ReGenesis: LLMs can grow into Reasoning generalists via self improvement

Authors: Xiangyu Peng, Congying Xia, Xinyi Yang, Caiming Xiong, Chien-Sheng (Jason) Wu, ***Chen Xing***

2025.4.26

Motivation

Recent research has demonstrated that **post-training with explicit intermediate reasoning trajectories can improve the performance of large language models** (LLMs) across a wide range of complicated reasoning tasks, such as mathematical reasoning, commonsense reasoning, etc

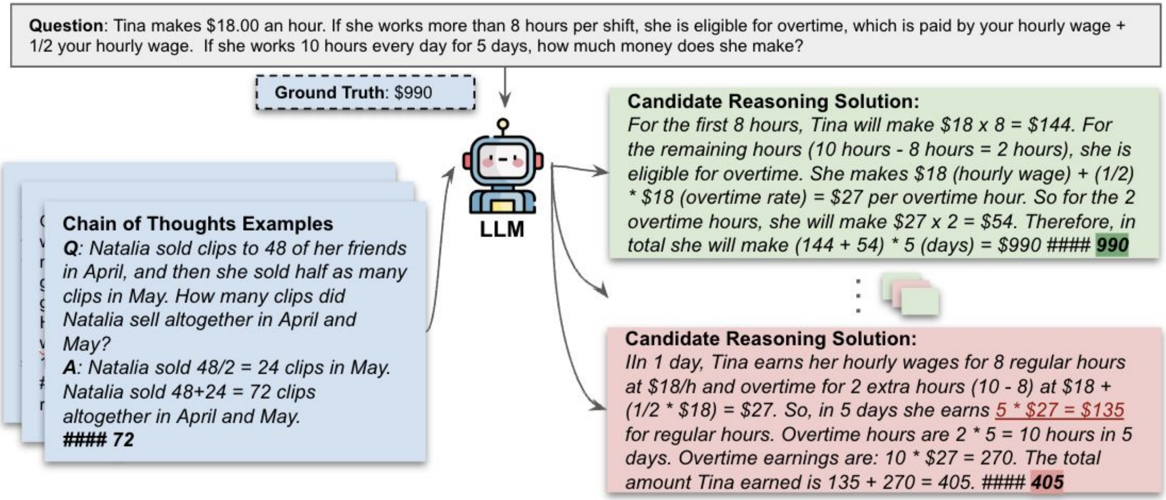
However, their acquisition of **high-quality reasoning** trajectory data in the post-training phase demands **meticulous supervision for each reasoning step, either from humans or superior models.**

Therefore, although high-quality explicit reasoning paths can help post training, they are either **expensive** and **time-consuming** due to the additional substantial human supervision, or constrained by the license of superior models.

How far an LLM can improve its reasoning by self-generating reasoning paths as training data, **without any additional supervision beyond final answers**?

Self-Improvement

Existing approaches towards this direction, **STaR**(Zelikman et al., 2022) and **LMSI**(Huang et al.,
2023), prompt the given LLM to generate a chain-of-thought (**CoT**) (Wei et al., 2022) reasoning
steps and filter them with **final ground truth** labels or **self consistency**



Self-Improvement

However, through our extensive experiments, we have found that they struggle to improve the given LLMs on out-of-domain (**OOD**) tasks that are different from the fine-tuning data that they generate reasoning paths and further train the model on.

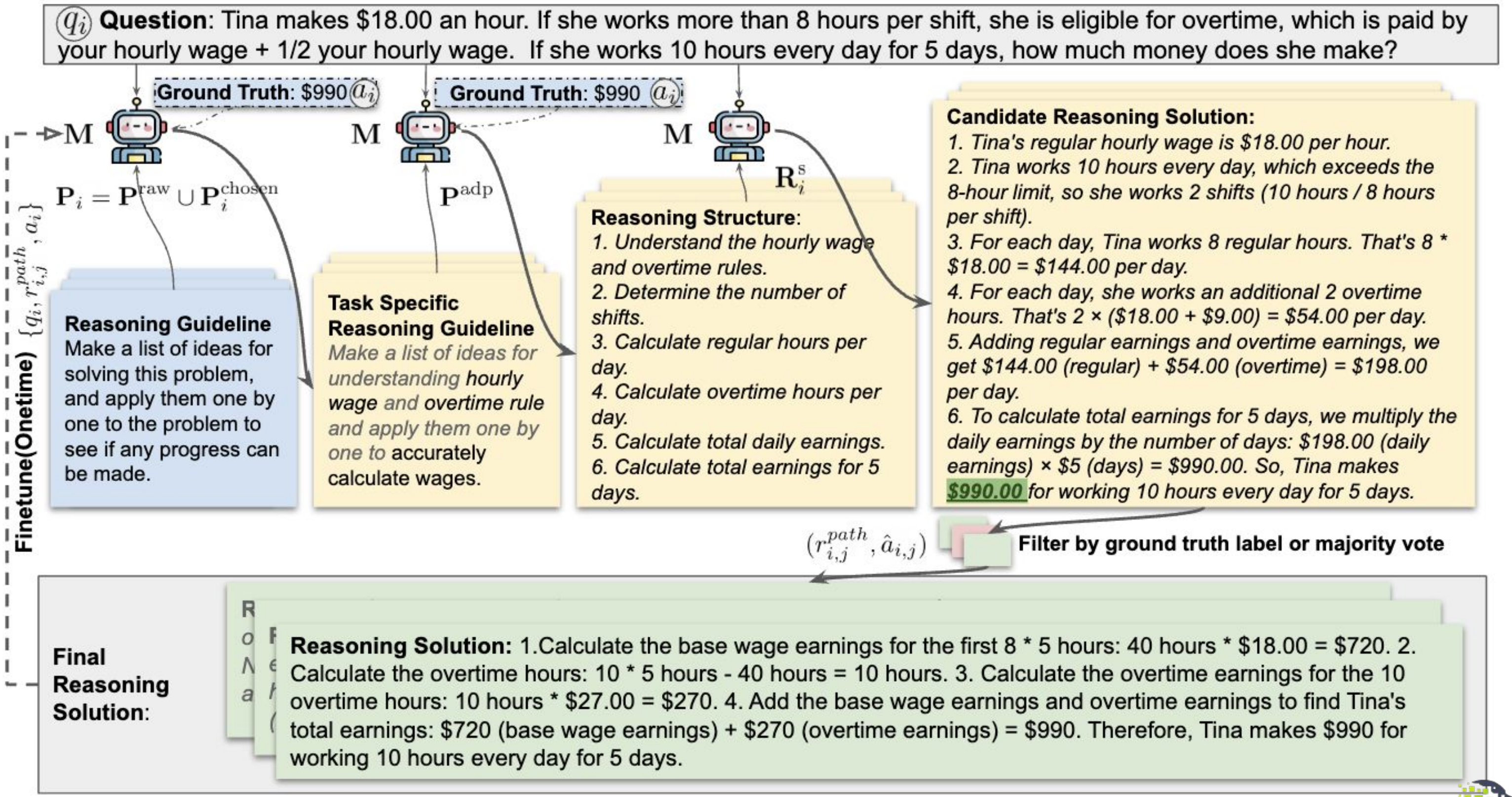
Training Datasets	Training Methods	Test Datsets					
		ASDIV (Math)	SVAMP (Math)	AQUA (Math)	BBH (Logical)	ANLI (NLI)	OpenbookQA (Commonsense)
GSM8K (Math)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	54.0% ↓	40.0% ↓	29.1% ↓	53.6% ↓	44.6% ↑	72.6% ↓
	LMSI w/ GT	77.3% ↑	72.2% ↓	31.1% ↓	59.5% ↓	43.4% ↑	73.2% ↓
	STaR	79.6% ↑	71.5% ↓	46.9% ↑	47.4% ↓	45.0% ↑	72.8% ↓
	Ours	**81.2%** ↑	**83.9%** ↑	**48.8%** ↑	**69.3%** ↑	**49.5%** ↑	**81.4%** ↑
NumGLUE (Math)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	53.8% ↓	54.9% ↓	32.3% ↓	42.4% ↓	37.0% ↓	64.8% ↓
	LMSI w/ GT	75.7% ↓	78.2% ↑	40.6% ↓	59.5% ↓	35.1% ↓	72.6% ↓
	STaR	**79.6%** ↑	76.1% ↑	37.0% ↓	58.5% ↓	41.9% ↑	71.6% ↓
	Ours	76.9% ↓	**79.4%** ↑	**48.4%** ↑	**61.7%** ↑	**50.0%** ↑	**79.8%** ↑
ReClor (Logical)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	62.0% ↓	56.0% ↓	22.0% ↓	44.6% ↓	47.1% ↑	71.4% ↓
	LMSI w/ GT	77.9% ↑	76.8% ↑	45.6% ↑	53.9% ↓	35.1% ↓	72.4% ↓
	STaR	76.1% ↓	74.6% ↓	46.1% ↑	60.6% ↓	42.5% ↑	77.2% ↑
	Ours	**76.4%** ↓	**76.5%** ↑	**49.6%** ↑	**66.8%** ↑	**44.8%** ↑	**81.4%** ↑
ARC-c (Logical)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	57.7% ↓	57.2% ↓	18.9% ↓	37.6% ↓	36.3% ↓	78.4% ↑
	LMSI w/ GT	70.9% ↓	72.0% ↓	32.7% ↓	60.8% –	32.5% ↓	79.4% ↑
	STaR	77.0% ↓	76.2% ↑	40.6% ↓	60.7% ↓	**47.4%** ↑	**84.2%** ↑
	Ours	**81.6%** ↑	**79.5%** ↑	**46.5%** ↑	**66.0%** ↑	46.4% ↑	82.8% ↑
StrategyQA (Commonsense)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	69.4% ↓	72.3% ↓	43.7% ↑	52.3% ↓	44.9% ↑	62.2% ↓
	LMSI w/ GT	56.3% ↓	56.8% ↓	40.6% ↓	60.6% ↓	39.8% ↑	68.4% ↓
	STaR	79.8% ↑	76.2% ↑	**43.3%** ↑	62.9% ↑	37.3% ↓	77.4% ↑
	Ours	**81.3%** ↑	**81.1%** ↑	42.9% ↑	**65.9%** ↑	**55.3%** ↑	**80.4%** ↑

