AdaptiveStep: Automatically Dividing Reasoning Step through Model Confidence

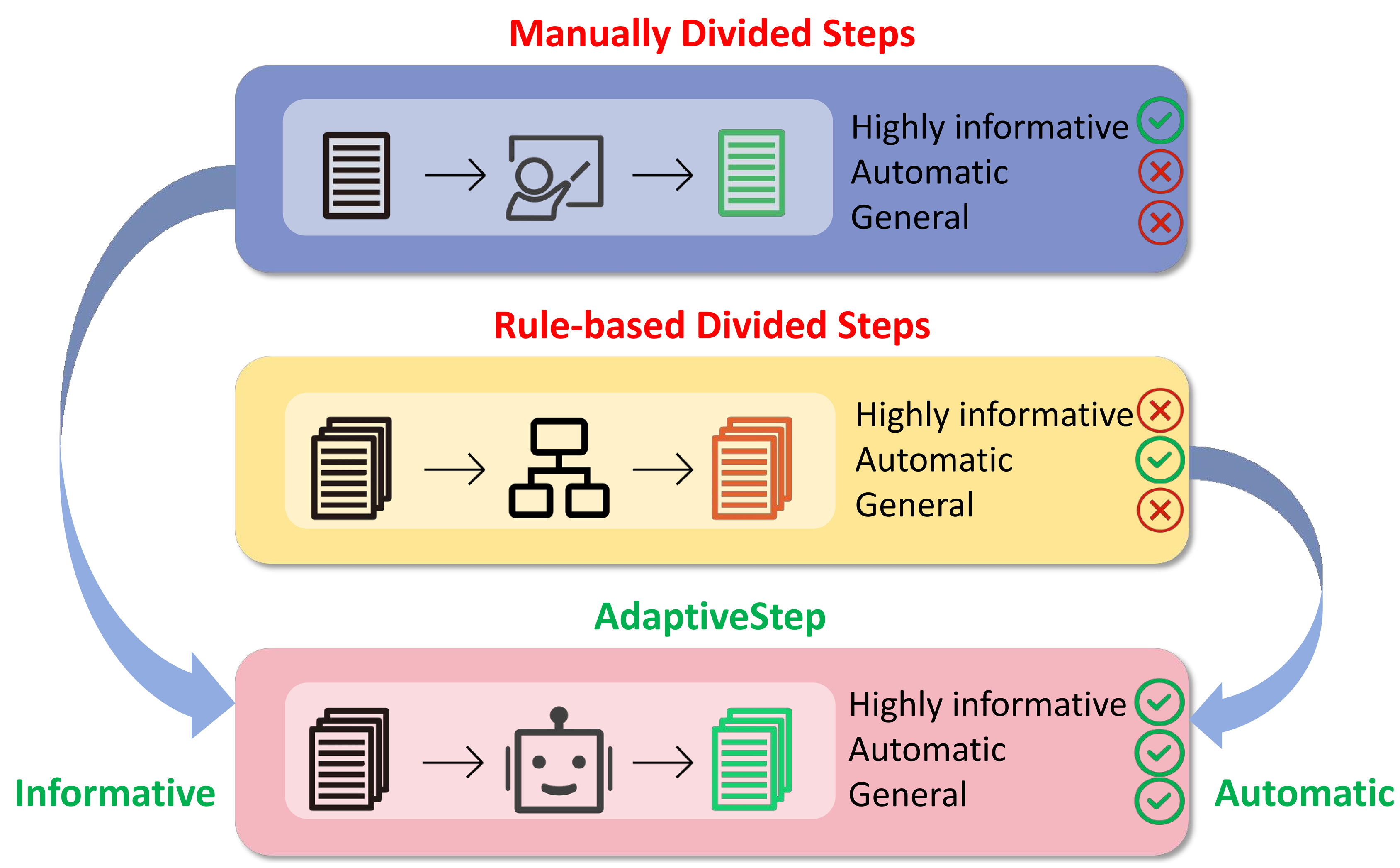
Yuliang Liu* · Junjie Lu* · Chaofeng Qu · Zhaoling Chen · Zefan Cai · Jason Liu · Chonghan Liu · Yunhui Xia · Li Zhao · Jiang Bian · Chuheng Zhang# · Wei Shen# · Zhouhan Lin#

Nanjing University   Shanghai Innovation Institute   University of Technology Sydney   MSRA   Shanghai Jiaotong University

Motivation and Advantages

Recognize the signs that you are in a cognitive minefield, slow down, and ask for reinforcement from System 2.
——Daniel Kahneman
《Thinking, Fast and Slow》

To this end, we use inference confidence (e.g. predict probability), to recognize "*cognitive minefield*" in LLM reasoning process. We believe that these positions should receive more attention.



