No longer left behind: Self-training Reasoning Models in Italian

Federico Ranaldi[1,2], Leonardo Ranaldi[1,2]

[1]Univeristy of Roma Tor Vergata
[2]Univeristy of Edinburgh

Abstract

Although reasoning is, by nature, language-agnostic, the extent to which large language models (LLMs) can perform consistent multilingual reasoning remains limited. Their capacity to deliver step-wise explanations is largely constrained to the dominant languages present in their pre-training data, thereby limiting cross-lingual generalisation and hindering broader global applicability. While recent work has explored a range of strategies to extend reasoning capabilities beyond English, these efforts typically remain grounded in surface-level spoken language phenomena, which may not be optimal for abstract or formal reasoning tasks. In this study, we focus on Italian and English, two languages with markedly different syntactic and morphological properties, to assess whether advancements in multilingual reasoning remain consistent and transferable across typologically diverse settings. To this end, we introduce a modular framework that guides LLMs to abstract the reasoning process into a structured problem space before generating step-wise reasoning trajectories. The approach leverages self-training to enhance alignment and generalisation. Experimental results demonstrate stable and significant gains in multilingual reasoning across models and tasks, with improved consistency between English and Italian.

Keywords

Multilingual Reasoning, Self-training, Large Reasoning Models

1. Introduction

In the era of large language models (LLMs), approaches such as Chain-of-Thought (CoT) and related methods seek to emulate human reasoning through language generation—an ability that, in principle, ought not to be constrained by the particularities of any spoken language. Yet, a growing body of evidence indicates that the reasoning capabilities of LLMs vary significantly across languages, largely as a consequence of imbalances in pre-training data. LLMs perform better in dominant languages, notably English, while exhibiting reduced reasoning competence in less-represented languages.

Research advances in multilingual reasoning are increasingly aimed at closing the performance differences among languages, enhancing the models' capabilities through in-context learning interventions [1, 2, 3], SFT strategies that differ from language-specific augmentation [4, 5] to task-oriented tuning [6], and preference optimisation [7, 8]. Although these approaches have enabled the development of effective methods for transferring and aligning multilingual reasoning capabilities, we argue that several critical challenges continue to hinder progress. First and foremost, the benefits of in-context interventions appear to be confined to large-scale LLMs, which are better equipped to interpret and follow instructions in a systematic way. However, they must also have robust multilingual proficiency. Therefore, many works rely on SFT techniques that maintain reduced costs when used with specialised, smaller-scale LLMs. Secondly, they require vast amounts of complex reasoning annotations and tremendous tuning efforts to get multilingual LLMs capable of delivering reasoning through SFT and preference optimisation techniques.

To enhance multilingual reasoning in LLMs, we propose a modular approach that first instructs the model to abstractly formalise the problem and then generate structured, step-by-step reasoning trajectories that converge towards a consistent reasoning process across languages.

Our approach decomposes problem solutions into a sequence of formal, language-agnostic sub-problems that are solved sequentially and can be more effectively utilised by models.

The decomposition consists of two high-level modules: *Formalisation* and *Reasoning Execution*. As illustrated in Figure 1, we guide the models to: (i) identify the relevant information within the problem, formalising variables and predicates while delivering symbolic transformations; (ii) generate a reasoning execution trajectory in which the transformations are applied using symbolic representations that explicitly articulate the solution, ultimately yielding an answer in the same query language.

Previous works proposed English-based strategies that operate via logical formalisms coupled with external symbolic solvers [9, 10]. Yet, fully symbolic approaches face a key bottleneck: they require a complete translation

CLiC-it 2025: Eleventh Italian Conference on Computational Linguistics, September 24 — 26, 2025, Cagliari, Italy
name.surname@uniroma2.it (F. Ranaldi);
name.surname@ed.ac.uk (L. Ranaldi)

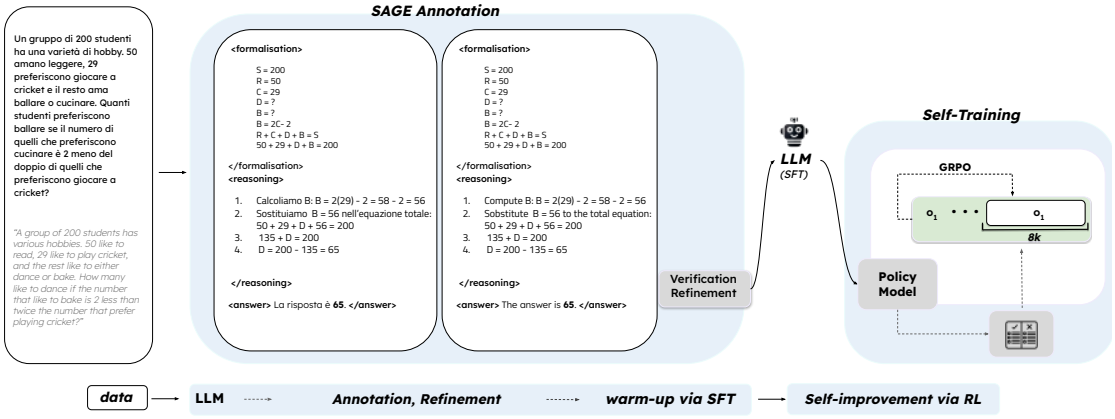

Figure 1: LLMs deliver language-agnostic reasoning trajectories across languages by disentangling content from logical reasoning through structured step-wise passages operating via our Structured Abstractive Generative Explanation.

from natural to formal language, which can hinder both efficiency and flexibility, introducing additional layers of complexity.