Self-Improvement

Existing **self-improvement** methods don't **generalize well on OOD tasks** and cannot make given LLMs **reasoning generalists.**

Training Datasets	Training Methods	Test Datsets					
		ASDIV (Math)	SVAMP (Math)	AQUA (Math)	BBH (Logical)	ANLI (NLI)	OpenbookQA (Commonsense)
GSM8K (Math)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	54.0% ↓	40.0% ↓	29.1% ↓	53.6% ↓	44.6% ↑	72.6% ↓
	LMSI w/ GT	77.3% ↑	72.2% ↓	31.1% ↓	59.5% ↓	43.4% ↑	73.2% ↓
	STaR	79.6% ↑	71.5% ↓	46.9% ↑	47.4% ↓	45.0% ↑	72.8% ↓
	Ours	**81.2%** ↑	**83.9%** ↑	**48.8%** ↑	**69.3%** ↑	**49.5%** ↑	**81.4%** ↑
NumGLUE (Math)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	53.8% ↓	54.9% ↓	32.3% ↓	42.4% ↓	37.0% ↓	64.8% ↓
	LMSI w/ GT	75.7% ↓	78.2% ↑	40.6% ↓	59.5% ↓	35.1% ↓	72.6% ↓
	STaR	**79.6%** ↑	76.1% ↑	37.0% ↓	58.5% ↓	41.9% ↑	71.6% ↓
	Ours	76.9% ↓	**79.4%** ↑	**48.4%** ↑	**61.7%** ↑	**50.0%** ↑	**79.8%** ↑
ReClor (Logical)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	62.0% ↓	56.0% ↓	22.0% ↓	44.6% ↓	47.1% ↑	71.4% ↓
	LMSI w/ GT	77.9% ↑	76.8% ↑	45.6% ↑	53.9% ↓	35.1% ↓	72.4% ↓
	STaR	76.1% ↓	74.6% ↓	46.1% ↑	60.6% ↓	42.5% ↑	77.2% ↑
	Ours	**76.4%** ↓	**76.5%** ↑	**49.6%** ↑	**66.8%** ↑	**44.8%** ↑	**81.4%** ↑
ARC-c (Logical)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	57.7% ↓	57.2% ↓	18.9% ↓	37.6% ↓	36.3% ↓	78.4% ↑
	LMSI w/ GT	70.9% ↓	72.0% ↓	32.7% ↓	60.8% –	32.5% ↓	79.4% ↑
	STaR	77.0% ↓	76.2% ↑	40.6% ↓	60.7% ↓	**47.4%** ↑	**84.2%** ↑
	Ours	**81.6%** ↑	**79.5%** ↑	**46.5%** ↑	**66.0%** ↑	46.4% ↑	82.8% ↑
StrategyQA (Commonsense)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	69.4% ↓	72.3% ↓	43.7% ↑	52.3% ↓	44.9% ↑	62.2% ↓
	LMSI w/ GT	56.3% ↓	56.8% ↓	40.6% ↓	60.6% ↓	39.8% ↑	68.4% ↓
	STaR	79.8% ↑	76.2% ↑	**43.3%** ↑	62.9% ↑	37.3% ↓	77.4% ↑
	Ours	**81.3%** ↑	**81.1%** ↑	42.9% ↑	**65.9%** ↑	**55.3%** ↑	**80.4%** ↑