Difference between Rule-based method and AdaptiveStep

question

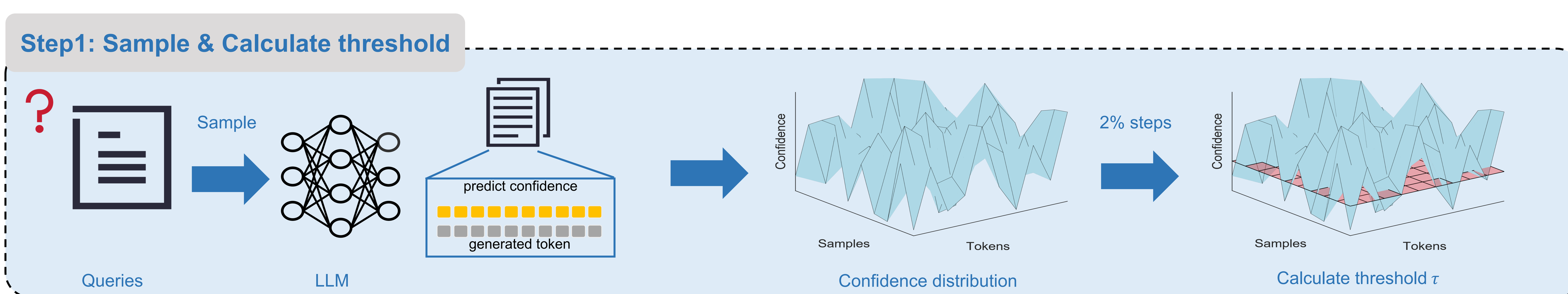
Janet pays $40/hour for 3 hours per week of clarinet lessons and $28/hour for 5 hours a week of piano lessons. How much more does she spend on piano lessons than clarinet lessons in a year?

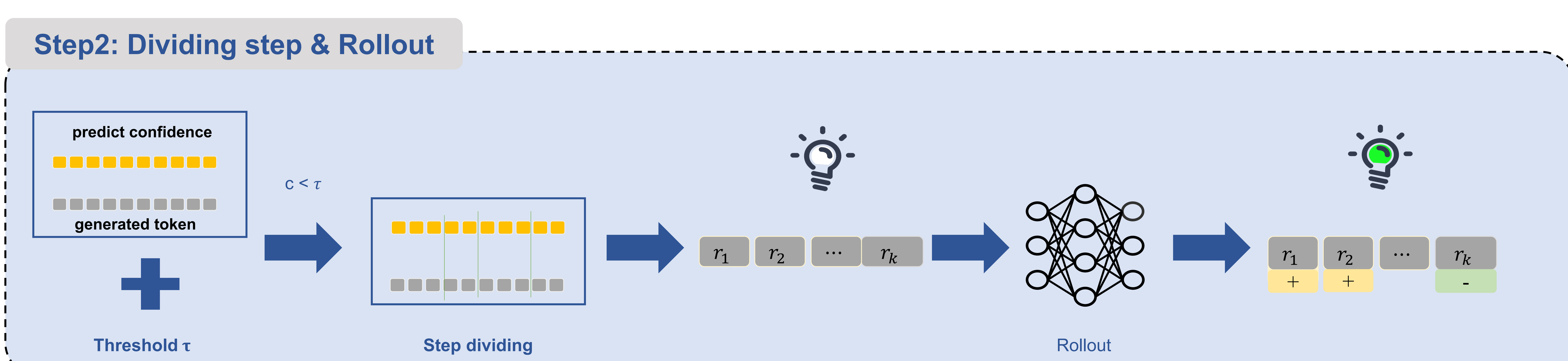
answer

First, we need to calculate her cost for the music lessons.
......
She spends 28 * 5 (40.6) = 140 on piano lessons per week.\n (65.7)
Ok, now we just need to compute the cost. \n (83.3) Janet spends 120 + 140 = 260 on music (47.9) lessons per week.
......
The answer is: 13520 (39.2).