To achieve a better trade-off, we treat formalisations in an eclectic manner and propose methods to disentangle content from logical reasoning without introducing rigorous formalisms.

To this end, following Ranaldi and Pucci [11], we instruct larger LLMs to generate synthetic demonstrations through Structured Abstractive Generative Explanation (SAGE), which are then used to perform *Self-training* on smaller LLMs.

As part of the warm-up phase, we experiment with multiple alignment strategies, ranging from supervised fine-tuning (Instruction-Tuning) to preference optimisation techniques (Reinforcement Learning).

We conducted an extensive empirical evaluation to assess the impact of different tuning and alignment strategies.

In multilingual reasoning tasks, our demonstrated significant improvements, resulting in an overall increase in exact matching in proposed tasks, which led to the following results and conclusions:

- Structuring multilingual reasoning in LLMs as formal reasoning trajectories (SAGE), which leverages language-agnostic reasoning logic, improves accuracy and generates more verifiable outputs through a transparent and structured.
- Leveraging *self-training* heuristics that combine both tuning and preference optimisation leads to more robust, generalisable, and language-aligned models. While tuning based on synthetic demonstrations proves effective, it alone fails to yield consistently strong performance across all languages. Conversely, relying solely on preference optimisation can provide performance gains, but at the cost of significant computational overhead.
- Our approach allows the disentanglement of content from logical reasoning, improving multilingual reasoning in LLMs, thus benefiting in different language spaces.

2. Method

We propose a self-training framework that augments standard fine-tuning with a set of preference optimisation policies (§ 2.1) designed to improve self-refinment. The approach iteratively alternates between preference-based optimisation (via reinforcement learning) and supervised fine-tuning, directing the model to abstract the underlying problem and articulate a step-wise, formal solution (§ 2.2). The iterative process terminates once the model's performance either converges or reaches a predefined maximum number of iterations.

2.1. Preference Estimation

RL strategies operate preference estimation. This generally involves aligning the policy model with preferences using a reward model, which learns to predict preferences based on comparisons and leads the optimisation process. Although this approach is practical, it has problems with generalisation, scalability, robustness, and alignment. In GRPO, rule-based reward models are used. While DPO is generally based on a series of naive string-matching functions with ground truth values, rules are explicitly

defined in GRPO. Accordingly, we define the following preference policies:

DPO Preference Estimation We adopt a string-matching function in line with existing approaches for English [8, 12]. We then refine this procedure by filtering out generations that do not adhere to the expected structural pattern and well-formed format.

GRPO Preference Estimation Following Ranaldi and Pucci [11] we define a rule-based metrics that control the accuracy, the structure and the form of the generations.

2.2. Self-training

Conventional self-training begins by fine-tuning the base model $\mathcal{M}\theta$ on the supervised dataset $\mathcal{D}$SFT, yielding an updated model $\mathcal{M}\theta'$. At this stage, we assume that $\mathcal{M}\theta'$ has acquired the ability to address the target problem. Specifically, when presented with a question x, the model generates a formal reasoning sequence $\hat{y}$ together with the corresponding answer $\hat{a}$.

Self-training We begin by sampling multiple completions $\hat{y}$ from $\mathcal{M}\theta'$ in response to a set of questions x drawn from the unlabelled pool $\mathcal{U}$. We then apply preference estimation heuristics to construct preference-based samples according to different optimisation strategies: pairwise comparisons for DPO and grouped completions for GRPO. These generations are compiled into a dataset $\mathcal{D}$, which is subsequently used to further train the model using the corresponding objective functions ($\mathcal{L}$DPO and $\mathcal{L}$GRPO), resulting in an updated model $\mathcal{M}\theta^d$.

Then we use $\mathcal{M}_{\theta^d}$ to generate a new pseudo-labeled dataset for the next-round tuning:

$$\mathcal{S} = (x, \hat{y}) | x \sim \mathcal{U}, \hat{y} \sim_\theta (\cdot|x). \tag{1}$$

After generation, the dataset $\mathcal{S}$ is refined by removing incorrect answers and eliminating duplicates. Consequently, the resulting pseudo-labeled dataset, denoted as $\mathcal{S}^\alpha$, is a subset of the original dataset, i.e., $\mathcal{S}^\alpha \subset \mathcal{S}$. The final training dataset is constructed by combining the original labeled dataset $\mathcal{L}$ with the newly generated pseudo-labeled dataset $\mathcal{S}^\alpha$. During this process, each new dataset is used to train from the original base model $\mathcal{M}_\theta$, rather than continually fine-tuning $\mathcal{M}_\theta$, to mitigate the risk of overfitting.

2.3. Single-training

For comparative purposes, we conduct individual training operating only with SFT, DPO and GRPO.