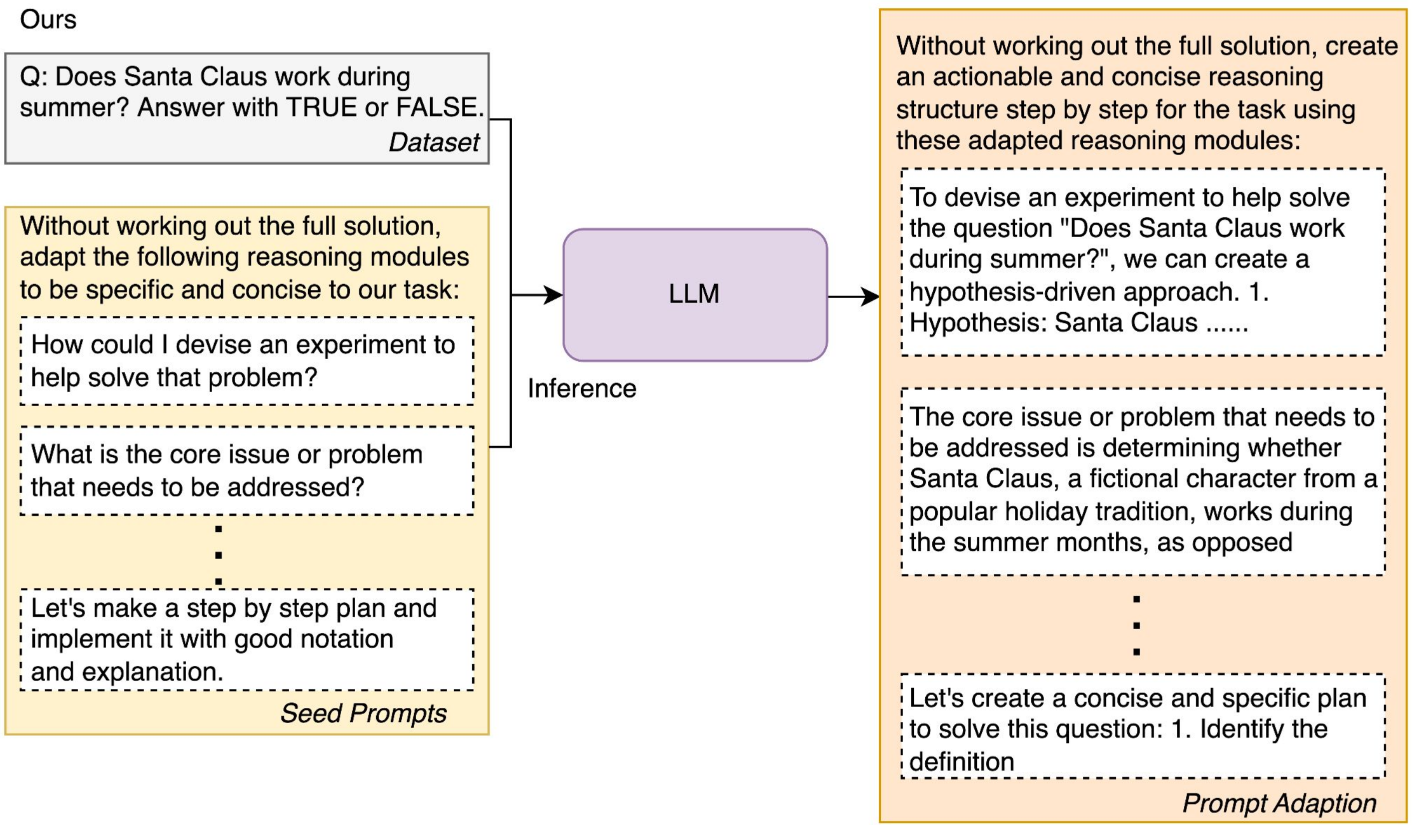
REGENESIS



q_i Question: Tina makes $18.00 an hour. If she works more than 8 hours per shift, she is eligible for overtime, which is paid by your hourly wage + 1/2 your hourly wage. If she works 10 hours every day for 5 days, how much money does she make?
Ground Truth: $990 a_i
Ground Truth: $990 a_i
M
M
M
$\mathbf{P}_i = \mathbf{P}^{\text{raw}} \cup \mathbf{P}_i^{\text{chosen}}$
$\mathbf{P}^{\text{adp}}$
$\mathbf{R}_i^s$
Finetune(Onetime)
$\{q_i, r_{i,j}^{path}, a_i\}$
Reasoning Guideline
Make a list of ideas for solving this problem, and apply them one by one to the problem to see if any progress can be made.
Task Specific Reasoning Guideline
Make a list of ideas for understanding hourly wage and overtime rule and apply them one by one to accurately calculate wages.
Reasoning Structure:
1. Understand the hourly wage and overtime rules.
2. Determine the number of shifts.
3. Calculate regular hours per day.
4. Calculate overtime hours per day.
5. Calculate total daily earnings.
6. Calculate total earnings for 5 days.
Candidate Reasoning Solution:
1. Tina's regular hourly wage is $18.00 per hour.
2. Tina works 10 hours every day, which exceeds the 8-hour limit, so she works 2 shifts (10 hours / 8 hours per shift).
3. For each day, Tina works 8 regular hours. That's 8 * $18.00 = $144.00 per day.
4. For each day, she works an additional 2 overtime hours. That's 2 × ($18.00 + $9.00) = $54.00 per day.
5. Adding regular earnings and overtime earnings, we get $144.00 (regular) + $54.00 (overtime) = $198.00 per day.
6. To calculate total earnings for 5 days, we multiply the daily earnings by the number of days: $198.00 (daily earnings) × $5 (days) = $990.00. So, Tina makes $990.00 for working 10 hours every day for 5 days.
$(r_{i,j}^{path}, \hat{a}_{i,j})$
Filter by ground truth label or majority vote
Final Reasoning Solution:
Reasoning Solution: 1.Calculate the base wage earnings for the first 8 * 5 hours: 40 hours * $18.00 = $720. 2. Calculate the overtime hours: 10 * 5 hours - 40 hours = 10 hours. 3. Calculate the overtime earnings for the 10 overtime hours: 10 hours * $27.00 = $270. 4. Add the base wage earnings and overtime earnings to find Tina's total earnings: $720 (base wage earnings) + $270 (overtime earnings) = $990. Therefore, Tina makes $990 for working 10 hours every day for 5 days.