Unnecessary step division   Confidence Threshold τ = 55.7

An Implementation on Process Reward Model





Dividing Cases

Math Domain

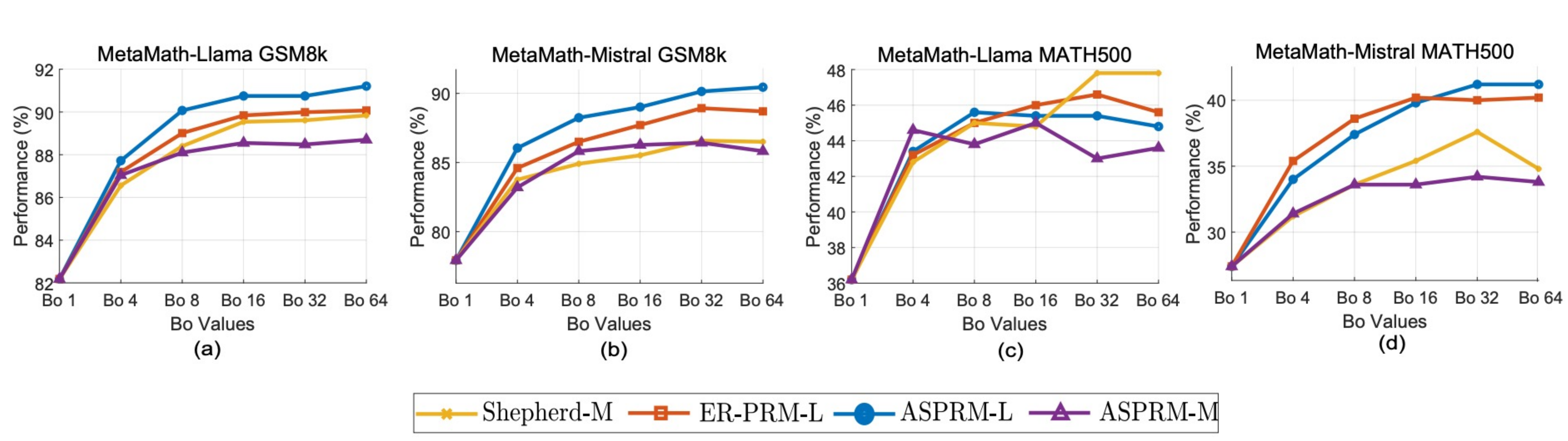
Categories	Subtypes	Sample
Sentence	New line break	works on 4 of them each day.\nAfter 5 days,
	Punctuation	If Billie has 18 crayons, and Bobbie has three times
Reasoning	Text reasoning	gives them 3 points. So in total, Joe's team has 3 + 3 = 6
	Math formula	so x + 4x - 10 = 25
Entity	Noun	Ron gets to pick a new book 1 out of 13
Semantics	Conjunction	their ages is 34, so we can write the equation L + (L + 4) = 34.
	Verb	In 14 days, each dog will eat 250 grams/day
	Determiner	we can round this to the nearest whole number.