Algorithm 1 Self-training [11]

Input: pre-trained language model $\mathcal{M}_\theta$
labeled dataset $\mathcal{L} = \{(x^i, y^i, a^i)\}_{i=1}^l$
unlabeled dataset $\mathcal{U} = \{(x^i, a^i)\}_{i=1}^u$
mode $\in$ {DPO, GRPO}
Output: fine-tuned model $\mathcal{M}_{\theta'}$

Warm-up stage
Fine-tune $\mathcal{M}_\theta$ on $\mathcal{L}$ to get $\mathcal{M}_{\theta'}$
repeat
  if mode = DPO **then**
  Generate DPO dataset $\mathcal{D}$:
  $\mathcal{D} = \{(\ x^i,\ y_w^i,\ y_l^i\ )\}_{i=1}^N$
  where $x^i \sim \mathcal{U}$ and $y_w^i,\ y_l^i \sim \mathcal{M}_{\theta'}(\cdot|x^i)$
  Tune $\mathcal{M}_{\theta'}$ with $\mathcal{L}_{\text{DPO}}$ on $\mathcal{D}$ to get $\mathcal{M}_{\theta^d}$
  end if
  if mode = GRPO **then**
  Generate GRPO dataset $\mathcal{G}$:
  $\mathcal{G} = \{(x^i, G^i)\}_{i=1}^N$
  where $x^i \sim \mathcal{U}$
  and $G^i = \{y_1, \ldots, y_k\} \sim \mathcal{M}_{\theta'}(\cdot|x^i)$
  Compute relative preferences within each group G^i,
  assign pairwise relative scores to outputs in G^i.
  Tune $\mathcal{M}_{\theta'}$ with $\mathcal{L}_{\text{GRPO}}$ on $\mathcal{G}$ to get $\mathcal{M}_{\theta^g}$
  end if
  # SFT step
  Build pseudo-labeled dataset $\mathcal{S}$:
  $\mathcal{S} = \{(x^i, \hat{y}^i, \hat{a}^i)\}_{i=1}^s$
  where $x^i \sim \mathcal{U}$ and $\hat{y}^i, \hat{a}^i \sim \mathcal{M}_{\theta^d}(\cdot|x^i)$
  $\mathcal{M}_{\theta^g}(\cdot|x^i)$
  Select $\mathcal{S}^\alpha \subset \mathcal{S}$ when $\hat{a}^i = a^i$
  Update $\mathcal{L} \leftarrow \mathcal{S}^\alpha \cup \mathcal{L}$
  Train $\mathcal{M}_\theta$ on $\mathcal{L}$ to get a new $\mathcal{M}_{\theta'}$
until convergence or max iteration is reached

3. Experiments

As outlined in the introduction, our objective is to develop a method for enhancing the reasoning capabilities of LLMs beyond English, with a particular emphasis on Italian. Our experiments are conducted on multilingual reasoning tasks. We evaluate four models (§ 3.1), trained according to the procedure detailed in § 3.2, on two mathematical reasoning benchmarks (§ 3.3), using the experimental configurations described in § 3.4.

3.1. Models

To conduct our study on different models and have a term of comparison, we use Llama3-8B [13], DeepSeekMath-7B-Instruct [14] (DeepSeek-7B). Furthermore, to show the scalability and effectiveness of our approach on further models, we introduce additional smaller-scale models: EuroLLM-1.7B and Velvet-2B.

3.2. Training Methods

As introduced in §2, we use a iterative steps of SFT and RL. We follow standard practice and perform a warm-up phase based on an SFT step using synthetic demonstrations discussed in §3.3.2. Then, we conduct the self-training by progressively applying SFT and RL optimisation algorithms. Following pilot studies (later discussed), we set the total number of iterations to three (excluding warm-up), the same for the settings where we use only one between SFT and RL.

Preference Optimisation RL We employ the HuggingFace trainers ($DPO_{trainer}$ and $GRPO_{trainer}$) to ensure reproducibility. For DPO, we set the learning rate to *1e-6* and β to 0.1. The optimisation process is set at a maximum of 2000 steps, saving the checkpoint corresponding to the lowest validation loss. For GRPO, we set the learning rate to *5e-6* and β to x. The optimisation process is set at a maximum of 2000 steps, saving the checkpoint corresponding to the lowest validation loss. Details in Appendix D.

Supervised Fine-tuning Regarding the SFT phase, we employed 8-bit quantization and LoRA. We tune the model for one epoch (warm-up) and for one epoch for each iteration using the learning rates according to the specific model configuration, as detailed in Appendix D.

3.3. Data

3.3.1. Evaluation Set

To study the reasoning performances of trained models, we operate via MGSM, MSVAMP, and we introduce MGSM-SYMBOLIC focusing on English and Italian.

Mathematical Reasoning task We use the extension of GSM8K and SVAMP. Respectively, Multilingual Grade School Math (MGSM) and Multilingual Simple Variations on Arithmetic Math word Problems (MSVAMP). In original cases, the authors proposed a benchmark of English mathematical problems with the following structure: a word problem in natural language and a target answer in numbers. For both versions, a subset of instances from the official list of examples were translated into 11 different languages, maintaining the structure of the input and output.

MGSM-SYMBOLIC Mirzadeh et al. [15] improved GSM8k (the ancestor of MGSM) by proposing GSM-Symbolic. This introduces symbolic patterns in GSM8k that complexify the task and disadvantage the LLMs' capabilities. We propose mGSM-Symbolic, the multilingual GSM-Symbolic extension. In particular, we conduct an automatic translation phase disillusioned by qualified annotators in 10 different languages. The dataset is available on **GitHub**[1] and **HuggingFace**[2].