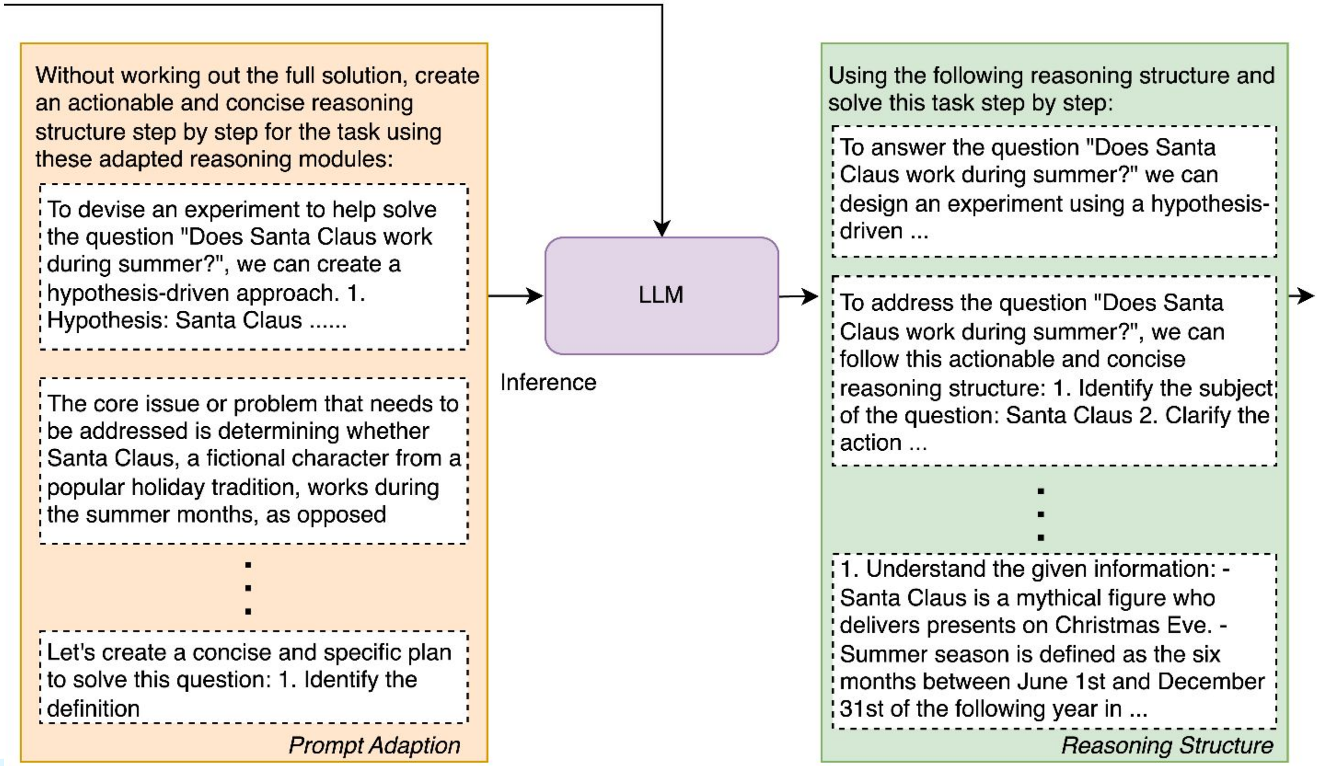
Pipeline — Prompt Adaption

For each instruction, we use the language model M to rephrase each seed prompt, tailoring it more specifically to the task.



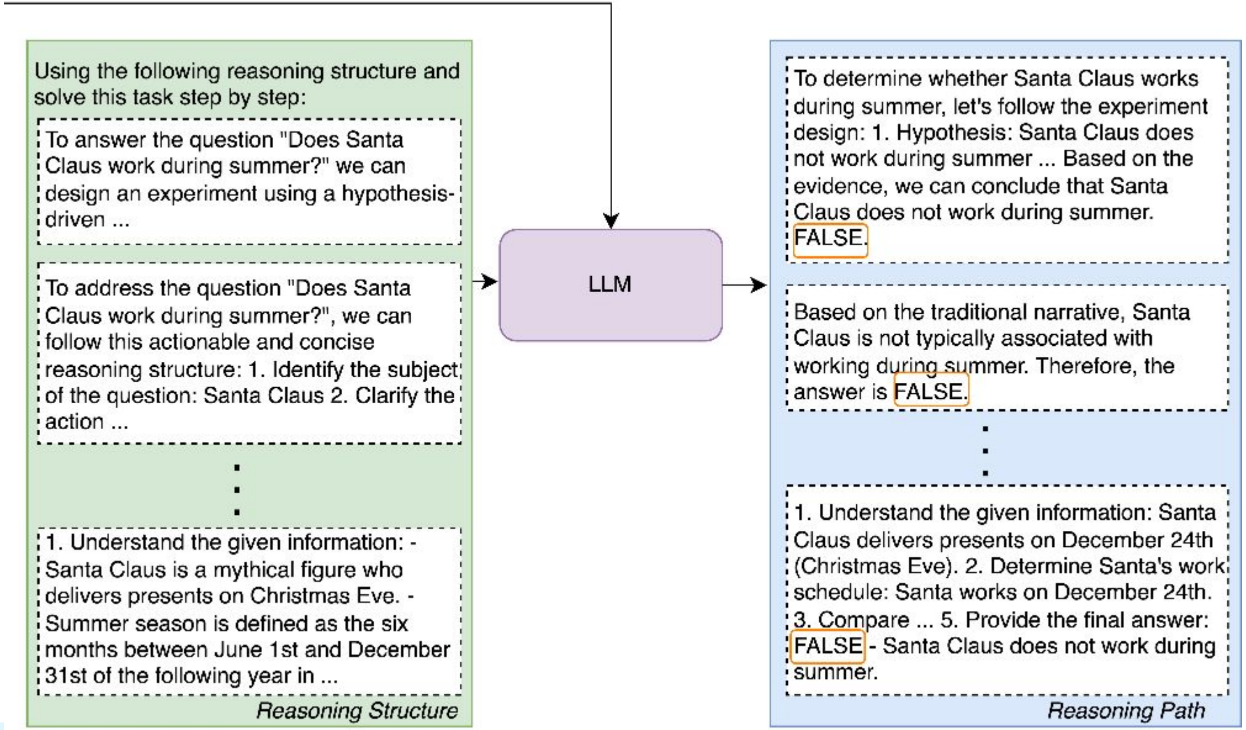
Pipeline — Generating Reasoning Structures

Following the similar reasoning process, we prompt the same language model M to develop a detailed reasoning structures for each instruction, utilizing the adapted prompts accordingly.



