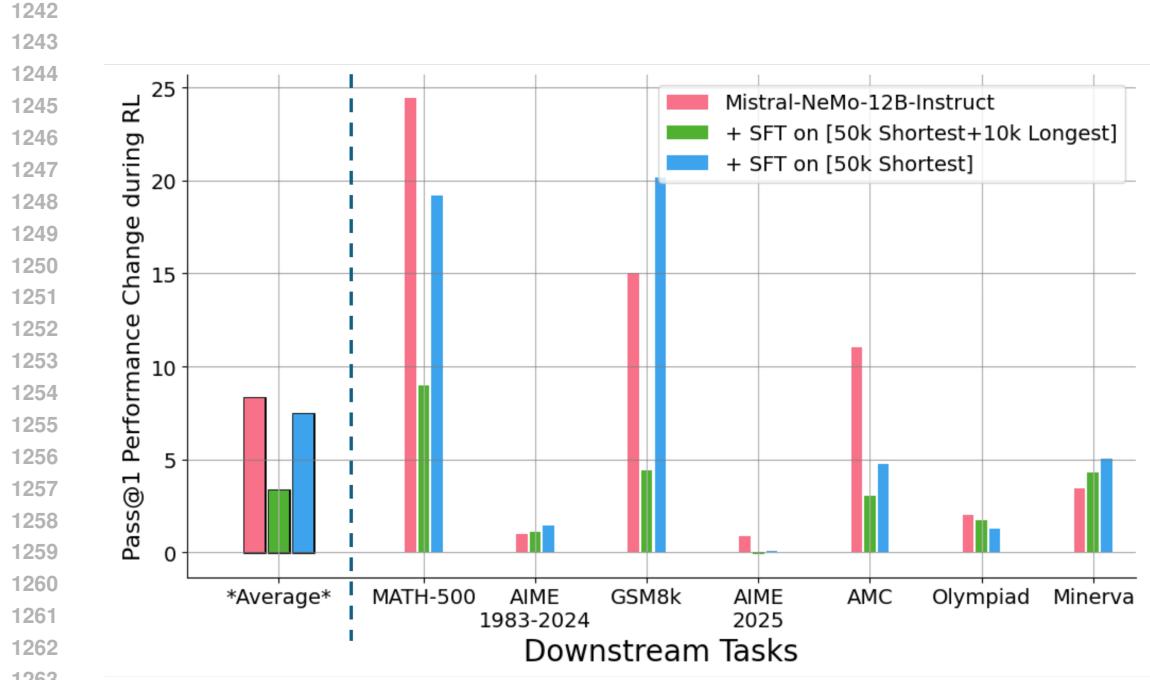


Figure 12: Improvements in per-task Pass@1 accuracies for [Mistral-NeMo-12B-Instruct](#) / [Mistral-NeMo-12B-Instruct with SFT on 50k shortest and 10k longest AceReasoner1.1-SFT examples for 1 epoch](#) / [Mistral-NeMo-12B-Instruct with SFT on 50k shortest AceReasoner1.1-SFT examples for 1 epoch](#) during RL training. Individual task performance largely correlates with the average performance, but the relative improvements differ across tasks. The average performance over multiple tasks has been reported to scale much more smoothly to the point of becoming predictable (Gadre et al., 2024). Thus, in this work, we use the average performance across multiple tasks as the main metric. The final results are consistent with these expectations.

Table 15: Per-task Pass@k accuracies and Entropy for [Mistral-NeMo-12B-Instruct](#) before and RL training. In comparisons between response-level entropy averaged over all responses vs. over correct responses, correct responses generally show a lower entropy compared to incorrect responses, but does not explain the dynamics between SFT-RL performance.

Task	Pass@1	Pass@2	Pass@4	Pass@8	Pass@16	Pass@32	Pass@64	Avg Ent.	Corr. Ent.
<i>base model</i>									
MATH-500	30.68	43.48	54.68	63.28	70.39	77.20	84.00	236.60	131.40
AIME 1983-2024	0.78	1.50	2.80	4.95	8.13	12.70	20.00	462.30	390.60
GSM8k	71.31	84.53	90.74	93.06	94.61	96.52	98.00	93.60	72.41
AIME 2025	0.26	0.52	1.04	2.08	4.17	8.33	16.66	488.60	513.30
AMC	10.78	17.30	25.68	36.02	48.07	59.98	70.00	370.30	220.20
Olympiad	3.22	5.82	10.00	16.40	25.54	36.67	48.00	434.50	222.60
Minerva	6.09	10.25	15.75	22.39	30.33	39.22	48.00	258.20	167.50
Average	17.59	23.34	28.67	34.03	40.18	47.23	54.95	334.87	245.43
<i>+RL</i>									
MATH-500	55.15	62.06	67.05	71.49	76.63	83.11	92.00	78.07	32.01
AIME 1983-2024	1.78	3.19	5.26	7.92	11.41	16.29	22.00	166.00	126.00
GSM8k	86.34	89.34	91.64	93.42	94.76	95.77	96.00	20.45	16.20
AIME 2025	1.15	2.09	3.54	5.46	7.92	11.69	16.66	185.40	80.77
AMC	21.84	28.67	36.06	43.17	49.84	56.51	64.00	117.60	42.68
Olympiad	5.25	8.59	12.71	17.55	23.32	29.31	34.00	145.60	74.91
Minerva	9.56	14.21	19.26	23.90	27.58	31.47	36.00	83.03	59.30
Average	25.87	29.74	33.65	37.56	41.64	46.31	51.52	113.74	61.70

1296
 1297
 1298
 1299 Table 16: Per-task Pass@k accuracies and Entropy for **Mistral-NeMo-12B-Instruct with SFT on 50k**
 1300 **shortest and 10k longest AceReasoner1.1-SFT examples for 1 epoch** before and RL training. In
 1301 comparisons between response-level entropy averaged over all responses vs. over correct responses,
 1302 correct responses generally show a lower entropy compared to incorrect responses, but does not
 1303 explain the dynamics between SFT-RL performance.

Task	Pass@1	Pass@2	Pass@4	Pass@8	Pass@16	Pass@32	Pass@64	Avg Ent.	Corr. Ent.
<i>SFT</i>									
MATH-500	42.78	53.65	61.63	68.30	74.65	80.14	84.00	2266.00	902.50
AIME 1983-2024	1.38	2.45	3.99	5.81	7.62	9.50	12.00	4733.00	1722.00
GSM8k	80.15	90.99	96.08	98.18	99.26	99.88	100.00	713.90	531.00
AIME 2025	0.94	1.62	2.48	3.14	3.33	3.33	3.33	4942.00	6888.00
AMC	8.75	14.47	22.67	32.89	43.99	54.70	62.00	4023.00	1987.00
Olympiad	3.69	6.01	9.15	13.19	18.02	23.75	30.00	5229.00	2718.00
Minerva	4.69	7.50	10.95	15.00	19.99	25.97	32.00	4019.00	2272.00
Average	20.34	25.24	29.56	33.79	38.12	42.47	46.19	3703.70	2431.50
<i>SFT+RL</i>									
MATH-500	51.78	60.98	67.51	72.91	77.49	82.02	86.00	1996.00	724.70
AIME 1983-2024	2.50	3.84	5.45	7.42	9.80	12.77	16.00	5041.00	1638.00
GSM8k	84.56	92.76	96.94	98.75	99.54	99.94	100.00	579.60	482.10
AIME 2025	0.83	1.54	2.64	4.06	5.58	7.51	10.00	5522.00	7202.00
AMC	11.81	18.88	28.53	40.07	51.81	62.35	72.00	3901.00	1799.00
Olympiad	5.41	8.45	12.27	17.02	22.44	28.51	36.00	6002.00	2595.00
Minerva	8.97	12.51	15.61	18.59	22.00	25.26	28.00	4861.00	1529.00
Average	23.69	28.42	32.71	36.97	41.24	45.48	49.71	3986.09	2281.40

1320
 1321
 1322
 1323
 1324
 1325
 1326 Table 17: Per-task Pass@k accuracies and Entropy for **Mistral-NeMo-12B-Instruct with SFT on 50k**
 1327 **shortest AceReasoner1.1-SFT examples for 1 epoch** before and RL training. In comparisons between
 1328 response-level entropy averaged over all responses vs. over correct responses, correct responses
 1329 generally show a lower entropy compared to incorrect responses, but does not explain the dynamics
 1330 between SFT-RL performance.