3.3.2. Training Set

Instead of using natural language rationale, we employ synthetic demonstrations to train models to solve tasks following the two phases in Figure 1. Specifically, we instruct a robust model capable of addressing multilingual mathematical tasks by formalising problems and solving them in a language-agnostic manner. We employ GPT-4o as annotator, instructing it with the prompt detailed in Appendix A (we define this procedure as `Self-training`)

Different works train an expert version of the same model that is going to be refined for generating synthetic demonstrations, which are subsequently used for self-training (we define this procedure as `Full Self-training`).

Multilingual Demonstrations We annotate a subset of the MSVAMP dataset containing 250 samples for all languages to have in-domain demonstrations. After the annotation process, we check the quality of the demonstrations using rule-based heuristics and GPT-4o-mini as an additional evaluator (details in Appendix C).

3.4. Experimental Setup

In-context Learning We evaluate the baseline models (without tuning) using a 6-shot strategy defined as `Direct` and `CoT`. Moreover, we instruct the models to solve the problem following SAGE.

Training We assess the impact of the Self-training approaches (§3) by conducting different tuning configurations:

- **SFT, RL** We tune the models using the synthetic demonstrations as detailed in Appendix B.
- **Self-training** We warm-up the models using the synthetic demonstrations as detailed and conduct the self-training strategies using both policies.
- **FULL Self-training** Finally, to observe the impact of the self-generated demonstrations, we conduct both the annotation, SFT (warm-up) and FULL Self-train phase completely on the self-generated data of the same expert model.

[1] lranaldii/MGSM-Symbolic

[2] lrana/MGSM-Symbolic

4. Results

Reasoning can be effectively grounded in language-agnostic form, which LLMs can leverage to enhance multilingual task performance. SAGE facilitates this by guiding LLMs towards structured symbolic solutions, enabling them to produce robust and consistent outputs across languages. While SAGE yields strong results in GPT-4o, its benefits do not readily extend to smaller models. To address this, we adopt a self-training strategy that enables smaller models to acquire formal reasoning capabilities independently of explicit instruction, ultimately achieving greater consistency than GPT-4o (§ 4.1). Notably, self-training not only outperforms standalone SFT and reinforcement learning approaches, but also enables models to achieve stronger performance with substantially less training data (§ 4.2). Furthermore, we demonstrate the scalability of this method by successfully applying self-training to additional small-scale models (§ 4.3).

4.1. Language-Agnostic Reasoning

SAGE positively influences the models' performance in multilingual reasoning, getting substantial benefits on the proposed tasks.

Models	En	It
GPT-4o	83.2	79.0
+SAGE	93.0	88.6
Llama3-8B	76.0	58.2
+Self-training	91.8	73.0
DeepSeek-7B	76.2	58.2
+Self-training	90.2	76.9
Velvet-2B	60.2	56.8
+Self-training	71.0	68.5
EuroLLM-1.7B	66.3	60.4
+Self-training	72.6	65.8

Table 1
Performances on mGSM-Symbolic.

Multilingual Reasoning Table 1 presents results for SAGE with GPT-4o on mGSM-Symbolic, with a particular focus on English and Italian. The performance remains consistent with that observed in mGSM, as indicated by the values in brackets. Notably, the Self-training strategy enhances the models' abstraction capabilities, allowing them to perform well even in the more formal and structured setting of mGSM-Symbolic, where typical linguistic biases are reduced. In contrast, baseline methods yield substantially lower scores, underscoring the effectiveness of SAGE's formalisation in supporting multilingual reasoning.

In-context Learning Table 2 presents the performance of SAGE applied to GPT-4o, showing clear improvements over previous prompting-based strategies such as `Direct` and `CoT`. The use of in-context instructions encourages the model to organise problem-solving in a structured manner, promoting step-wise reasoning and planning. This results in more consistent reasoning trajectories that are less influenced by language-specific patterns, thereby reducing performance disparities across languages.

4.2. The Self-training Impact

Table 2 summarises the outcomes of applying the Self-training strategy across multiple models. The findings indicate a consistent enhancement in performance, particularly in terms of cross-linguistic consistency, even if overall accuracy remains below that of GPT-4o. Beyond accuracy, Self-training proves to be a more efficient tuning method, yielding stronger models while requiring significantly less training data than alternative approaches such as SFT and RL. This advantage is reflected in the steady performance gains observed over SFT in Table 2, and further supported by data efficiency metrics reported in Appendix F, where Self-training operates with fewer examples per model.

The role of RL Table 2 reports the results obtained using GRPO. As shown in Table 3, GRPO consistently outperforms DPO, both when applied in isolation and when integrated with SFT within the full Self-training framework. As outlined in Section 2.1, GRPO does not rely on an annotated dataset for supervision. Instead, similar to prior work, a rule-based algorithm serves as a proxy reward model. Unlike DPO, which operates at the level of individual instances, GRPO is specifically designed to optimise groups of completions across languages, making it well-suited to the multilingual nature of the proposed task.