Code Domain

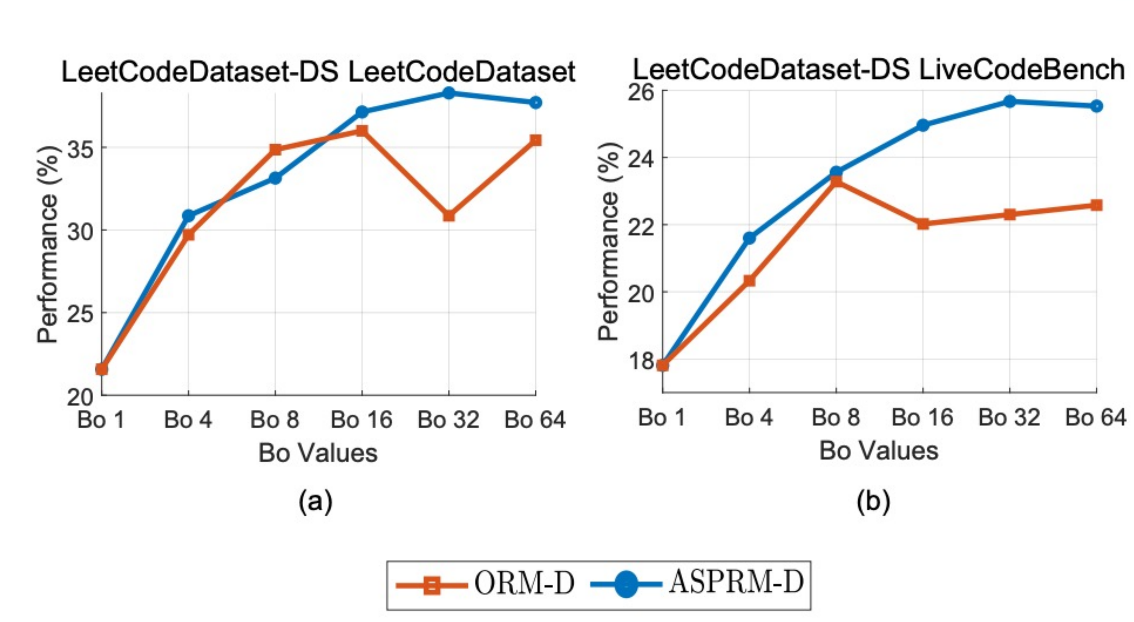
Categories	Subtypes	Sample
Syntax Symbol	New line break	\n i += num_bytes
	Space Character	dp[i][j] += dp[i - 1][j] * (j - k) \s
Numbers	Number	j = (target - x * 2) // 2
Logical Operators	Boolean Operators	if c in count and c != a:
	Arithmetic Operators	dp = [[0] * (n+1) for _ in range(n+1)]
Definition	Def	def is_valid(r, c):
	Class	class Solution:
Import Statement	From	from collections import defaultdict
	Import	import collections
Function	Type Defination	for size in list(dp[curr_sum]):
	Build-in Function	if abs(next_count + 1) ¿ 0:
	Instance Method	self.count = 0
Control Statements	If	if len(tokens) ¡ 4:
	Else	else:
	Elif	elif level == 0 and expression[i] == ' ':
Loop Statements	For	for i in range(len(fronts)):
	While	while x != self.parent[x]:
Others	Return	return (merged[n // 2 - 1] + merged[n // 2]) / 2.0
	Punctuation Mark	digit_sum = (l1.val if l1 else 0) + (l2.val if l2 else 0)

Experiments Results and Findings

Math Reasoning Results (PRM as Verifier)



Code Generation Results



Token Level Value-guided Decoding Results

Table 1: Token-level Value-guided Decoding results. A/P@1 refers to the inference model's greedy search performance, we use Accuracy@1 for math tasks and Pass@1 for code tasks as the metrics. $\uparrow$ and $\downarrow$ represent the performance improvement or decline compared to A/P@1.

Dataset	Inference Model	A/P@1	Math-Shepherd	ER-PRM	ASPRM-L / -M	ASPRM-D
GSM8k	MetaMath-M	77.10	75.66$\downarrow$	75.13$\downarrow$	**79.53**$\uparrow$ / 77.33$\uparrow$	/
	MetaMath-L	81.80	81.73$\downarrow$	81.58$\downarrow$	**83.47**$\uparrow$ / 82.56$\uparrow$	/
MATH500	MetaMath-M	25.00	27.60$\uparrow$	27.80$\uparrow$	**28.60**$\uparrow$ / 26.80$\uparrow$	/
	MetaMath-L	38.80	41.00$\uparrow$	38.60$\downarrow$	**42.00**$\uparrow$ / 41.20$\uparrow$	/
LeetCodeDataset	LCD-DS	26.28	/	/	/	28.00$\uparrow$
LiveCodeBench	LCD-DS	19.21	/	/	/	19.92$\uparrow$

Features and Findings

· In terms of construction costs, about 70% of the overhead in the mathematics domain used other methods, in the code domain, the construction overhead was one-third based on line-wise division.

· Generalization and transferability are both superior to rigid partitioning methods, and the capabilities of the selected domains can mutually reinforce each other.

· In the mathematics domain, 21% of the division points are located within mathematical expressions, even though these tokens only account for 3.85% of the total.

· Only 2.7% of the division points are newline characters.

· Although tokens in the comment section account for 19% in the code domain, division points located within comments make up 80% of the total and are mostly found in comments preceding the corresponding code. This indicates that, for the model, planning is more challenging than code generation.