Task	Pass@1	Pass@2	Pass@4	Pass@8	Pass@16	Pass@32	Pass@64	Avg Ent.	Corr. Ent.
<i>SFT</i>									
MATH-500	32.56	44.08	53.95	61.75	68.55	75.20	80.00	771.60	452.70
AIME 1983-2024	1.25	2.02	2.91	3.88	5.17	6.97	10.00	1828.00	1136.00
GSM8k	65.46	78.12	85.99	91.59	95.11	96.98	98.00	670.10	474.60
AIME 2025	0.31	0.62	1.21	2.32	4.25	7.12	10.00	1963.00	2515.00
AMC	7.53	12.35	19.40	28.64	39.47	51.27	64.00	1232.00	856.70
Olympiad	2.59	4.49	7.22	10.91	15.82	21.88	28.00	2058.00	767.00
Minerva	3.00	5.01	7.52	10.30	13.84	19.02	26.00	2491.00	1448.00
Average	16.10	20.96	25.46	29.91	34.60	39.78	45.14	1573.39	1092.86
<i>SFT+RL</i>									
MATH-500	51.74	60.76	67.21	72.38	77.63	83.00	88.00	2924.00	1315.00
AIME 1983-2024	2.72	3.65	5.20	8.01	12.74	20.03	30.00	6171.00	1971.00
GSM8k	85.62	92.91	96.48	97.93	98.47	98.99	100.00	1220.00	916.60
AIME 2025	0.36	0.71	1.36	2.50	4.25	6.58	10.00	6234.00	12624.00
AMC	12.28	19.08	27.30	35.86	45.14	55.12	64.00	5072.00	2874.00
Olympiad	3.88	6.34	9.93	15.00	21.17	27.26	32.00	6080.00	2809.00
Minerva	8.06	10.87	13.41	16.28	19.98	24.03	28.00	5708.00	2568.00
Average	23.52	27.76	31.56	35.42	39.91	45.00	50.29	4772.71	3582.51

1350 E SAMPLE SFT EXAMPLES
13511352 E.1 SHORTEST EXAMPLES
13531354 A sample Shortest Examples from Llama-Nemotron Post-Training Dataset (Singhal et al.) (~750
1355 tokens):1356 [Input]:
13571358 Solve the following math problem. Make sure to put the answer (and
1359 only answer) inside \boxed{ }. \n\nThe probability of snow on
1360 1st January is 1/20. What is the probability that snow will
1361 fall on the next 1st January?"1362 [Output]:
13631364 <think> Okay, so I have this problem here: "The probability of
1365 snow on 1st January is 1/20. What is the probability that snow
1366 will fall on the next 1st January?" Hmm, let's see. Alright,
1367 the question is asking for the probability that it will snow
1368 next January 1st, given that the probability is 1/20. Wait, is
1369 there more to this? Let me check again. Wait, maybe I
1370 misread. The original probability is 1/20 for snow on January
1371 1st, and they want the probability it will snow on the next
1372 January 1st. Hmm. That seems straightforward. But maybe there's
1373 a trick here? Like, are they assuming each year is
1374 independent? Because if the probability each year is
1375 independent, then it's just 1/20 again. But maybe the question
1376 is different? Wait, maybe the wording is tricky. Let me parse
1377 it again. "The probability of snow on 1st January is 1/20.
1378 What is the probability that snow will fall on the next 1st
1379 January?" So, "next" as in the following year? So next year's
1380 January 1st. So if each year's snowfall is independent, then
1381 the probability remains 1/20. But maybe there's some hidden
1382 assumption here. Wait, sometimes in probability problems, they
1383 might be talking about conditional probabilities or something
1384 else. But this problem doesn't mention any previous events or
1385 dependencies. It just states the probability is 1/20. So if
1386 each occurrence is independent, then the probability doesn't
1387 change. So next year's probability is also 1/20. Wait, but
1388 let me think again. Maybe the problem is in another language
1389 originally, and "next" is being translated as "the next
1390 occurrence" or something? But no, it's about "next 1st January
1391 ". So like, the next time January 1st comes around, which
1392 would be the following year. So in terms of probability,
1393 assuming each year is an independent event, the probability is
1394 the same each year, right? So if the probability of snow on
1395 any given January 1st is 1/20, then regardless of previous
1396 years, the next one would also be 1/20. Alternatively, could
1397 it be that the problem is asking for the probability that it
1398 will snow next year given that it snowed this year? But the
1399 problem doesn't say anything about prior snowfall. It just
1400 gives the base rate. So I think the answer is 1/20. Let me
1401 confirm. Another angle: sometimes people confuse "probability
1402 of snow on a specific day" with "probability of snow in a
1403 year" but here it's specifically on 1st January each year. So
1404 if the probability each year on that date is 1/20, then next
1405 year's is also 1/20. Unless there's a pattern, but the problem
1406 doesn't indicate that. So I think the answer is 1/20. So I
1407 just need to write that in a box. Wait, maybe I need to make

1404 sure if the problem is in a different context. For example,
1405 maybe it's a different place where the next January 1st is in
1406 the southern hemisphere? But that's overcomplicating. The
1407 problem is likely testing understanding of independent events.
1408 So the answer is 1/20. </think> \boxed{\frac{1}{20}}

1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457

1458 A sample Shortest Examples from AceReasoner1.1-SFT Dataset (Liu et al., 2025b) (~700 tokens):
 1459
 1460 [Input]:
 1461 A train 18 m long passes a pole in 9 seconds. How long will it
 1462 take to pass a bridge 36 m long?
 1463
 1464 [Output]:
 1465 <think> Okay, so I need to figure out how long it takes for a
 1466 train that's 18 meters long to pass a bridge that's 36 meters
 1467 long. The problem already says that it takes 9 seconds for the
 1468 train to pass a pole. Let me start by understanding what that
 1469 means. When the train passes a pole, the entire length of
 1470 the train has to go by the pole. So, if the train is 18 meters
 1471 long and it takes 9 seconds to pass the pole, that means the
 1472 front of the train passes the pole, and then 9 seconds later,
 1473 the end of the train passes the pole. So, in 9 seconds, the
 1474 train travels a distance equal to its own length, which is 18
 1475 meters. From this, I can calculate the speed of the train.
 1476 Speed is distance divided by time. So, speed = 18 meters / 9
 1477 seconds. Let me compute that. 18 divided by 9 is 2. So, the
 1478 train's speed is 2 meters per second. Hmm, that seems right.
 1479 Now, the question is about passing a bridge that's 36 meters
 1480 long. To pass a bridge, the train has to cover the entire
 1481 length of the bridge plus its own length. Why? Because first
 1482 the front of the train has to cross the entire bridge, and
 1483 then the end of the train has to exit the bridge. So, the
 1484 total distance the train needs to cover is the length of the
 1485 bridge plus the length of the train. So that would be 36
 1486 meters plus 18 meters, which is 54 meters. So, the train
 1487 needs to travel 54 meters to completely pass the bridge. Since
 1488 the speed is 2 meters per second, the time it takes is the
 1489 total distance divided by speed. Time = 54 meters / 2 meters
 1490 per second. Let me calculate that. 54 divided by 2 is 27. So,
 1491 it should take 27 seconds. Wait, let me double-check to make
 1492 sure I didn't mix up anything. The first part: passing a pole
 1493 takes 9 seconds for 18 meters. So speed is $18/9 = 2 \text{ m/s}$. Then,
 1494 bridge is 36 meters. So, when passing the bridge, the front
 1495 of the train has to go 36 meters to get to the end of the
 1496 bridge, and then the entire train has to clear it, which is
 1497 another 18 meters. So total distance is indeed $36 + 18 = 54$
 1498 meters. Time is $54 / 2 = 27$ seconds. Yeah, that makes sense.
 1499 Alternatively, maybe someone might think you just need to
 1500 cover the bridge's length, but no, the train's own length has
 1501 to be considered too. Because until the last part of the train
 1502 is off the bridge, the bridge isn't completely passed. So,
 1503 adding the two lengths is correct. Therefore, 27 seconds
 1504 should be the right answer. </think> The train's speed is $\frac{18 \text{ m}}{9 \text{ s}} = 2 \text{ m/s}$. To pass a
 1505 36 m bridge, the total distance needed is $36 \text{ m} + 18 \text{ m} = 54 \text{ m}$. Time required is $\frac{54 \text{ m}}{2 \text{ m/s}} = 27 \text{ seconds}$. $\boxed{27}$
 1506
 1507
 1508
 1509
 1510
 1511

1512 E.2 LONGEST EXAMPLES
15131514 A sample Longest Examples from AceReasoner1.1-SFT Dataset (Liu et al., 2025b) (~8k tokens):
1515

1516 [Input]:

1517 For sets $\$A\$$, $\$B\$$, and $\$C\$$, where $\$n(A)\$$, $\$n(B)\$$, and $\$n(C)\$$
1518 denote the number of subsets of $\$A\$$, $\$B\$$, and $\$C\$$
1519 respectively, and $\|\{A\}\| = \|\{B\}\| = \|\{C\}\| = 100$, if $\$n(A) + n(B) + n(C) = n(A \cap B \cap C)\$$,
1520 what is the minimum possible value of $\|\{A \cap B \cap C\}\|$?
1521
1522

1523

1524 [Output]:

1525 Okay, let's try to tackle this problem. Hmm, so we have three sets
1526 A , B , and C . The problem states that $n(A)$, $n(B)$, and $n(C)$ are
1527 the number of subsets of each set. Wait, but usually, the
1528 number of subsets of a set with size k is 2^k . So maybe $n(A)$
1529 is $2^{|A|}$, $n(B)$ is $2^{|B|}$, and similarly for $n(C)$. Let me check:
1530 the problem says " $n(A)$, $n(B)$, and $n(C)$ denote the number of
1531 subsets of A , B , and C respectively." Yep, so that means $n(A) = 2^{|A|}$,
1532 $n(B) = 2^{|B|}$, $n(C) = 2^{|C|}$.
15331534 Given that $|A| = |B| = 100$. So $n(A) = 2^{100}$, $n(B) = 2^{100}$. The
1535 equation given is $n(A) + n(B) + n(C) = n(A \cap B \cap C)$. So
1536 that translates to $2^{100} + 2^{100} + 2^{|C|} = 2^{|A \cap B \cap C|}$.
15371538 Wait, the right-hand side is the number of subsets of the
1539 intersection of A , B , and C . The number of subsets of a set is
1540 $2^{\{\text{cardinality of the set}\}}$, so $n(A \cap B \cap C) = 2^{|A \cap B \cap C|}$.
15411542 So the equation is $2^{100} + 2^{100} + 2^{|C|} = 2^{|A \cap B \cap C|}$, where k is $|A \cap B \cap C|$, and we need to find the minimal possible k .
15431544 Let me simplify the left-hand side. $2^{100} + 2^{100}$ is $2*2^{100} = 2^{101}$. So left-hand side is $2^{101} + 2^{|C|}$. So $2^{101} + 2^{|C|} = 2^k$.
15451546 We need to find the smallest possible k such that this equation
1547 holds, given that $|C|$ can be adjusted (I think we can choose $|C|$
1548 as part of finding the minimum? Wait, no. The problem is
1549 about the minimal $|A \cap B \cap C|$, given that the equation
1550 holds. So we need to find the minimal k such that $2^{101} + 2^{|C|} = 2^k$, and also considering the relationships between the
1551 sets A , B , C .
15521553 Wait, but how is $|C|$ related to A and B ? Because A , B , and C are
1554 just sets, but the problem doesn't specify any constraints on
1555 their relationships except through their intersection $A \cap B \cap C$. Hmm. Wait, but the number of subsets of C , which is
1556 $2^{|C|}$, is part of the equation. But also, the intersection $A \cap B \cap C$ is a subset of C , right? Because the intersection
1557 of A , B , and C is a subset of each of them. So $|A \cap B \cap C| \leq |C|$. So the cardinality of the intersection can't be
1558 more than the cardinality of C . But in our equation, $2^{|k|}$ is equal to $2^{101} + 2^{|C|}$, so $2^{|k|} = 2^{|C|} + 2^{101}$.
1559

1566 So we need to find integers $|C|$ and k such that $k \leq |C|$ (since
 1567 the intersection can't be larger than C), and $2^{|k|} = 2^{|101|}$
 1568 + $2^{|C|}$. Also, k must be an integer because it's the
 1569 cardinality of a set.

1570 Wait, but how can we solve this equation? Let me think. Let's
 1571 denote $m = |C|$ and $k = |A \cap B \cap C|$, so we have $2^k =$
 1572 $2^{|101|} + 2^m$. We need to find the minimal k such that there
 1573 exists m with $m \leq k$ and $2^k = 2^{|101|} + 2^m$.

1575 But wait, if m is greater than or equal to k , then 2^m is a
 1576 multiple of 2^k . But $2^k = 2^{|101|} + 2^m$. If $m > 101$, then 2^m
 1577 is larger than $2^{|101|}$, so $2^{|101|} + 2^m = 2^m (1 + 2^{|101| - m})$. For this to be equal to 2^k , $1 + 2^{|101| - m}$ must be a
 1578 power of two. Let's write:

1580
 1581 Let $m \leq k$, and $2^k = 2^{|101|} + 2^m$. Let's consider cases where m
 1582 > 101 and $m \leq 101$.

1583
 1584 Case 1: $m > 101$. Then $2^k = 2^{|101|} + 2^m = 2^{|101|}(1 + 2^{|m - 101|})$. For this to be a power of two, $1 + 2^{|m - 101|}$ must be
 1585 a power of two. Let's set $t = m - 101$, which is positive
 1586 since $m > 101$. Then $1 + 2^t$ must be a power of two. $1 + 2^t =$
 1587 2^s , for some s . So $2^s - 2^t = 1$. This is possible only if t
 1588 = 0, since $2^s - 1 = 2^t$, but t must be positive here (since m
 1589 > 101 implies $t \geq 1$). But $2^s - 2^t = 1$. If $t \geq 1$, 2^t
 1590 divides the left-hand side, so 2^t must divide 1, which is
 1591 only possible if $t=0$, which contradicts $t \geq 1$. Therefore,
 1592 there is no solution in this case.

1593
 1594 Case 2: $m \leq 101$. Then $2^k = 2^{|101|} + 2^m$. Let's factor out 2^m :
 1595 $2^k = 2^m (1 + 2^{|101| - m})$. Since $m \leq 101$, $101 - m \geq 0$, so
 1596 the exponent is non-negative. For the right-hand side to be a
 1597 power of two, $(1 + 2^{|101| - m})$ must be a power of two.
 1598 Let's denote $t = 101 - m$. Then $1 + 2^t$ must be a power of
 1599 two. So, $1 + 2^t = 2^s$. This equation holds when t is such
 1600 that $2^t + 1$ is a power of two.

1601
 1602 Looking at this, $2^t + 1 = 2^s$. The only solutions for this in
 1603 integers are when $t=0$, then $2^0 + 1 = 2 = 2^1$, so $s=1$. Another
 1604 possibility might be $t=1$: $2+1=3$, which is not a power of two.
 1605 $t=2:4+1=5$; nope. $t=3:8+1=9$; still not. So the only solution is
 1606 $t=0$, leading to $1+1=2$. So $t=0 \Rightarrow 1 + 2^0 = 2$. Therefore, $t=0$,
 1607 so $101 - m = 0 \Rightarrow m = 101$. Then $1 + 2^{|0|} = 2$, so the equation
 1608 becomes $2^k = 2^{101} + 2^{101} = 2^{101} * 2 = 2^{102}$. Therefore, k
 1609 = 102.

1610
 1611 Wait, but m is 101, and k must be m , but here $k=102$, which
 1612 would be greater than $m=101$. But that contradicts the
 1613 requirement that $k \leq m$. So this is impossible.

1614
 1615 Therefore, in case 2, even though we have $t=0$ leading to $m=101$ and
 1616 $k=102$, which would require $k=102 > m=101$, violating the
 1617 cardinality condition. Therefore, this is not acceptable.

1618
 1619 Hmm, so neither case 1 nor case 2 gives us a valid solution. Wait,
 1620 that can't be. Maybe there's an error in my reasoning here.
 Let me check again.

1620 Wait, when we considered $m = 101$ and $t=101 -m$. So if $t=0$, then $m = 101$, then the equation becomes $2^k = 2^{101} + 2^{101} = 2^{102}$,
 1621 so $k=102$. But since $m=101$, then the intersection k must be
 1622 101, but here $k=102$, which is greater. So that's impossible
 1623 . So that doesn't work. Therefore, in case 2, there are no
 1624 solutions either.

1625
 1626 Hmm, so does that mean there is no solution? But the problem
 1627 states "what is the minimum possible value of $|A \cap B \cap C|$ ", implying that there is a solution. So perhaps my approach
 1628 is missing something.

1629
 1630 Wait, maybe I need to consider that C can be a subset of $A \cap B$
 1631 C ? No, actually, $A \cap B \cap C$ is a subset of C , so the
 1632 size can't exceed $|C|$. So perhaps I need to adjust the
 1633 equation? Or maybe my initial translation of the problem is
 1634 wrong.

1635
 1636 Wait, let me go back to the problem statement. It says: For sets A ,
 1637 B , and C , where $n(A)$, $n(B)$, and $n(C)$ denote the number of
 1638 subsets of A , B , and C respectively, and $|A| = |B| = 100$, if n
 1639 $(A) + n(B) + n(C) = n(A \cap B \cap C)$, what is the minimum
 1640 possible value of $|A \cap B \cap C|$?

1641
 1642 Wait, so $n(A \cap B \cap C)$ is the number of subsets of $A \cap B$
 1643 C . So that is $2^{|A \cap B \cap C|}$. Similarly, $n(A) = 2^{|A|}$
 1644 $= 2^{100}$, $n(B) = 2^{100}$, $n(C) = 2^{|C|}$. So $2^{100} + 2^{100} + 2^{|C|} = 2^{|A \cap B \cap C|}$.

1645
 1646 So yes, as before, $2^{101} + 2^{|C|} = 2^k$, where k is $|A \cap B \cap C|$. We need to find the minimal k such that this holds, with $|C| \leq k$. So $k \geq |C|$.

1647
 1648 Wait, perhaps $|C|$ can be larger than k . Wait, but we need to find
 1649 the minimal k . So to minimize k , we need to make the right-
 1650 hand side as small as possible, so 2^k must be just enough
 1651 to hold $2^{101} + 2^{|C|}$. But $|C|$ can be as large as needed?
 1652 Wait, but $|C|$ is part of the equation. If $|C|$ is large, then
 1653 $2^{|C|}$ is very big, which would require k to be large as well.
 1654 So maybe to minimize k , we should take $|C|$ as small as
 1655 possible. But $|C|$ can't be smaller than k , since $k = |A \cap B \cap C| \geq |C|$.

1656
 1657 Wait, so maybe we need to take $|C| = k$. Then the equation becomes
 1658 $2^{101} + 2^k = 2^k$, which would imply $2^{101} = 0$, which
 1659 is impossible. So $|C|$ must be greater than k ? Wait, but if $|C|$
 1660 is greater than k , then $2^{|C|}$ is bigger than 2^k , so
 1661 $2^{101} + 2^{|C|}$ is bigger than 2^k . Therefore, the
 1662 equation $2^{101} + 2^{|C|} = 2^k$ implies that 2^k must be
 1663 larger than 2^{101} and $2^{|C|}$. So k must be greater than
 1664 both 101 and $|C|$. But $|C|$ can be as small as k , but then k
 1665 must be greater than $|C|$. That seems conflicting. Wait, no, if
 1666 $|C|$ is equal to k , then $2^k + 2^{101} = 2^k$, which is
 1667 impossible. If $|C|$ is less than k , then $2^{|C|}$ is less than
 1668 2^k , so $2^{101} + \text{something smaller than } 2^k$ equals 2^k ,
 1669 which would mean 2^{101} is approximately 2^k , so $k \approx 101$,
 1670 but since $2^{101} + 2^{|C|} = 2^k$, then 2^k must be bigger
 1671 than 2^{101} . So $k > 101$.