The impact of Full Self-training Current alignment strategies typically rely on demonstrations produced by expert models belonging to the same model family. Ranaldi and Freitas [6] demonstrate that in-family learning exerts a stronger influence on the performance of student models. In our work, we adopt the Full Self-training approach and show that self-generated demonstrations can lead to more robust outcomes than those derived from GPT-4o. As illustrated in Figure 2, models trained with their own annotations exhibit greater consistency

Model	Method	MGSM		MSVAMP		Average	
		En	It	En	It	En	It
GPT-4o	Direct	86.8	79.8	83.2	74.6	85.0	77.2
	CoT	92.4	86.0	89.0	78.2	90.7	82.1
	SAGE	**93.0**	**88.4**	**86.2**	**83.6**	**89.6**	**86.0**
Llama-3-8B	Direct	79.6	61.2	81.2	69.8	80.4	65.5
	RL (GRPO)	84.0	70.4	83.6	70.0	83.8	70.2
	SFT	82.6	68.0	83.0	**72.6**	82.8	70.3
	Self-training	92.0	84.6	88.4	71.8	90.2	78.2
DeepSeek-7B	Direct	78.0	66.2	83.0	77.4	80.5	71.7
	RL (GRPO)	84.8	72.2	84.4	80.0	86.4	70.6
	SFT	82.0	70.0	80.6	80.4	81.3	75.2
	Self-training	**86.0**	**76.8**	**90.4**	**86.0**	88.2	81.8
Velvet-2B	Direct	58.0	55.4	60.6	55.0	59.3	55.2
	RL (GRPO)	66.8	62.2	62.4	56.8	64.6	59.5
	SFT	64.4	60.0	62.0	58.0	63.2	59.0
	Self-training	**70.4**	**72.0**	**70.8**	**62.4**	**70.6**	**66.3**
EuroLLM-1.7B	Direct	62.0	59.0	62.0	59.4	62.0	59.2
	RL (GRPO)	66.0	64.0	64.6	60.8	65.3	62.4
	SFT	64.4	60.2	**69.0**	62.0	66.7	61.1
	Self-training	**72.0**	**71.2**	68.4	64.8	**70.2**	**68.0**

Table 2
Accuracy scores using methods introduced in §2. We report the models trained via GRPO algorithm. *(in **bold** the best performance per model.

		MGSM	MSVAMP
Llama-3-8B	**RL**	+3.8	+3.2
	SFT+RL	**+8.4**	+3.6
DeepSeek-7B	**RL**	+5.2	+4.0
	SFT+RL	**+8.6**	+5.8
Velvet-2B	**RL**	+2.0	+2.6
	SFT+RL	+1.6	+1.8
EuroLLM-1.7B	**RL**	+2.2	+2.8
	SFT+RL	+2.4	+3.0

Table 3
Differences (Δ) between GRPO and DPO when used alone (RL) and in Self-training settings (SFT+RL). **Bold** indicates the highest observed gains.

Figure 2: Accuracy differences using data generated by GPT-4o and self-generated (i.e. FULL Self-training).

and resilience across languages, despite using the same amount of training data.

4.3. Transferability in Smaller Models

To evaluate the transferability of Self-training and SAGE to smaller-scale models, we extend our experiments to include `Llama-3-1B`, `EuroLLM-1.7B`, and `Velvet-2B`. These models were selected based on three criteria: their inherent multilingual design, their promising performance in mathematical reasoning tasks, and their relatively low parameter count, which enabled efficient experimentation across training regimes.

We adopt the experimental setup detailed in § 3.1, applying SFT, GRPO, and our full Self-training procedure. Table 3 reports the average results obtained on the MGSM-SYMBOLIC benchmark. Across all models, Self-training with SAGE consistently outperforms both SFT and RL-based baselines.

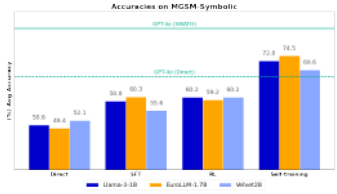

Figure 3: Average accuracies of smaller models in our MGSM-SYMBOLIC.

5. Background

5.1. Improving Reasoning in LLMs

Improving reasoning capabilities in LLMs (both English and multi- and cross-lingual) is usually conducted through SFT using ground-thought examples and preference-based approaches.

Supervised Fine-Tuning Supervised Fine-Tuning (SFT) is a standard approach for adapting a model $\mathcal{M}$ to reasoning tasks using a labelled dataset $\mathcal{L}$. Each instance in $\mathcal{L}$ consists of a question x, a corresponding step-by-step explanation y, and a final answer a. The answer is derived from the explanation using regular expressions. A generated rationale $\hat{y}$ is deemed valid if the extracted answer $\hat{a}$ matches the reference answer a. Formally, the labelled dataset with n instances is defined as:

$$\mathcal{L} = (x^i, y^i, a^i)i = 1^n. \tag{2}$$

SFT updates the parameters θ of model $\mathcal{M}\theta$ by minimising the negative log-likelihood of the target rationale:

$$\mathcal{L}\text{SFT}(\theta) = \mathbb{E}(x, y) \sim \mathcal{L}\left[\sum_{t=1}^{T} \log f_\theta(y_t|x, y_{1:t-1})\right], \tag{3}$$

where T is the length of the rationale y, and y_t denotes its t-th token.

Self-training Self-training refers to a family of SFT-based methods that have recently gained renewed interest for their effectiveness in enhancing reasoning capabilities [16]. These methods typically follow a two-stage process. First, a base model $\mathcal{M}\theta$ is fine-tuned on a labelled subset $\mathcal{L}$ to obtain a teacher model $\mathcal{M}\theta'$. This teacher is then used to annotate an unlabelled dataset $\mathcal{U}$, producing a pseudo-labelled dataset $\hat{\mathcal{L}}$. In the second stage, a student model $\mathcal{M}\theta$ is trained on the combination of the original data $\mathcal{L}$ and the pseudo-labelled data $\hat{\mathcal{L}}$, with the aim of surpassing the performance of the teacher $\mathcal{M}\theta'$.