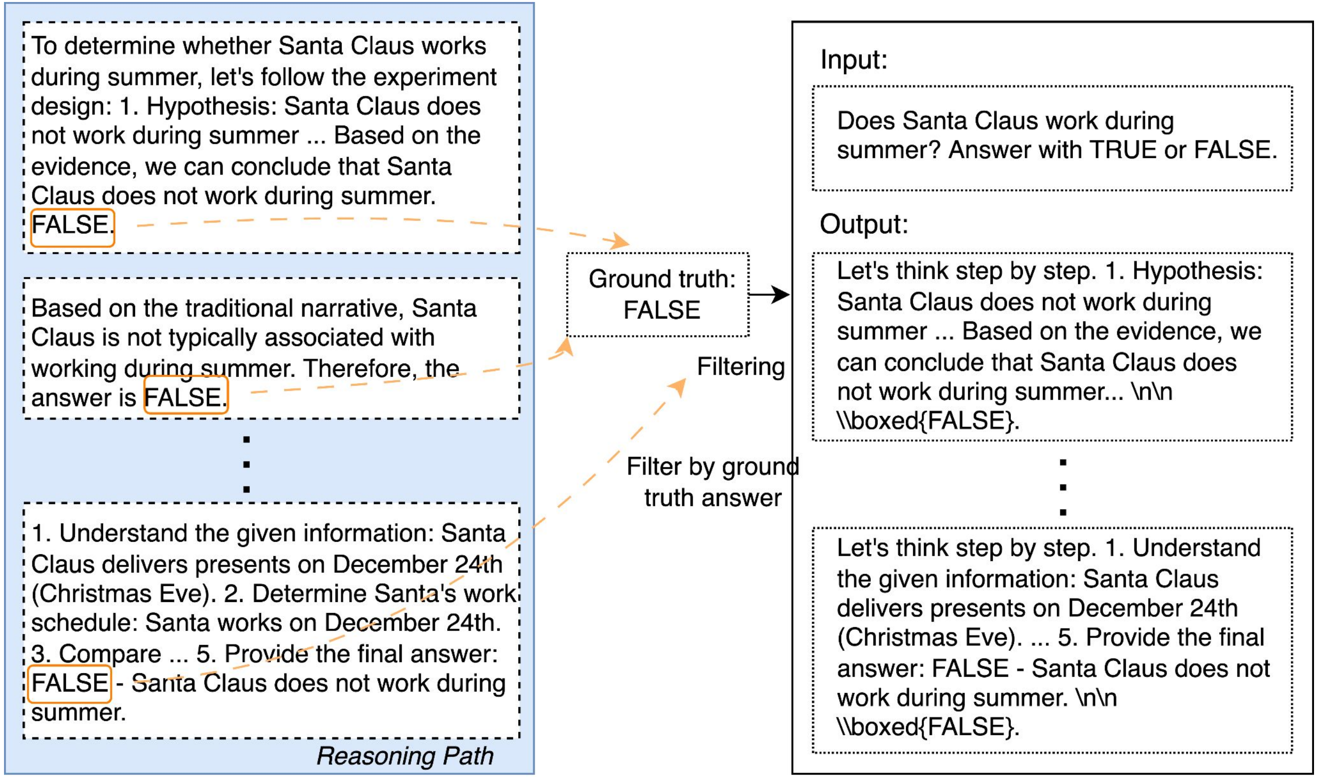
Pipeline — Reasoning Paths Generation

Each reasoning structure is used to create a detailed reasoning solution that includes a reasoning path and a corresponding solution corresponding to each seed prompt for each instruction.



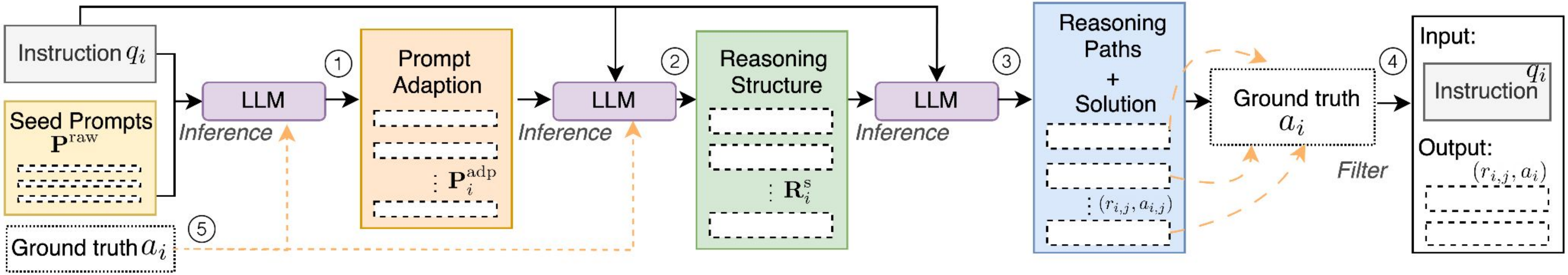
Pipeline — Filtering with the Ground-truth/Majority Vote

We use the ground-truth solutions to filter out those reasoning solutions that are incorrect.



Pipeline — Adding Ground-truth as a hint to generate the Reasoning Structure

For instructions that do not have correct reasoning solutions after our initial reasoning path generation, we include the ground-truth as a hint to generate adapted prompts and reasoning structures.



Example

Question: Tina makes \$18.00 an hour. If she works more than 8 hours per shift, she is eligible for overtime, which is paid by your hourly wage $+1/2$ your hourly wage. If she works 10 hours every day for 5 days, how much money does she make?

Seed Prompt: Make a list of ideas for solving this problem, and apply them one by one to the problem to see if any progress can be made.

Adapted Prompt: Make a list of ideas for understanding hourly wage and overtime rule and apply them one by one to accurately calculate wages.

Reasoning Structure: 1. Understand the hourly wage and overtime rules. 2. Determine the number of shifts. 3. Calculate regular hours per day. 4. Calculate overtime hours per day. 5. Calculate total daily earnings. 6. Calculate total earnings for 5 days.

Reasoning Path: 1. Tina's regular hourly wage is \$18.00 per hour. 2. Tina works 10 hours every day, which exceeds the 8-hour limit, so she works 2 shifts (10 hours / 8 hours per shift).b3. For each day, Tina works 8 regular hours. That's 8 * \$18.00 = \$144.00 per day. 4. For each day, she works an additional 2 overtime hours. That's 2 × (\$18.00 + \$9.00) = \$54.00 per day. 5. Adding regular earnings and overtime earnings, we get \$144.00 (regular) +\$54.00 (overtime) = \$198.00 per day. 6. To calculate total earnings for 5 days, we multiply the daily earnings by the number of days: \$198.00(daily earnings) × 5 (days) = \$990.00. So, Tina makes **\$990.00** for working 10 hours every day for 5 days.

Research Questions

For any existing datasets, we propose a **general reasoning path generation pipeline** and we want to explore:

(RQ1) How can we enhance the model's performance on **in-domain** tasks using reasoning paths?

(RQ2) How can we simultaneously improve the model's generalization performance on **out-of-domain** tasks?

Experiments — Baselines

1. Mistral-7B-Instruct-v0.3 (**w/o FT**). **No further fine-tuning** was performed in our experiments.

2. Fine-Tuning with Ground Truth (FT w/ GT). Fine-tune the Mistral-7B-Instruct-v0.3 model directly using the **ground-truth answers**, without incorporating any reasoning paths.

3. Fine-Tuning with Self-Improvement approach (**LMSI**). Following the methods of Huang et al. (2023), we first sample 32 reasoning paths using **few-shot CoT prompting**. We then filter these reasoning paths by selecting the one with the **majority vote** answer. Finally, we fine-tune Mistral-7B-Instruct-v0.3 using these self-generated solutions as the target outputs.