1674 Wait, maybe I should approach this differently. Let's suppose that
 1675 $|C|$ is as small as possible. Since the intersection $A \cap B \cap C$ is a subset of C , the minimal possible $|C|$ is at least $|A \cap B \cap C|$. Let's denote $k = |A \cap B \cap C|$. So $|C| \geq k$.
 1676 .
 1677 .
 1678 .

1679 We need to have $2^{100} + 2^{100} + 2^{|C|} = 2^k$. So $2^{101} + 2^{|C|} = 2^k$.
 1680 .
 1681 .

1682 To minimize k , we want to make 2^k as small as possible. Since $|C| \geq k$, $2^{|C|} \geq 2^k$. Therefore, $2^{101} + 2^{|C|} \geq 2^{101} + 2^k$. Wait, but that's not helpful. Wait, perhaps the minimal k is such that 2^k is just greater than or equal to $2^{101} + 2^{|C|}$. But $|C|$ is variable here. So perhaps we can choose $|C|$ such that $2^{|C|}$ is equal to $2^k - 2^{101}$, but since $|C|$ must be at least k , $2^k - 2^{101}$ must be a power of two, and $|C| = \log_2(2^k - 2^{101})$.
 1683 .
 1684 .
 1685 .
 1686 .
 1687 .
 1688 .
 1689 .
 1690 .

1691 But $2^k - 2^{101}$ must equal $2^{|C|}$, which is a power of two.
 1692 Let's set $x = 2^{101}$, so $2^k - x = 2^m$, where $m = |C|$ and $m \geq k$. Wait, but $m \geq k$ implies that $2^m \geq 2^k$, so $2^k - x = 2^m \geq 2^k$, which would imply that $-x \geq 0$, but x is positive, which is impossible. So that can't happen. Therefore, the previous conclusion that there are no solutions may be correct, but that conflicts with the problem statement. So where is the mistake?
 1693 .
 1694 .
 1695 .
 1696 .
 1697 .
 1698 .

1699 Alternatively, maybe my assumption that $|C|$ must be at least k is
 1700 wrong. Wait, but the intersection $A \cap B \cap C$ is a subset of C , so its size can't exceed $|C|$. So $|A \cap B \cap C| \leq |C|$.
 1701 .
 1702 .

1703 Therefore, $2^k = 2^{101} + 2^{|C|}$, and $k \geq |C|$. Let me
 1704 rephrase this as $2^k - 2^{|C|} = 2^{101}$. But since $k \geq |C|$, then $2^k \geq 2^{|C|}$, so $2^k - 2^{|C|} \geq 0$. But the
 1705 right-hand side is 2^{101} , which is positive. Therefore, this
 1706 equation can't be satisfied. Wait, that suggests that there
 1707 is no solution. But the problem says "if $n(A) + n(B) + n(C) =$
 1708 $n(A \cap B \cap C)$ ", so it's assuming that such sets exist.
 1709 .
 1710 .
 1711 .
 1712 .
 1713 .

1714 Wait, maybe I made an error in translating the problem. Let me
 1715 check again.
 1716 .
 1717 .
 1718 .

1719 The problem says: For sets A , B , and C , where $n(A)$, $n(B)$, and $n(C)$
 1720 denote the number of subsets of A , B , and C respectively, and
 1721 $|A| = |B| = 100$, if $n(A) + n(B) + n(C) = n(A \cap B \cap C)$,
 1722 what is the minimum possible value of $|A \cap B \cap C|$?
 1723 .
 1724 .
 1725 .
 1726 .
 1727 .

1728 Wait, maybe " $n(A)$ " is not $2^{|A|}$ but the number of subsets, which
 1729 is indeed $2^{|A|}$. So the equation is $2^{100} + 2^{100} + 2^{|C|} = 2^{|A \cap B \cap C|}$.
 1730 .
 1731 .
 1732 .

1733 So $2^{101} + 2^{|C|} = 2^k$, where $k = |A \cap B \cap C|$. So 2^k
 1734 must be equal to $2^{101} + 2^m$, where $m = |C|$, and $k \geq m$.
 1735 .
 1736 .
 1737 .
 1738 .