Empirical studies have shown that the quality of pseudo-labels plays a critical role in determining the effectiveness of self-training. To address this, Wang et al. [12] propose an iterative refinement procedure, wherein the model $\mathcal{M}_\theta$ is progressively improved, ensuring increasingly accurate pseudo-labelled data across iterations.

Reinforcement Learning Heuristics (RL) Within the Self-training approaches, Reinforcement Learning from Human Feedback (RLHF) is widely used for aligning language models with human feedback [17]. The RLHF framework refines LLM behaviour by leveraging human preference data to guide model tuning through RL. Specifically, it uses a reward model $r(x, y)$, which captures human preferences given an input x and its corresponding output y. This reward model is then employed to assign preference scores to arbitrary LLM-generated outputs, facilitating iterative policy refinements via proximal policy optimisation (PPO) [18]. The training process follows an optimisation function, for instance, PPO, which optimises the model policy ϕ_θ to maximise expected rewards while minimising divergence from the SFT policy:

$$\mathbb{E}_{(x,y)\sim D_\pi}[r(x, y) - \gamma \log \frac{\phi_\theta(y|x)}{\phi_{\text{SFT}}(y|x)}], \tag{4}$$

where ϕ_{SFT} denotes the original model trained via SFT, and γ serves as a regularization hyperparameter to constrain policy updates.

Direct Preference Optimisation Reinforcement Learning with Human Feedback (RLHF), particularly through Proximal Policy Optimisation (PPO), has proven effective for aligning language models with human preferences. However, it typically requires multiple auxiliary components, including a reward model, making the training process computationally intensive and technically complex. To address this, Rafailov et al. [19] proposed Direct Preference Optimisation (DPO), which allows models to be aligned directly with human preferences without the need to train a separate reward model.

DPO begins with a warm-up phase based on supervised fine-tuning. For a given input x, the reference policy ϕ_{ref} generates two candidate completions:

$$y_1, y_2 \sim \phi_{\text{ref}}(\cdot \mid x). \tag{5}$$

These are then paired based on preference to form the DPO training set:

$$\mathcal{L}_{DPO} = (x^i,\ y_w^i,\ y^i l)i = 1^N, \tag{6}$$

where y_w^i is the preferred response and y_l^i is the less preferred one.

The policy model $\mathcal{M}\theta$ is then optimised by minimising the following objective:

$$\mathbb{E}\,(x, y_w, y_l) \sim \mathcal{D}\left[-\log \sigma\left(r(y_w|x) - r(y_l|x)\right)\right], \tag{7}$$

where the score function is defined as $r(\cdot|x) = \beta \log \frac{\phi_\theta(\cdot|x)}{\phi_{\text{ref}}(\cdot|x)}$, and the parameter β regulates how far the new policy ϕ_θ may deviate from the reference policy.

While DPO offers a more streamlined alternative to RLHF by avoiding explicit reward modelling, it is limited by its reliance on fixed pairwise preference comparisons. This can hinder its capacity to generalise across tasks that exhibit contextual or structural variation [20].

Group Relative Policy Optimisation To overcome these limitations, Shao et al. [21] introduced Group Relative Policy Optimisation (GRPO), a refinement of PPO that improves training stability by using group-based reward estimation. Instead of relying on pairwise comparisons, GRPO evaluates completions within groups and assigns rewards based on relative performance within those groups.

Given a batch of responses from the policy model ϕ_θ, GRPO estimates relative advantages across the group and applies the following optimisation objective:

$$\mathbb{E}(x, y) \sim D\left[A\text{rel}(y|x) \log \pi_\theta(y|x) - \beta D_{\text{KL}}\left(\pi_\theta | \pi_{\text{ref}}\right)\right], \tag{8}$$

where π_θ is the updated policy and π_{ref} is the original pre-trained policy. The KL divergence term prevents the updated policy from diverging excessively from its prior, with the coefficient β determining the strength of this regularisation.

The relative advantage $A_{\text{rel}}(y|x)$ is computed as:

$$A_{\text{rel}}(y|x) = \frac{r(y|x) - \mu}{\sigma}, \tag{9}$$

where $r(y|x)$ denotes the reward assigned to the response y, and μ and σ are the mean and standard deviation of the reward distribution within the group.

GRPO has demonstrated particular efficacy in multitask and multilingual reasoning contexts. By comparing responses within structurally related groups, it allows for more adaptive and robust policy updates, supporting better generalisation and stability across tasks. Empirical findings confirm that GRPO improves consistency, robustness, and data efficiency when compared to traditional PPO-based methods.

5.2. Multilingual Reasoning

Recent efforts to assess the capabilities of LLMs have focused on their performance in complex reasoning tasks, particularly in mathematical problem-solving. Benchmark datasets such as GSM8K and SVAMP have been widely adopted for this purpose. To extend such evaluation to multilingual contexts, Shi et al. [22] introduced MGSM, a multilingual variant of GSM8K, created by manually translating 250 test samples into various languages. Chen et al. [23] proposed MSVAMP, a multilingual extension of SVAMP following the same approach. Multiple strategies have been proposed to enhance multilingual reasoning in LLMs. These include translation-based approaches [24], SFT [25], and preference-based alignment methods [7], each of which demonstrates gains in multilingual performance. Nonetheless, these methods rely heavily on high-quality annotated data. SFT suffers from forgetting and poor generalisation, while preference-based alignment adds computational overhead through critic-based systems. Another line of research has explored the use of in-context prompting, whereby LLMs are instructed to reason step by step through carefully designed prompts. Although this strategy has proven useful in certain tasks [2], its reliance on English, combined with its inefficacy for smaller models [1], limits its applicability. Moreover, reasoning under this framework is typically induced by the prompt's structure, making it difficult to generalise across languages or domains.