4. Fine-Tuning with Self-Improvement approach and ground truth (LMSI w/ GT). Same reasoning generation process as LMSI. However, instead of using a majority vote to filter out incorrect reasoning paths, we utilize **ground truth data**.

5. Fine-Tuning with Self-Taught Reasoner approach (**STaR**). Following the methodology outlined by Zelikman et al. (2022), we first sample 32 reasoning paths using few-shot CoT prompting. Next, we filter these paths based on ground truth and incorporate this ground truth as hints in the questions to generate reasoning paths for questions lacking correct solutions.

In Domain

Table 2: Comparison of zero-shot accuracy between fine-tuned and non-fine-tuned Mistral-7B-Instruct-v0.3 models using different prompting methods. All fine-tuned models are trained on a single training set from one dataset and evaluated on the corresponding test set across 5 math, logical and commonsense reasoning datasets.

Training Methods	Prompting Method at Inference	GSM8K (Math)	NumGlue (Math)	ARC-c (Logical)	ReClor (Logical)	StrategyQA (Commonsense)
w/o FT	CoT Prompting	44.0%	40.6%	77.2%	57.6%	77.4%
	CoT Prompting (3-shot)	68.3%	47.2%	79.1%	59.4%	80.8%
	Self-Consistency	60.0%	38.2%	80.6%	56.2%	80.8%
FT w/ GT	CoT Prompting	13.8%	55.0%	77.4%	70.4%	85.6%
	Self-Consistency	15.2%	55.9%	77.2%	**71.6%**	85.6%
LMSI	CoT Prompting	51.8%	46.5%	67.9%	51.8%	78.3%
	Self-Consistency	62.3%	57.1%	71.7%	50.8%	79.0%
LMSI w/ GT	CoT Prompting	57.4%	51.8%	75.9%	58.0%	80.2%
	Self-Consistency	66.3%	62.2%	77.5%	59.4%	81.7%
STaR	CoT Prompting	46.3%	48.3%	76.5%	57.8%	84.4%
	Self-Consistency	66.0%	64.5%	84.1%	63.8%	85.9%
ReGenesis	CoT Prompting	63.6%	52.2%	78.0%	68.4%	81.5%
	Self-Consistency	**76.6%**	**74.7%**	**85.4%**	70.6%	**91.3%**

In Domain

Table 2: Comparison of zero-shot accuracy between fine-tuned and non-fine-tuned Mistral-7B-Instruct-v0.3 models using different prompting methods. All fine-tuned models are trained on a single training set from one dataset and evaluated on the corresponding test set across 5 math, logical and commonsense reasoning datasets.

Training Methods	Prompting Method at Inference	GSM8K (Math)	NumGlue (Math)	ARC-c (Logical)	ReClor (Logical)	StrategyQA (Commonsense)
`w/o FT`	CoT Prompting	44.0%	40.6%	77.2%	57.6%	77.4%
	CoT Prompting (3-shot)	68.3%	47.2%	79.1%	59.4%	80.8%
	Self-Consistency	60.0%	38.2%	80.6%	56.2%	80.8%
`FT w/ GT`	CoT Prompting	13.8%	55.0%	77.4%	70.4%	85.6%
	Self-Consistency	15.2%	55.9%	77.2%	**71.6%**	85.6%
`LMSI`	CoT Prompting	51.8%	46.5%	67.9%	51.8%	78.3%
	Self-Consistency	62.3%	57.1%	71.7%	50.8%	79.0%
`LMSI w/ GT`	CoT Prompting	57.4%	51.8%	75.9%	58.0%	80.2%
	Self-Consistency	66.3%	62.2%	77.5%	59.4%	81.7%
`STaR`	CoT Prompting	46.3%	48.3%	76.5%	57.8%	84.4%
	Self-Consistency	66.0%	64.5%	84.1%	63.8%	85.9%
`ReGenesis`	CoT Prompting	63.6%	52.2%	78.0%	68.4%	81.5%
	Self-Consistency	**76.6%**	**74.7%**	**85.4%**	70.6%	**91.3%**

In Domain

Table 2: Comparison of zero-shot accuracy between fine-tuned and non-fine-tuned Mistral-7B-Instruct-v0.3 models using different prompting methods. All fine-tuned models are trained on a single training set from one dataset and evaluated on the corresponding test set across 5 math, logical and commonsense reasoning datasets.

Training Methods	Prompting Method at Inference	GSM8K (Math)	NumGlue (Math)	ARC-c (Logical)	ReClor (Logical)	StrategyQA (Commonsense)
w/o FT	CoT Prompting	44.0%	40.6%	77.2%	57.6%	77.4%
	CoT Prompting (3-shot)	68.3%	47.2%	79.1%	59.4%	80.8%
	Self-Consistency	60.0%	38.2%	80.6%	56.2%	80.8%
FT w/ GT	CoT Prompting	13.8%	55.0%	77.4%	70.4%	85.6%
	Self-Consistency	15.2%	55.9%	77.2%	**71.6%**	85.6%
LMSI	CoT Prompting	51.8%	46.5%	67.9%	51.8%	78.3%
	Self-Consistency	62.3%	57.1%	71.7%	50.8%	79.0%
LMSI w/ GT	CoT Prompting	57.4%	51.8%	75.9%	58.0%	80.2%
	Self-Consistency	66.3%	62.2%	77.5%	59.4%	81.7%
STaR	CoT Prompting	46.3%	48.3%	76.5%	57.8%	84.4%
	Self-Consistency	66.0%	64.5%	84.1%	63.8%	85.9%
ReGenesis	CoT Prompting	63.6%	52.2%	78.0%	68.4%	81.5%
	Self-Consistency	**76.6%**	**74.7%**	**85.4%**	70.6%	**91.3%**

Out of Domain

In this experiment, we assess the fine-tuned language models from Experiment on six out-of-domain (OOD) tasks. Our objective is to determine whether fine-tuning with or without reasoning paths influences the models' generalization capabilities.

Table 3: Zero-shot accuracy comparison between non-fine-tuned Mistral-7B-Instruct-v0.3 model and the models finetuned on one of five in-domain datasets separately and evaluated using the "Self-Consistency" prompting method across six out-of-domain tasks.