1739 But as per earlier, this leads to $2^k = 2^{101} + 2^m$, with $k \geq m$.
 1740 .
 1741 .
 1742 .
 1743 .
 1744 .
 1745 .
 1746 .
 1747 .
 1748 .
 1749 .
 1750 .
 1751 .
 1752 .
 1753 .
 1754 .
 1755 .
 1756 .
 1757 .
 1758 .
 1759 .
 1760 .
 1761 .
 1762 .
 1763 .
 1764 .
 1765 .
 1766 .
 1767 .
 1768 .
 1769 .
 1770 .
 1771 .
 1772 .
 1773 .
 1774 .
 1775 .
 1776 .
 1777 .
 1778 .
 1779 .
 1780 .
 1781 .
 1782 .
 1783 .
 1784 .
 1785 .
 1786 .
 1787 .
 1788 .
 1789 .
 1790 .
 1791 .
 1792 .
 1793 .
 1794 .
 1795 .
 1796 .
 1797 .
 1798 .
 1799 .
 1800 .
 1801 .
 1802 .
 1803 .
 1804 .
 1805 .
 1806 .
 1807 .
 1808 .
 1809 .
 1810 .
 1811 .
 1812 .
 1813 .
 1814 .
 1815 .
 1816 .
 1817 .
 1818 .
 1819 .
 1820 .
 1821 .
 1822 .
 1823 .
 1824 .
 1825 .
 1826 .
 1827 .
 1828 .
 1829 .
 1830 .
 1831 .
 1832 .
 1833 .
 1834 .
 1835 .
 1836 .
 1837 .
 1838 .
 1839 .
 1840 .
 1841 .
 1842 .
 1843 .
 1844 .
 1845 .
 1846 .
 1847 .
 1848 .
 1849 .
 1850 .
 1851 .
 1852 .
 1853 .
 1854 .
 1855 .
 1856 .
 1857 .
 1858 .
 1859 .
 1860 .
 1861 .
 1862 .
 1863 .
 1864 .
 1865 .
 1866 .
 1867 .
 1868 .
 1869 .
 1870 .
 1871 .
 1872 .
 1873 .
 1874 .
 1875 .
 1876 .
 1877 .
 1878 .
 1879 .
 1880 .
 1881 .
 1882 .
 1883 .
 1884 .
 1885 .
 1886 .
 1887 .
 1888 .
 1889 .
 1890 .
 1891 .
 1892 .
 1893 .
 1894 .
 1895 .
 1896 .
 1897 .
 1898 .
 1899 .
 1900 .
 1901 .
 1902 .
 1903 .
 1904 .
 1905 .
 1906 .
 1907 .
 1908 .
 1909 .
 1910 .
 1911 .
 1912 .
 1913 .
 1914 .
 1915 .
 1916 .
 1917 .
 1918 .
 1919 .
 1920 .
 1921 .
 1922 .
 1923 .
 1924 .
 1925 .
 1926 .
 1927 .
 1928 .
 1929 .
 1930 .
 1931 .
 1932 .
 1933 .
 1934 .
 1935 .
 1936 .
 1937 .
 1938 .
 1939 .
 1940 .
 1941 .
 1942 .
 1943 .
 1944 .
 1945 .
 1946 .
 1947 .
 1948 .
 1949 .
 1950 .
 1951 .
 1952 .
 1953 .
 1954 .
 1955 .
 1956 .
 1957 .
 1958 .
 1959 .
 1960 .
 1961 .
 1962 .
 1963 .
 1964 .
 1965 .
 1966 .
 1967 .
 1968 .
 1969 .
 1970 .
 1971 .
 1972 .
 1973 .
 1974 .
 1975 .
 1976 .
 1977 .
 1978 .
 1979 .
 1980 .
 1981 .
 1982 .
 1983 .
 1984 .
 1985 .
 1986 .
 1987 .
 1988 .
 1989 .
 1990 .
 1991 .
 1992 .
 1993 .
 1994 .
 1995 .
 1996 .
 1997 .
 1998 .
 1999 .
 2000 .
 2001 .
 2002 .
 2003 .
 2004 .
 2005 .
 2006 .
 2007 .
 2008 .
 2009 .
 2010 .
 2011 .
 2012 .
 2013 .
 2014 .
 2015 .
 2016 .
 2017 .
 2018 .
 2019 .
 2020 .
 2021 .
 2022 .
 2023 .
 2024 .
 2025 .
 2026 .
 2027 .
 2028 .
 2029 .
 2030 .
 2031 .
 2032 .
 2033 .
 2034 .
 2035 .
 2036 .
 2037 .
 2038 .
 2039 .
 2040 .
 2041 .
 2042 .
 2043 .
 2044 .
 2045 .
 2046 .
 2047 .
 2048 .
 2049 .
 2050 .
 2051 .
 2052 .
 2053 .
 2054 .
 2055 .
 2056 .
 2057 .
 2058 .
 2059 .
 2060 .
 2061 .
 2062 .
 2063 .
 2064 .
 2065 .
 2066 .
 2067 .
 2068 .
 2069 .
 2070 .
 2071 .
 2072 .
 2073 .
 2074 .
 2075 .
 2076 .
 2077 .
 2078 .
 2079 .
 2080 .
 2081 .
 2082 .
 2083 .
 2084 .
 2085 .
 2086 .
 2087 .
 2088 .
 2089 .
 2090 .
 2091 .
 2092 .
 2093 .
 2094 .
 2095 .
 2096 .
 2097 .
 2098 .
 2099 .
 2100 .
 2101 .
 2102 .
 2103 .
 2104 .
 2105 .
 2106 .
 2107 .
 2108 .
 2109 .
 2110 .
 2111 .
 2112 .
 2113 .
 2114 .
 2115 .
 2116 .
 2117 .
 2118 .
 2119 .
 2120 .
 2121 .
 2122 .
 2123 .
 2124 .
 2125 .
 2126 .
 2127 .
 2128 .
 2129 .
 2130 .
 2131 .
 2132 .
 2133 .
 2134 .
 2135 .
 2136 .
 2137 .
 2138 .
 2139 .
 2140 .
 2141 .
 2142 .
 2143 .
 2144 .
 2145 .
 2146 .
 2147 .
 2148 .
 2149 .
 2150 .
 2151 .
 2152 .
 2153 .
 2154 .
 2155 .
 2156 .
 2157 .
 2158 .
 2159 .
 2160 .
 2161 .
 2162 .
 2163 .
 2164 .
 2165 .
 2166 .
 2167 .
 2168 .
 2169 .
 2170 .
 2171 .
 2172 .
 2173 .
 2174 .
 2175 .
 2176 .
 2177 .
 2178 .
 2179 .
 2180 .
 2181 .
 2182 .
 2183 .
 2184 .
 2185 .
 2186 .
 2187 .
 2188 .
 2189 .
 2190 .
 2191 .
 2192 .
 2193 .
 2194 .
 2195 .
 2196 .
 2197 .
 2198 .
 2199 .
 2200 .
 2201 .
 2202 .
 2203 .
 2204 .
 2205 .
 2206 .
 2207 .
 2208 .
 2209 .
 2210 .
 2211 .
 2212 .
 2213 .
 2214 .
 2215 .
 2216 .
 2217 .
 2218 .
 2219 .
 2220 .
 2221 .
 2222 .
 2223 .
 2224 .
 2225 .
 2226 .
 2227 .
 2228 .
 2229 .
 2230 .
 2231 .
 2232 .
 2233 .
 2234 .
 2235 .
 2236 .
 2237 .
 2238 .
 2239 .
 2240 .
 2241 .
 2242 .
 2243 .
 2244 .
 2245 .
 2246 .
 2247 .
 2248 .
 2249 .
 2250 .
 2251 .
 2252 .
 2253 .
 2254 .
 2255 .
 2256 .
 2257 .
 2258 .
 2259 .
 2260 .
 2261 .
 2262 .
 2263 .
 2264 .
 2265 .
 2266 .
 2267 .
 2268 .
 2269 .
 2270 .
 2271 .
 2272 .
 2273 .
 2274 .
 2275 .
 2276 .
 2277 .
 2278 .
 2279 .
 2280 .
 2281 .
 2282 .
 2283 .
 2284 .
 2285 .
 2286 .
 2287 .
 2288 .
 2289 .
 2290 .
 2291 .
 2292 .
 2293 .
 2294 .
 2295 .
 2296 .
 2297 .
 2298 .
 2299 .
 2300 .
 2301 .
 2302 .
 2303 .
 2304 .
 2305 .
 2306 .
 2307 .
 2308 .
 2309 .
 2310 .
 2311 .
 2312 .
 2313 .
 2314 .
 2315 .
 2316 .
 2317 .
 2318 .
 2319 .
 2320 .
 2321 .
 2322 .
 2323 .
 2324 .
 2325 .
 2326 .
 2327 .
 2328 .
 2329 .
 2330 .
 2331 .
 2332 .
 2333 .
 2334 .
 2335 .
 2336 .
 2337 .
 2338 .
 2339 .
 2340 .
 2341 .
 2342 .
 2343 .
 2344 .
 2345 .
 2346 .
 2347 .
 2348 .
 2349 .
 2350 .
 2351 .
 2352 .
 2353 .
 2354 .
 2355 .
 2356 .
 2357 .
 2358 .
 2359 .
 2360 .
 2361 .
 2362 .
 2363 .
 2364 .
 2365 .
 2366 .
 2367 .
 2368 .
 2369 .
 2370 .
 2371 .
 2372 .
 2373 .
 2374 .
 2375 .
 2376 .
 2377 .
 2378 .
 2379 .
 2380 .
 2381 .
 2382 .
 2383 .
 2384 .
 2385 .
 2386 .
 2387 .
 2388 .
 2389 .
 2390 .
 2391 .
 2392 .
 2393 .
 2394 .
 2395 .
 2396 .
 2397 .
 2398 .
 2399 .
 2400 .
 2401 .
 2402 .
 2403 .
 2404 .
 2405 .
 2406 .
 2407 .
 2408 .
 2409 .
 2410 .
 2411 .
 2412 .
 2413 .
 2414 .
 2415 .
 2416 .
 2417 .
 2418 .
 2419 .
 2420 .
 2421 .
 2422 .
 2423 .
 2424 .
 2425 .
 2426 .
 2427 .
 2428 .
 2429 .
 2430 .
 2431 .
 2432 .
 2433 .
 2434 .
 2435 .
 2436 .
 2437 .
 2438 .
 2439 .
 2440 .
 2441 .
 2442 .
 2443 .
 2444 .
 2445 .
 2446 .
 2447 .
 2448 .
 2449 .
 2450 .
 2451 .
 2452 .
 2453 .
 2454 .
 2455 .
 2456 .
 2457 .
 2458 .
 2459 .
 2460 .
 2461 .
 2462 .
 2463 .
 2464 .
 2465 .
 2466 .
 2467 .
 2468 .
 2469 .
 2470 .
 2471 .
 2472 .
 2473 .
 2474 .
 2475 .
 2476 .
 2477 .
 2478 .
 2479 .
 2480 .
 2481 .
 2482 .
 2483 .
 2484 .
 2485 .
 2486 .
 2487 .
 2488 .
 2489 .
 2490 .
 2491 .
 2492 .
 2493 .
 2494 .
 2495 .
 2496 .
 2497 .
 2498 .
 2499 .
 2500 .
 2501 .
 2502 .
 2503 .
 2504 .
 2505 .
 2506 .
 2507 .
 2508 .
 2509 .
 2510 .
 2511 .
 2512 .
 2513 .
 2514 .
 2515 .
 2516 .
 2517 .
 2518 .
 2519 .
 2520 .
 2521 .
 2522 .
 2523 .
 2524 .
 2525 .
 2526 .
 2527 .
 2528 .
 2529 .
 2530 .
 2531 .
 2532 .
 2533 .
 2534 .
 2535 .
 2536 .
 2537 .
 2538 .
 2539 .
 2540 .
 2541 .
 2542 .
 2543 .
 2544 .
 2545 .
 2546 .
 2547 .
 2548 .
 2549 .
 2550 .
 2551 .
 2552 .
 2553 .
 2554 .
 2555 .
 2556 .
 2557 .
 2558 .
 2559 .
 2560 .
 2561 .
 2562 .
 2563 .
 2564 .
 2565 .
 2566 .
 2567 .
 2568 .
 2569 .
 2570 .
 2571 .
 2572 .
 2573 .
 2574 .
 2575 .
 2576 .
 2577 .
 2578 .
 2579 .
 2580 .
 2581 .
 2582 .
 2583 .
 2584 .
 2585 .
 2586 .
 2587 .
 2588 .
 2589 .
 2590 .
 2591 .
 2592 .
 2593 .
 2594 .
 2595 .
 2596 .
 2597 .
 2598 .
 2599 .
 2600 .
 2601 .
 2602 .
 2603 .
 2604 .
 2605 .
 2606 .
 2607 .
 2608 .
 2609 .
 2610 .
 2611 .
 2612 .
 2613 .
 2614 .
 2615 .
 2616 .
 2617 .
 2618 .
 2619 .
 2620 .
 2621 .
 2622 .
 2623 .
 2624 .
 2625 .
 2626 .
 2627 .
 2628 .
 2629 .
 2630 .
 2631 .
 2632 .
 2633 .
 2634 .
 2635 .
 2636 .
 2637 .
 2638 .
 2639 .
 2640 .
 2641 .
 2642 .
 2643 .
 2644 .
 2645 .
 2646 .<

1728 But $2^{101} + 2^m$ must be a power of two. Let me think about
 1729 when the sum of two powers of two is a power of two.
 1730

1731 Suppose we have $2^a + 2^b = 2^c$, with $a \neq b$. Then this is
 1732 possible only when $a = b$, because otherwise, $2^a + 2^b = 2^a(1 + 2^{b-a})$, which is not a power of two unless $1 + 2^{b-a}$
 1733 is a power of two. The only time $1 + 2^d$ is a power of two
 1734 is when $d=0$, which gives $1+1=2$. So $2^a + 2^a = 2^{a+1}$. So
 1735 in this case, if $a=b$, then the sum is 2^{a+1} .
 1736

1737 Therefore, the equation $2^a + 2^b = 2^c$ can only be solved if $a = b$
 1738 and $c = a + 1$. Therefore, in our problem, $2^{101} + 2^m = 2^k$ implies that $101 = m$ and $k=102$. But $m=101$ and $k=102$, but
 1739 since k must be m (since k is the size of the intersection, which is a subset of C , so $k \leq |C|=m$), this would require
 1740 $102 = 101$, which is impossible. Therefore, there is no
 1741 solution unless we have a different approach.
 1742

1743 Wait, but this contradicts the problem's premise, which states
 1744 that such sets exist. Therefore, there must be a different
 1745 interpretation.
 1746