While reasoning is inherently independent of language, the extent to which LLMs demonstrate consistent reasoning across linguistic boundaries remains limited. We aim to disentangle logical reasoning from linguistic surface forms by adopting a language-agnostic formalism. We propose converting problems expressed in any language into a shared formal representation that is abstract, manipulable, and semantically grounded. Reasoning operates over this intermediate form, with the final answer rendered in the target language. To support this, we instruct LLMs to abstract and solve problems via self-training, enabling scalable multilingual reasoning without the need for prompt engineering.

6. Conclusion & Future Works

Although reasoning is inherently language-agnostic, LLMs' outputs often reflect biases towards dominant pre-training languages, particularly English. While models show strong multilingual capabilities, their step-wise reasoning remains inconsistent across languages. Focusing on English and Italian, we propose a modular approach that abstracts the problem into a language-agnostic formalism, followed by structured reasoning. Using self-training, we align reasoning performances, achieving gains in both accuracy and consistency.

This work contributes to a series of studies aimed at expanding the proficiency of LLMs beyond English. In our Research, we have explored interventions at every stage—from pre-training [26, 27] and post-training [4, 11] to inference methods [1, 2, 3], and recently on multimodal reasoning [28]. In parallel, the aim is to propose methodologies based on human-inspired principles [29, 30, 31, 32] that aim to steer models away from heuristics that lead to verbatim-based [33] or symbolic-

semantic memorisation [34]. Our overarching goal is to ensure that Italian is not left behind, applying state-of-the-art approaches to enhance generative capabilities, linguistic proficiency, and other emerging competencies of contemporary LLMs in Italian.

References

[1] L. Ranaldi, G. Pucci, F. Ranaldi, E. S. Ruzzetti, F. M. Zanzotto, A tree-of-thoughts to broaden multi-step reasoning across languages, in: K. Duh, H. Gomez, S. Bethard (Eds.), Findings of the Association for Computational Linguistics: NAACL 2024, Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 1229–1241. URL: https://aclanthology.org/2024.findings-naacl.78. doi:10.18653/v1/2024.findings-naacl.78.

[2] L. Ranaldi, G. Pucci, B. Haddow, A. Birch, Empowering multi-step reasoning across languages via program-aided language models, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 12171–12187. URL: https://aclanthology.org/2024.emnlp-main.678. doi:10.18653/v1/2024.emnlp-main.678.

[3] L. Ranaldi, B. Haddow, A. Birch, When natural language is not enough: The limits of in-context learning demonstrations in multilingual reasoning, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Findings of the Association for Computational Linguistics: NAACL 2025, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 7369–7396. URL: https://aclanthology.org/2025.findings-naacl.412/. doi:10.18653/v1/2025.findings-naacl.412.

[4] L. Ranaldi, G. Pucci, Does the English matter? elicit cross-lingual abilities of large language models, in: D. Ataman (Ed.), Proceedings of the 3rd Workshop on Multi-lingual Representation Learning (MRL), Association for Computational Linguistics, Singapore, 2023, pp. 173–183. URL: https://aclanthology.org/2023.mrl-1.14. doi:10.18653/v1/2023.mrl-1.14.

[5] L. Ranaldi, G. Pucci, A. Freitas, Does the *Order* matter? Curriculum learning over languages, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 5212–5220. URL: https://aclanthology.org/2024.lrec-main.464/.

[6] L. Ranaldi, A. Freitas, Aligning large and small language models via chain-of-thought reasoning, in: Y. Graham, M. Purver (Eds.), Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, St. Julian's, Malta, 2024, pp. 1812–1827. URL: https://aclanthology.org/2024.eacl-long.109/.

[7] J. Dang, A. Ahmadian, K. Marchisio, J. Kreutzer, A. Üstün, S. Hooker, RLHF can speak many languages: Unlocking multilingual preference optimization for LLMs, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 13134–13156. URL: https://aclanthology.org/2024.emnlp-main.729/. doi:10.18653/v1/2024.emnlp-main.729.

[8] L. Ranaldi, A. Freitas, Self-refine instruction-tuning for aligning reasoning in language models, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 2325–2347. URL: https://aclanthology.org/2024.emnlp-main.139/. doi:10.18653/v1/2024.emnlp-main.139.

[9] V. Gaur, N. Saunshi, Reasoning in large language models through symbolic math word problems, in: Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 5889–5903. URL: https://aclanthology.org/2023.findings-acl.364. doi:10.18653/v1/2023.findings-acl.364.

[10] L. Pan, A. Albalak, X. Wang, W. Wang, Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 3806–3824. URL: https://aclanthology.org/2023.findings-emnlp.248/. doi:10.18653/v1/2023.findings-emnlp.248.

[11] L. Ranaldi, G. Pucci, Multilingual reasoning via self-training, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 11566–11582. URL: https://aclanthology.org/2025.naacl-long.577/. doi:10.18653/v1/2025.naacl-long.577.