Training Datasets	Training Methods	Test Datsets					
		ASDIV (Math)	SVAMP (Math)	AQUA (Math)	BBH (Logical)	ANLI (NLI)	OpenbookQA (Commonsense)
GSM8K (Math)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	54.0% ↓	40.0% ↓	29.1% ↓	53.6% ↓	44.6% ↑	72.6% ↓
	LMSI w/ GT	77.3% ↑	72.2% ↓	31.1% ↓	59.5% ↓	43.4% ↑	73.2% ↓
	STaR	79.6% ↑	71.5% ↓	46.9% ↑	47.4% ↓	45.0% ↑	72.8% ↓
	Ours	**81.2%** ↑	**83.9%** ↑	**48.8%** ↑	**69.3%** ↑	**49.5%** ↑	**81.4%** ↑
NumGLUE (Math)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	53.8% ↓	54.9% ↓	32.3% ↓	42.4% ↓	37.0% ↓	64.8% ↓
	LMSI w/ GT	75.7% ↓	78.2% ↑	40.6% ↓	59.5% ↓	35.1% ↓	72.6% ↓
	STaR	**79.6%** ↑	76.1% ↑	37.0% ↓	58.5% ↓	41.9% ↑	71.6% ↓
	Ours	76.9% ↓	**79.4%** ↑	**48.4%** ↑	**61.7%** ↑	**50.0%** ↑	**79.8%** ↑
ReClor (Logical)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	62.0% ↓	56.0% ↓	22.0% ↓	44.6% ↓	47.1% ↑	71.4% ↓
	LMSI w/ GT	77.9% ↑	76.8% ↑	45.6% ↑	53.9% ↓	35.1% ↓	72.4% ↓
	STaR	76.1% ↓	74.6% ↓	46.1% ↑	60.6% ↓	42.5% ↑	77.2% ↑
	Ours	**76.4%** ↓	**76.5%** ↑	**49.6%** ↑	**66.8%** ↑	**44.8%** ↑	**81.4%** ↑
ARC-c (Logical)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	57.7% ↓	57.2% ↓	18.9% ↓	37.6% ↓	36.3% ↓	78.4% ↑
	LMSI w/ GT	70.9% ↓	72.0% ↓	32.7% ↓	60.8% –	32.5% ↓	79.4% ↑
	STaR	77.0% ↓	76.2% ↑	40.6% ↓	60.7% ↓	**47.4%** ↑	**84.2%** ↑
	Ours	**81.6%** ↑	**79.5%** ↑	**46.5%** ↑	**66.0%** ↑	46.4% ↑	82.8% ↑
StrategyQA (Commonsense)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	69.4% ↓	72.3% ↓	43.7% ↑	52.3% ↓	44.9% ↑	62.2% ↓
	LMSI w/ GT	56.3% ↓	56.8% ↓	40.6% ↓	60.6% ↓	39.8% ↑	68.4% ↓
	STaR	79.8% ↑	76.2% ↑	**43.3%** ↑	62.9% ↑	37.3% ↓	77.4% ↑
	Ours	**81.3%** ↑	**81.1%** ↑	42.9% ↑	**65.9%** ↑	**55.3%** ↑	**80.4%** ↑

Out of Domain

In this experiment, we assess the fine-tuned language models from Experiment on six out-of-domain (OOD) tasks. Our objective is to determine whether fine-tuning with or without reasoning paths influences the models' generalization capabilities.

Table 3: Zero-shot accuracy comparison between non-fine-tuned Mistral-7B-Instruct-v0.3 model and the models finetuned on one of five in-domain datasets separately and evaluated using the "Self-Consistency" prompting method across six out-of-domain tasks.

Training Datasets	Training Methods	Test Datsets					
		ASDIV (Math)	SVAMP (Math)	AQUA (Math)	BBH (Logical)	ANLI (NLI)	OpenbookQA (Commonsense)
GSM8K (Math)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	54.0% ↓	40.0% ↓	29.1% ↓	53.6% ↓	44.6% ↑	72.6% ↓
	LMSI w/ GT	77.3% ↑	72.2% ↓	31.1% ↓	59.5% ↓	43.4% ↑	73.2% ↓
	STaR	79.6% ↑	71.5% ↓	46.9% ↑	47.4% ↓	45.0% ↑	72.8% ↓
	Ours	**81.2%** ↑	**83.9%** ↑	**48.8%** ↑	**69.3%** ↑	**49.5%** ↑	**81.4%** ↑
NumGLUE (Math)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	53.8% ↓	54.9% ↓	32.3% ↓	42.4% ↓	37.0% ↓	64.8% ↓
	LMSI w/ GT	75.7% ↓	78.2% ↑	40.6% ↓	59.5% ↓	35.1% ↓	72.6% ↓
	STaR	**79.6%** ↑	76.1% ↑	37.0% ↓	58.5% ↓	41.9% ↑	71.6% ↓
	Ours	76.9% ↓	**79.4%** ↑	**48.4%** ↑	**61.7%** ↑	**50.0%** ↑	**79.8%** ↑
ReClor (Logical)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	62.0% ↓	56.0% ↓	22.0% ↓	44.6% ↓	47.1% ↑	71.4% ↓
	LMSI w/ GT	77.9% ↑	76.8% ↑	45.6% ↑	53.9% ↓	35.1% ↓	72.4% ↓
	STaR	76.1% ↓	74.6% ↓	46.1% ↑	60.6% ↓	42.5% ↑	77.2% ↑
	Ours	**76.4%** ↓	**76.5%** ↑	**49.6%** ↑	**66.8%** ↑	**44.8%** ↑	**81.4%** ↑
ARC-c (Logical)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	57.7% ↓	57.2% ↓	18.9% ↓	37.6% ↓	36.3% ↓	78.4% ↑
	LMSI w/ GT	70.9% ↓	72.0% ↓	32.7% ↓	60.8% –	32.5% ↓	79.4% ↑
	STaR	77.0% ↓	76.2% ↑	40.6% ↓	60.7% ↓	**47.4%** ↑	**84.2%** ↑
	Ours	**81.6%** ↑	**79.5%** ↑	**46.5%** ↑	**66.0%** ↑	46.4% ↑	82.8% ↑
StrategyQA (Commonsense)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	69.4% ↓	72.3% ↓	43.7% ↑	52.3% ↓	44.9% ↑	62.2% ↓
	LMSI w/ GT	56.3% ↓	56.8% ↓	40.6% ↓	60.6% ↓	39.8% ↑	68.4% ↓
	STaR	79.8% ↑	76.2% ↑	**43.3%** ↑	62.9% ↑	37.3% ↓	77.4% ↑
	Ours	**81.3%** ↑	**81.1%** ↑	42.9% ↑	**65.9%** ↑	**55.3%** ↑	**80.4%** ↑

Out of Domain

The results show the effectiveness of ReGenesis in enhancing the general reasoning capabilities of LLMs, enabling them to evolve into reasoning generalists through self-improvement.

Table 3: Zero-shot accuracy comparison between non-fine-tuned Mistral-7B-Instruct-v0.3 model and the models finetuned on one of five in-domain datasets separately and evaluated using the "Self-Consistency" prompting method across six out-of-domain tasks.