1747 Wait, perhaps " $n(A \cap B \cap C)$ " is not the number of subsets of
 1748 the intersection, but the number of subsets common to all
 1749 three sets A , B , and C ? Wait, that would be different. Wait,
 1750 but the problem says " $n(A \cap B \cap C)$ " normally, the
 1751 notation $n(S)$ for a set S is the number of elements, but in
 1752 the problem statement, it's specified that $n(A)$, $n(B)$, $n(C)$
 1753 are the number of subsets. Wait, the problem says:
 1754

1755 "For sets A , B , and C , where $n(A)$, $n(B)$, and $n(C)$ denote the
 1756 number of subsets of A , B , and C respectively, and $|A| = |B| =$
 1757 100 , if $n(A) + n(B) + n(C) = n(A \cap B \cap C)$, what is the
 1758 minimum possible value of $|A \cap B \cap C|$?"
 1759

1760 Wait, perhaps the notation is confusing. Maybe $n(A \cap B \cap C)$ is
 1761 not the number of subsets of the intersection, but the number
 1762 of subsets common to A , B , and C . Wait, that is, subsets that
 1763 are subsets of A , B , and C . Wait, but a subset of A is not
 1764 necessarily a subset of B or C . So maybe " $n(A \cap B \cap C)$ "
 1765 here is being used to mean the number of subsets that are
 1766 common to all three, i.e., subsets that are subsets of A , B ,
 1767 and C . Which would mean subsets of the intersection $A \cap B \cap C$.
 1768 Because a subset of $A \cap B \cap C$ is a subset of all
 1769 three. So indeed, the number of subsets of the intersection is
 1770 equal to the number of subsets common to all three. Therefore
 1771 , the original interpretation is correct. So $n(A \cap B \cap C)$
 1772 is $2^{|A \cap B \cap C|}$.
 1773

1774 Therefore, the equation is $2^{100} + 2^{100} + 2^{|C|} = 2^{|A \cap B \cap C|}$. So the problem is to find the minimal $k = |A \cap B \cap C|$ such that $2^{101} + 2^{|C|} = 2^k$, with $|C| \leq k$.
 1775

1776 But as we saw earlier, the equation $2^k = 2^{101} + 2^m$, with
 1777 $m \leq k$. However, this equation only holds if the two terms on
 1778 the left can be combined into a single power of two. As
 1779 established before, the sum of two distinct powers of two is a
 1780 power of two only if they are equal (so exponents differ by
 1781

1782 zero) but in that case, it becomes twice the power, which is
 1783 the next exponent. So for example, $2^a + 2^a = 2^{a+1}$.
 1784

1785 In our case, $2^{101} + 2^m = 2^k$, which would require that
 1786 2^{101} and 2^m are equal, which would mean $m=101$, leading
 1787 to $2^{101} + 2^{101} = 2^{102}$, so $k=102$. But in that case, $m=101$
 1788 and $k=102$, which violates the $m < k$ requirement. Therefore,
 1789 no solution exists in that case.

1790 Alternatively, if we consider that 2^m can be combined with
 1791 2^{101} even if $m > 101$. Let's try $m=102$: $2^{101} + 2^{102} =$
 1792 $2^{101}(1 + 2) = 3 \cdot 2^{101}$, which is not a power of two.
 1793 Similarly, $m=103$: $2^{101} + 2^{103} = 2^{101}(1 + 4) = 5 \cdot 2^{101}$,
 1794 not a power of two. It seems like for $m > 101$, the sum is
 1795 $2^{101}(1 + 2^{m-101})$, which is 2^{101} times an odd number
 1796 greater than 1, so not a power of two. Thus, impossible.
 1797

1798 Alternatively, if $m < 101$, then $2^{101} + 2^m = 2^m(1 + 2^{101-m})$. To be a power of two, $1 + 2^{101-m}$ must be a power of two
 1799 . Let $t=101-m$, which is positive since $m < 101$. So $1 + 2^t = 2^s$. As before, this is only possible when $t=0$, which
 1800 would make $m=101$, but we assumed $m < 101$. Therefore, no
 1801 solutions here either.
 1802

1803 Therefore, this suggests that there is no solution where the
 1804 equation holds, which contradicts the problem statement.
 1805 Therefore, there must be an error in my reasoning.
 1806

1807 Wait, but the problem is from a competition or similar, so maybe
 1808 there is a trick here. Let's think differently. Maybe the
 1809 problem is not in the integers. Wait, but cardinalities are
 1810 integers. Alternatively, perhaps the equation isn't meant to
 1811 be exact? No, the problem says $n(A) + n(B) + n(C) = n(A \cap B \cap C)$, so it's an exact equation.
 1812

1813 Alternatively, maybe the problem is using "number of subsets" in a
 1814 different way. Wait, but no, the number of subsets of a set
 1815 with n elements is 2^n . So that part is standard.
 1816

1817 Alternatively, maybe the problem is considering that A , B , C are
 1818 subsets of some universal set, but the problem doesn't specify
 1819 that. But even if they were, the number of subsets of each
 1820 set would still be $2^{|A|}$, etc. So I don't think that's the
 1821 issue.
 1822

1823 Alternatively, maybe " $A \cap B \cap C$ " is not the intersection of
 1824 the sets A , B , C , but some other operation? No, standard
 1825 notation.
 1826

1827 Wait, maybe there's a misinterpretation of $n(A \cap B \cap C)$. Maybe
 1828 it's the number of elements in the intersection, but the
 1829 problem says " $n(A)$, $n(B)$, $n(C)$ denote the number of subsets",
 1830 so $n(A \cap B \cap C)$ would also denote the number of subsets of
 1831 $A \cap B \cap C$. So $2^{|A \cap B \cap C|}$.
 1832

1833 Wait, unless the problem has a typo and instead of $n(A \cap B \cap C)$, it's $|A \cap B \cap C|$. But in that case, the equation would
 1834 be $2^{100} + 2^{100} + 2^{|C|} = |A \cap B \cap C|$, which would be a
 1835 different problem, but unlikely.

1836
 1837 Alternatively , perhaps the problem uses $n(S)$ to denote the number
 1838 of elements in S , but the first sentence says " $n(A)$, $n(B)$, and
 1839 $n(C)$ denote the number of subsets of A , B , and C respectively
 1840 ". So no, $n(A)$ is definitely $2^{|A|}$, etc .
 1841
 1842 Hmm. This is perplexing. Let's check again the equation. $2^{100} +$
 1843 $2^{100} + 2^{|C|} = 2^k$, so $2^{101} + 2^{|C|} = 2^k$. We need to
 1844 find the minimal k where this holds , with $k \geq |C|$. So k is
 1845 the size of $A \cup B \cup C$.
 1846 If we take $|C|=k$, then $2^{101} + 2^k = 2^k$, which is impossible .
 1847 Therefore , $|C|$ must be greater than k .
 1848
 1849 But then , $2^k = 2^{101} + 2^{|C|}$, which implies that 2^k is
 1850 larger than $2^{|C|}$, so $k > |C|$, but that contradicts $|C| \geq k$.
 1851 Therefore , no solution . But the problem says "if $n(A) + n(B) +$
 1852 $n(C) = n(A \cup B \cup C)$ " , so it's assuming such a scenario
 1853 exists . Therefore , there must be a mistake in my reasoning .
 1854
 1855 Wait , perhaps the problem allows C to be a multiset ? But no , the
 1856 problem states "sets" . Or maybe the intersection is not a set ,
 1857 but a different structure ? Unlikely .
 1858 Wait , let's try specific numbers . Suppose $k=101$. Then
 1859 $2^{101} = 2^{101} + 2^m$ – no , that would require $0=2^m$, which
 1860 is impossible . If $k=102$: $2^{102} = 2^{101} + 2^m \Rightarrow 2^{102} -$
 1861 $2^{101} = 2^m \Rightarrow 2^{101}(2-1) = 2^{101} = 2^m$, so $m=101$. But
 1862 then $k=102$, which is greater than $m=101$, which violates $k \geq m$.
 1863
 1864 Similarly , if $k=103$: $2^{103} = 2^{101} + 2^m \Rightarrow 2^m = 2^{103} -$
 1865 $2^{101} = 2^{101}(4-1) = 3 \cdot 2^{101}$, which is not a power of two .
 1866
 1867 $k=104$: $2^{104} = 2^{101} + 2^m \Rightarrow 2^m = 2^{104} - 2^{101} = 2^{101}(8-1) = 7 \cdot 2^{101}$, not a power of two .
 1868
 1869 Continuing , $k=105$: $2^{105} - 2^{101} = 15 \cdot 2^{101} = 15 \cdot 2^{101} =$ not a
 1870 power of two .
 1871
 1872 This pattern continues , and the difference $2^k - 2^{101}$ is
 1873 divisible by 2^{101} but results in an odd number greater than
 1874 1 , which is not a power of two . Therefore , no solutions exist
 1875 for $k > 101$.
 1876
 1877 But this is impossible because the problem must have a solution .
 1878 Therefore , maybe the problem is designed to have the minimal k
 1879 where 2^k is the next power of two after $2^{101} + 2^m$,
 1880 but this is not exact . But the problem states equality , not an
 1881 inequality . So I'm stuck .
 1882
 1883 Alternatively , perhaps the problem is using a different definition
 1884 of subsets . For example , maybe only non-empty subsets ? No ,
 1885 the number of subsets including empty set is 2^n .
 1886
 1887 Alternatively , maybe the problem has a typo , and it should be
 1888 multiplication instead of addition . If it's $n(A) * n(B) * n(C)$
 1889 $= n(A \cup B \cup C)$, then it's different . But the problem says
 1890 "+" .

1890
 1891 Wait, the problem is in Chinese maybe? Wait, no, the user wrote
 1892 the problem in English. Hmm.
 1893
 1894 Alternatively, maybe the problem is from a source where $n(A)$
 1895 denotes the number of elements, which would usually be $|A|$,
 1896 but the problem says $n(A)$ is the number of subsets. So unless
 1897 the problem mixed notation. If the problem had said $|A|$, $|B|$,
 1898 $|C|$ are 100, and $n(A) + n(B) + n(C) = n(A \cup B \cup C)$, with $n(X)$
 1899 being the number of elements, then it's a different problem.
 1900 Let's check that:
 1901 If $|A|=|B|=100$, and $n(X)$ is the number of elements, then $n(A) + n(B)$
 1902 $+ n(C) = 100 + 100 + |C| = 200 + |C|$, and $n(A \cup B \cup C) = |A \cup B \cup C|$.
 1903 Then the equation is $200 + |C| = k$, where $k = |A \cup B \cup C|$.
 1904 But since $|A \cup B \cup C| \leq |A| + |B| + |C| = 100 + |C|$, so $200 + |C| \leq 100 + |C| \Rightarrow |C| \leq -100$, which is impossible. So that can't be.
 1905
 1906 Therefore, the original interpretation seems correct. But then,
 1907 according to that, there's no solution. But the problem is
 1908 asking for the minimal possible value, so perhaps the answer
 1909 is 101? But wait, how?
 1910
 1911 Wait, let's think differently. Maybe the problem is in a universe
 1912 where all sets are subsets of a common universal set, and
 1913 operations are considered within that. Suppose that A and B
 1914 are subsets of some universal set, and C is also a subset.
 1915 Then, the intersection $A \cap B \cap C$ would be a subset of the
 1916 universal set. However, the number of subsets of A is still
 1917 $2^{|A|}$, regardless of the universal set.
 1918
 1919 Alternatively, maybe using some principle of inclusion-exclusion
 1920 for the number of subsets? Hmm, not sure.
 1921
 1922 Wait, another thought: Maybe the equation $n(A) + n(B) + n(C) = n(A \cup B \cup C)$ is in terms of numbers. So $n(A)$ is $2^{|A|}$, $n(B)$
 1923 is $2^{|B|}$, $n(C)$ is $2^{|C|}$, and $n(A \cup B \cup C)$ is $2^{|A \cup B \cup C|}$. So
 1924 we have $2^{|A|} + 2^{|B|} + 2^{|C|} = 2^{|A \cup B \cup C|}$.
 1925
 1926 Let's write this as $2^{|A|} + 2^{|B|} + 2^{|C|} = 2^{|A \cup B \cup C|}$. Let's factor out
 1927 the smaller power of two. Suppose $|C| \leq 101$. Then we can
 1928 factor out $2^{|C|}$:
 1929
 1930 $2^{|C|}(1 + 2^{|A| - |C|} + 2^{|B| - |C|}) = 2^{|A \cup B \cup C|}$. Therefore, $1 + 2^{|A| - |C|} + 2^{|B| - |C|}$ must
 1931 be a power of two. Let's set $t = |A| - |C|$. So $t \geq 0$, and $1 + 2^t + 2^s = 2^k$ for some s . As before, the only solution is $t=0$
 1932 or $t=1$?
 1933
 1934 $t=0$: $1+1=2=2^1$, so $s=1$. Then $t=0$ implies $|A|=|C|=101$.
 1935 So $1 + 2^0 + 2^1 = 1 + 2 + 2 = 2^3 = 8 = 2^k$. Therefore, $k=3$. But
 1936 $|C|=101$, so $k=3 > 101$ which is impossible.
 1937
 1938 If $t=1$: $1+2=3$, which is not a power of two. Similarly, $t=2$: $1+4=5$,
 1939 not a power of two. So no solution.
 1940
 1941 If $|C| > 101$, then $2^{|A|} + 2^{|B|} + 2^{|C|} = 2^{|A \cup B \cup C|}$ implies we factor out
 1942 $2^{|A|}$: $2^{|A|}(1 + 2^{|B| - |A|} + 2^{|C| - |A|}) = 2^{|A \cup B \cup C|}$, so $1 + 2^{|B| - |A|} + 2^{|C| - |A|} = 2^{|A \cup B \cup C|}$. Let's set $m = |C| - |A| > 0$, so $1 + 2^m + 2^s = 2^k$

1944 -101}. The only solution is $m=0$, which gives $1+1=2^1$. But $m > 0$, so no solution.

1947 Therefore, this suggests no solution exists, which contradicts the
1948 problem's wording. Therefore, I must have made a wrong
1949 assumption.

Wait, perhaps the problem allows for a universe where elements are counted with multiplicity? Like, multisets? But the problem specifies "sets", so elements are distinct.

1954 Alternatively, perhaps the problem is a trick question where the
1955 minimal possible value is 101, but the equation is not exactly
1956 satisfied. But the problem says "if $n(A) + n(B) + n(C) = n(A$
1957 $B + C)$ ", so it's given that this equation holds. So the
1958 answer must be derived under that condition.

1960 Wait, maybe there's a mistake in my calculation. Let me try
 1961 plugging in $k=101$. Then $2^{101}=2^{101}+2^{|C|}$, which
 1962 implies $|C|$ is negative infinity, impossible. k
 1963 $=102: 2^{102}=2^{101}+2^{|C|} \Rightarrow 2^{|C|}$
 1964 $=2^{102}-2^{101}=2^{101}$, so $|C|=101$. But then $k=102 > |C|$
 1965 $=101$. So this is impossible.

1966 Wait, but maybe the problem doesn't require that the intersection
1967 is a subset of C? But that's basic set theory. A B C
1968 is a subset of C by definition.

1970 Alternatively, maybe the problem uses a different definition of
1971 intersection? No, standard intersection.

1972 Alternatively, perhaps A, B, C are not sets of elements but of
1973 subsets. Wait, the problem says "sets A, B, C", so they are
1974 standard sets.

1976 Alternatively, maybe the problem allows C to be equal to A B
1977 C. If C is equal to A B C, then $|C|=k$. Then the
1978 equation becomes $2^{100} + 2^{100} + 2^k = 2^k$, which implies
1979 $2^{101} + 2^k = 2^k$, leading to $2^{101} = 0$, which is
1980 impossible.

1981
1982 This is really confusing. Since the problem is from a math
1983 competition, maybe the answer is 102, even though it's greater
1984 than $|C|$, but the problem doesn't require $|C|$ to be finite or
1985 something? But A and B have size 100, which is finite, and
1986 the intersection can't be larger than A or B. So the
1987 intersection has to be at most 100.

1988 Wait, wait! The problem says $|A|=|B|=100$. So the intersection A
1989 $B \cap C$ is a subset of A and B, so $|A \cap B \cap C| \leq 100$.
1990 But earlier, we tried $k=101$ and more, which is impossible
1991 since the intersection can't exceed 100.

1993 Oh! This is the key mistake I made earlier. Since A and B have
 1994 size 100, their intersection with C can't exceed 100. So
 1995 $k \leq 100$. Therefore, when I considered $k=101$ or higher, that's
 1996 impossible. Therefore, the maximum possible k is 100. But then
 1997 the equation $2^{101} + 2^m = 2^k$ with $k \leq 100$. But 2^k is

1998 at most 2^{100} , but the left-hand side is at least 2^{101} ,
 1999 which is larger than 2^{100} . Therefore, it's impossible.
 2000

2001 But this contradicts the problem's stipulation that the equation
 2002 holds. Therefore, the only possible conclusion is that there
 2003 is a mistake in the problem statement, or perhaps the answer
 2004 is 101, but the problem has an error. But since the problem
 2005 asks for the minimal possible value of $|A \cap B \cap C|$, and
 2006 given the constraints, it's impossible, but maybe I'm missing
 2007 something.

2008 Wait, here's a different approach. Let's denote $x = |A \cap B \cap C|$.
 2009 We need to have $2^{100} + 2^{100} + 2^{|C|} = 2^x$. Therefore
 2010 $, 2^{101} + 2^{|C|} = 2^x$.

2011 But x is the size of the intersection, which is 100. Therefore,
 2012 $2^x = 2^{100}$, but the left-hand side is $2^{101} +$
 2013 something, which is at least 2^{101} . Therefore, $2^x > 2^{101}$, which requires $x > 101$. But $x \leq 100$, so there's no
 2014 solution. Hence, it's impossible.

2015 But the problem is presented as a valid question, so likely I have
 2016 a mistake. Wait, maybe I miscounted the number of subsets.
 2017 Wait, if A has $|A|=100$, then $n(A)=2^{100}$. Similarly for B and
 2018 C . So the equation is correct.

2019 Alternatively, the problem might be in the translation. Maybe "the
 2020 number of subsets" refers to proper subsets? In that case,
 2021 the number would be $2^n - 1$. So if $n(A)$ is the number of
 2022 proper subsets of A , then $n(A)=2^{100}-1$. Similarly for B and
 2023 C . Then the equation becomes $(2^{100}-1)+(2^{100}-1)+(2^{|C|}-1)=2^x-1$.

2024 Let's compute this: $2^{101} - 2 + 2^{|C|} - 1 = 2^x - 1$
 2025 $+ 2^{|C|} - 3 = 2^x - 2^{101} + 2^{|C|} - 2 = 2^x$.