Training Datasets	Training Methods	Test Datsets					
		ASDIV (Math)	SVAMP (Math)	AQUA (Math)	BBH (Logical)	ANLI (NLI)	OpenbookQA (Commonsense)
GSM8K (Math)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	54.0% ↓	40.0% ↓	29.1% ↓	53.6% ↓	44.6% ↑	72.6% ↓
	LMSI w/ GT	77.3% ↑	72.2% ↓	31.1% ↓	59.5% ↓	43.4% ↑	73.2% ↓
	STaR	79.6% ↑	71.5% ↓	46.9% ↑	47.4% ↓	45.0% ↑	72.8% ↓
	Ours	**81.2%** ↑	**83.9%** ↑	**48.8%** ↑	**69.3%** ↑	**49.5%** ↑	**81.4%** ↑
NumGLUE (Math)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	53.8% ↓	54.9% ↓	32.3% ↓	42.4% ↓	37.0% ↓	64.8% ↓
	LMSI w/ GT	75.7% ↓	78.2% ↑	40.6% ↓	59.5% ↓	35.1% ↓	72.6% ↓
	STaR	**79.6%** ↑	76.1% ↑	37.0% ↓	58.5% ↓	41.9% ↑	71.6% ↓
	Ours	76.9% ↓	**79.4%** ↑	**48.4%** ↑	**61.7%** ↑	**50.0%** ↑	**79.8%** ↑
ReClor (Logical)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	62.0% ↓	56.0% ↓	22.0% ↓	44.6% ↓	47.1% ↑	71.4% ↓
	LMSI w/ GT	77.9% ↑	76.8% ↑	45.6% ↑	53.9% ↓	35.1% ↓	72.4% ↓
	STaR	76.1% ↓	74.6% ↓	46.1% ↑	60.6% ↓	42.5% ↑	77.2% ↑
	Ours	**76.4%** ↓	**76.5%** ↑	**49.6%** ↑	**66.8%** ↑	**44.8%** ↑	**81.4%** ↑
ARC-c (Logical)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	57.7% ↓	57.2% ↓	18.9% ↓	37.6% ↓	36.3% ↓	78.4% ↑
	LMSI w/ GT	70.9% ↓	72.0% ↓	32.7% ↓	60.8% –	32.5% ↓	79.4% ↑
	STaR	77.0% ↓	76.2% ↑	40.6% ↓	60.7% ↓	**47.4%** ↑	**84.2%** ↑
	Ours	**81.6%** ↑	**79.5%** ↑	**46.5%** ↑	**66.0%** ↑	46.4% ↑	82.8% ↑
StrategyQA (Commonsense)	w/o FT	77.2% –	75.4% –	41.3% –	60.8% –	38.4% –	75.6% –
	FT w/ GT	69.4% ↓	72.3% ↓	43.7% ↑	52.3% ↓	44.9% ↑	62.2% ↓
	LMSI w/ GT	56.3% ↓	56.8% ↓	40.6% ↓	60.6% ↓	39.8% ↑	68.4% ↓
	STaR	79.8% ↑	76.2% ↑	**43.3%** ↑	62.9% ↑	37.3% ↓	77.4% ↑
	Ours	**81.3%** ↑	**81.1%** ↑	42.9% ↑	**65.9%** ↑	**55.3%** ↑	**80.4%** ↑

Experiments — Performance without ground-truth labels

Table 5: Zero-shot accuracy of models fine-tuned using majority-vote filtering on GSM8K (Cobbe et al., 2021) and StrategyQA (Geva et al., 2021), tested on their respective datasets with various prompting methods.

Training Method	GSM8K (Math)	StrategyQA (Commonsense)
`w/o FT`	56.3%	80.8%
`LMSI`	62.3%	79.0%
`ReGenesis` (Majority)	**62.8%**	**83.1%**

Experiments — DIVERSE PREFERENCES FOR REASONING GUIDELINES

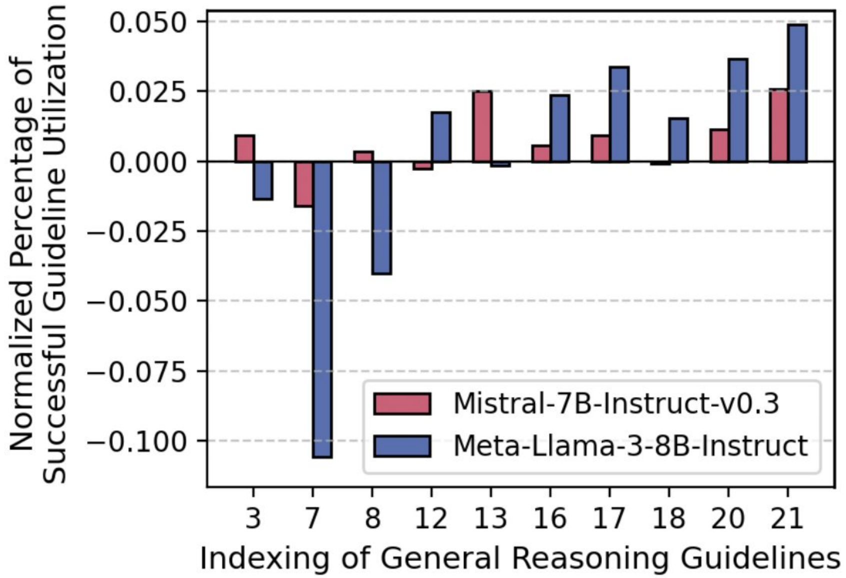


Figure 2: Normalized percentage of successful guideline utilization for selective general reasoning guidelines on the NumGLUE dataset, comparing Mistral-7B-Instruct-v0.3 and Meta-Llama-3-8B-Instruct models.

Experiments — Case Study: PRESERVATION OF TASK-AGNOSTIC REASONING IN TASK-SPECIFIC PATHS

salesforce

STAR

Cluster 1: Direct Calculation and Simplification
Cluster 2: Algebraic and Formula-based Approaches
Cluster 3: Stoichiometry and Chemical Reactions
Cluster 4: Problem Decomposition and Step-by-Step Calculation –
Cluster 5: Logical Reasoning and Pattern Recognition

ReGenesis

Cluster 1: Step-by-Step Process –
Cluster 2: Applying Logic and Formulas
Cluster 3: Reflective Thinking
Cluster 4: Sequential and Chronological Analysis
Cluster 5: Information Extraction –
Cluster 6: Problem Decomposition
Cluster 7: Systematic Thinking
Cluster 8: Questioning Assumptions and Critical Thinking
Cluster 9: Mathematical and Analytical Calculation
Cluster 10: Stoichiometry and Chemical Problem Solving

ICLR

Takeaway

1. We introduced ReGenesis, a novel framework designed to self-synthesize reasoning paths as post- training data to self-improve their general reasoning capacities without requiring additional supervision beyond final answers and human-designed reasoning examples.
2. This framework effectively self-synthesizes reasoning paths of any datasets, **regardless of whether they include ground-truth answers**.
3. Fine-tuning the language model with such self-synthesized dataset leads to significant improvements in performance on both **in-domain and out-of-domain** tasks.


Q
A