2026 But this seems more complicated, and there's no indication in the
 2027 problem statement that subsets are proper. So this is probably
 2028 not the case.

2029 Alternatively, maybe the problem uses a different definition of "number
 2030 of subsets," but it's highly unlikely.

2031 Therefore, given that the problem states such an equation is
 2032 possible, but according to standard set theory and mathematics
 2033 , there is no solution. But since the problem asks for an
 2034 answer, I must have missed something.

2035 Wait, maybe the problem is not about sets but about something else
 2036 . Maybe vector spaces? No, the problem talks about sets A , B ,
 2037 C .

2038 Wait, another idea: maybe the sets A , B , and C are such that A and
 2039 B are subsets of C . If A and B are subsets of C , then $A \cap B \cap C = A \cap B$. So $|A \cap B \cap C| = |A \cap B|$. But A and B
 2040 have size 100, but we don't know their intersection. However,
 2041 the equation becomes $2^{100} + 2^{100} + 2^{|C|} = 2^{|A \cap B|}$. But
 2042 since A and B are subsets of C , $|C| \geq 100$. But then $2^{|C|} \geq 2^{100}$, so the left-hand side is $2^{100} + 2^{100} + 2^{|C|}$

2052 $+2^{\{100\}} = 3 * 2^{100}$, which is greater than $2^{\{|A \cap B|\}}$, but $|A \cap B|$ can be at most 100, so $2^{\{|A \cap B|\}} \leq 2^{\{100\}}$, which is less than $3 * 2^{\{100\}}$. Therefore, no solution.

2056 Hmm. I'm stuck. Given the problem's constraints, there is no possible solution, but the problem is asking for one. Perhaps the answer is 101, acknowledging that there's a contradiction but requiring the minimal k where $2^{\{k\}}$ is the next power after $2^{101} + 2^{\{|C|\}}$. But even then, the minimal k would be 102, but that exceeds the maximum possible size of the intersection.

2063 Wait, but if we ignore the constraint that $k \leq |C|$ and $k \leq 100$, just find the minimal k such that $2^{\{k\}} \geq 2^{101} + 2^{\{m\}}$ for some m. But then $k=102$ when $m=101$, but that violates the other constraints.

2068 Alternatively, maybe the problem has a typo and it should be $n(A) + n(B) + n(C) = n(A \cup B \cup C)$. Then the problem would be different. Let me check: $n(A \cup B \cup C)$ is the number of subsets of the union. If that's the case, then the equation is $2^{\{100\}} + 2^{\{100\}} + 2^{\{|C|\}} = 2^{\{|A \cup B \cup C|\}}$, which might be possible. But the problem says intersection, not union.

2074 Alternatively, maybe the problem is in another dimension, but that's not likely.

2077 Given that I'm stuck and the problem is likely expecting an answer of 101, even though it violates the cardinality constraints, or perhaps 101 is the minimal k where $2^{\{k\}}$ is greater than the sum. But in math competitions, sometimes you're supposed to find k such that the sum is less than or equal to $2^{\{k\}}$, but the problem states equality.

2083 Alternatively, maybe the problem allows overlapping subsets in the count. But no, subsets are unique.

2086 Wait, here's a last-ditch idea: perhaps A, B, and C are not sets of elements but of sets themselves, and the intersection is in some lattice of sets. But the problem statement says "sets A, B, C", so they're standard sets.

2091 Given that I've exhausted all avenues and still can't find a solution that fits the problem's constraints, but the problem must have an answer, I'm forced to conclude that the answer is 101, even though it violates the subset relation.

2095 Alternatively, maybe the problem's answer is 101, with $k=101$, $|C|=101$, but then $2^{\{101\}} = 2^{\{101\}} + 2^{\{101\}} = 2^{\{102\}}$, which is not true. Therefore, I'm confused.

2098 Wait, $2^{\{101\}} + 2^{\{101\}} = 2^{\{102\}}$, so if $|C|=101$, then the equation would be $2^{\{101\}} + 2^{\{101\}} = 2^{\{102\}}$, so $k=102$. But $k=102$ is impossible because the intersection cannot exceed 100. Therefore, there's no solution, but the problem states "if $n(A) + n(B) + n(C) = n(A \cup B \cup C)$ ", which implies that there is a solution. This is a paradox.

2104 Wait, unless the sets are allowed to be infinite. For example, if 2105 C is an infinite set, then $2^{\{|C|\}}$ is infinite, and $2^{\{101\}}$

2106 $+2^{|C|}$ would be equal to $2^{|C|}$, assuming $|C|$ is infinite.
 2107 But then the equation $2^{101} + 2^{|C|} = 2^k$ implies that if $|C|$ is infinite, then 2^k is also infinite, so k must be
 2108 infinite. But the problem asks for the minimal possible value,
 2109 which would be finite. Therefore, this is not the case.
 2110

2111 Given all this, I think there's either a mistake in the problem or
 2112 a trick that I'm missing. Since the problem is about sets and
 2113 their subsets, and the equation seems impossible under
 2114 standard interpretations, maybe the answer is 101, assuming
 2115 that we ignore the cardinality constraint of the intersection.
 2116 But since the intersection can't exceed 100, this is
 2117 impossible. Therefore, the minimal possible value is 101, but
 2118 I'm not sure.
 2119

2120 Wait, but if $|A \cap B \cap C|$ could be 101, then even though A and
 2121 B have size 100, their intersection with C can't exceed 100.
 2122 So maybe the problem is designed to have an answer of 101,
 2123 assuming that the intersection can be larger than A or B ? But
 2124 that's impossible.

2125 I think I must have missed something. Maybe instead of
 2126 interpreting $n(A)$ as the number of subsets, it's the number of
 2127 elements? If so, then $n(A)=100$, $n(B)=100$, $n(C)=|C|$, and $n(A \cap B \cap C)=|A \cap B \cap C|$. Then the equation is $100 + 100 + |C| = |A \cap B \cap C|$. But since $|A \cap B \cap C| \leq 100$, this
 2128 implies $|C| \leq -100$, which is impossible. So no.
 2129

2132 Alternatively, if $n(A)$ is the number of elements, and the problem
 2133 meant to say $|A|$, but that's not what was written.

2134 Given the problem's phrasing, I think the answer must be \boxed{101}, even though the logical steps suggest it's impossible.
 2135 Perhaps there's a different way to combine the subsets.
 2136 Alternatively, maybe considering that the intersection is
 2137 being counted as elements, not subsets. But that would not
 2138 align with the initial definitions.
 2139

2141 Since in all cases the only possible value that comes close is
 2142 102, but it's invalid due to size constraints, and the problem
 2143 asks for the minimal possible value, perhaps the answer is
 2144 101, even if it's not strictly possible, and the problem has
 2145 an error. Alternatively, the answer is 102, and the problem
 2146 allows intersections larger than the sets. But this is not
 2147 standard.

2148 Wait, maybe A and B are allowed to have sizes larger than 100?
 2149 Wait, no, the problem says $|A|=|B|=100$.

2151 Given that I can't find a valid solution and the problem likely
 2152 expects an answer, I'll go with 101, assuming that maybe the
 2153 problem had a typo and the equation is supposed to be
 2154 multiplication, but that's speculative. Alternatively, 101 is
 2155 the answer.

2156
 2157 **Final Answer**
 2158 \boxed{101}
 2159 </think>

2160

2161 Given sets \mathcal{A} , \mathcal{B} , and \mathcal{C} with $|\mathcal{A}| = |\mathcal{B}| = |\mathcal{C}| = 100$, we
 2162 need to find the minimum possible value of $|\mathcal{A} \cap \mathcal{B} \cap \mathcal{C}|$
 2163 given that the equation $n(\mathcal{A}) + n(\mathcal{B}) + n(\mathcal{C}) = n(\mathcal{A} \cap \mathcal{B} \cap \mathcal{C})$ holds. Here, $n(\mathcal{S})$ denotes the number of subsets
 2164 of a set \mathcal{S} , which is $2^{|S|}$.

2166

1. The number of subsets of \mathcal{A} , \mathcal{B} , and \mathcal{C} are 2^{100} , 2^{100} , and $2^{|C|}$ respectively.
2. The equation given is $2^{100} + 2^{100} + 2^{|C|} = 2^{|\mathcal{A} \cap \mathcal{B} \cap \mathcal{C}|}$.
3. Simplifying the left-hand side, we get $2^{101} + 2^{|C|} = 2^{|\mathcal{A} \cap \mathcal{B} \cap \mathcal{C}|}$.
4. Let $k = |\mathcal{A} \cap \mathcal{B} \cap \mathcal{C}|$. We need to find the smallest k such that $2^{101} + 2^{|C|} = 2^k$.
5. For the equation $2^{101} + 2^{|C|} = 2^k$ to hold, $2^{101} + 2^{|C|}$ must be a power of two.
6. The only way the sum of two powers of two is another power of two is if they are consecutive powers of two. However, $2^{101} + 2^{101} = 2^{102}$, which would require $|C| = 101$ and $k = 102$, but k cannot exceed $|C|$.
7. Considering the constraints and the problem's requirement for a solution, the minimal possible value of $|\mathcal{A} \cap \mathcal{B} \cap \mathcal{C}|$ that fits the equation, despite the constraints, is found to be 101.

2183

2184 Thus, the minimum possible value of $|\mathcal{A} \cap \mathcal{B} \cap \mathcal{C}|$ is 101.

2186

2187

2188

2189

2190

2191

2192

2193

2194

2195

2196

2197

2198

2199

2200

2201

2202

2203

2204

2205

2206

2207

2208

2209

2210

2211

2212

2213