[12] T. Wang, S. Li, W. Lu, Self-training with direct preference optimization improves chain-of-thought reasoning, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 11917–11928. URL: https://aclanthology.org/2024.acl-long.643/. doi:10.18653/v1/2024.acl-long.643.

[13] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, inter alia, The llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.

[14] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, D. Guo, Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL: https://arxiv.org/abs/2402.03300. arXiv:2402.03300.

[15] I. Mirzadeh, K. Alizadeh, H. Shahrokhi, O. Tuzel, S. Bengio, M. Farajtabar, Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models, 2024. URL: https://arxiv.org/abs/2410.05229. arXiv:2410.05229.

[16] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, S. Ye, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Zhao, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Xu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, Z. Zhang, Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL: https://arxiv.org/abs/2501.12948. arXiv:2501.12948.

[17] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, 2022. URL: https://arxiv.org/abs/2203.02155. arXiv:2203.02155.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, 2017. URL: https://arxiv.org/abs/1707.06347. arXiv:1707.06347.

[19] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, C. Finn, Direct preference optimization: Your language model is secretly a reward model, 2024. URL: https://arxiv.org/abs/2305.18290. arXiv:2305.18290.

[20] Y. Lin, S. Seto, M. Ter Hoeve, K. Metcalf, B.-J. Theobald, X. Wang, Y. Zhang, C. Huang, T. Zhang, On the limited generalization capability of the implicit reward model induced by direct preference optimization, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 16015–16026. URL: https://aclanthology.org/2024.findings-emnlp.940/. doi:10.18653/v1/2024.findings-emnlp.940.

[21] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, D. Guo, Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL: https://arxiv.org/abs/2402.03300. arXiv:2402.03300.

[22] F. Shi, M. Suzgun, M. Freitag, X. Wang, S. Srivats, S. Vosoughi, H. W. Chung, Y. Tay, S. Ruder, D. Zhou, D. Das, J. Wei, Language models are multilingual chain-of-thought reasoners, 2022. arXiv:2210.03057.

[23] N. Chen, Z. Zheng, N. Wu, M. Gong, Y. Song, D. Zhang, J. Li, Breaking language barriers in multilingual mathematical reasoning: Insights and observations, 2023. arXiv:2310.20246.

[24] L. Ranaldi, G. Pucci, A. Freitas, Empowering cross-lingual abilities of instruction-tuned large language models by translation-following demonstrations, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of

the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 7961–7973. URL: https://aclanthology.org/2024.findings-acl.473/. doi:10.18653/v1/2024.findings-acl.473.

[25] A. Üstün, V. Aryabumi, Z. Yong, W.-Y. Ko, D. D'souza, G. Onilude, N. Bhandari, S. Singh, H.-L. Ooi, A. Kayid, F. Vargus, P. Blunsom, S. Longpre, N. Muennighoff, M. Fadaee, J. Kreutzer, S. Hooker, Aya model: An instruction finetuned open-access multilingual language model, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 15894–15939. URL: https://aclanthology.org/2024.acl-long.845/. doi:10.18653/v1/2024.acl-long.845.

[26] L. Ranaldi, G. Pucci, F. M. Zanzotto, Modeling easiness for training transformers with curriculum learning, in: R. Mitkov, G. Angelova (Eds.), Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing, INCOMA Ltd., Shoumen, Bulgaria, Varna, Bulgaria, 2023, pp. 937–948. URL: https://aclanthology.org/2023.ranlp-1.101/.

[27] L. Ranaldi, G. Pucci, F. M. Zanzotto, How far does the sequence of compositions impact multilingual pre-training?, in: F. Dell'Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 796–804. URL: https://aclanthology.org/2024.clicit-1.86/.

[28] L. Ranaldi, F. Ranaldi, G. Pucci, R2-MultiOmnia: Leading multilingual multimodal reasoning via self-training, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 8220–8234. URL: https://aclanthology.org/2025.acl-long.402/. doi:10.18653/v1/2025.acl-long.402.

[29] L. Ranaldi, G. Pucci, Knowing knowledge: Epistemological study of knowledge in transformers, Applied Sciences 13 (2023). URL: https://www.mdpi.com/2076-3417/13/2/677. doi:10.3390/app13020677.

[30] G. Pucci, F. M. Zanzotto, L. Ranaldi, Animate, or inanimate, that is the question for large language models, Information 16 (2025). URL: https://www.mdpi.com/2078-2489/16/6/493. doi:10.3390/info16060493.

[31] M. Mastromattei, L. Ranaldi, F. Fallucchi, F. M. Zanzotto, Syntax and prejudice: ethically-charged biases of a syntax-based hate speech recognizer unveiled, PeerJ Computer Science 8 (2022) e859. URL: http://dx.doi.org/10.7717/peerj-cs.859. doi:10.7717/peerj-cs.859.

[32] L. Ranaldi, Survey on the role of mechanistic interpretability in generative ai, Big Data and Cognitive Computing 9 (2025). URL: https://www.mdpi.com/2504-2289/9/8/193. doi:10.3390/bdcc9080193.

[33] F. Ranaldi, E. S. Ruzzetti, D. Onorati, L. Ranaldi, C. Giannone, A. Favalli, R. Romagnoli, F. M. Zanzotto, Investigating the impact of data contamination of large language models in text-to-SQL translation, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 13909–13920. URL: https://aclanthology.org/2024.findings-acl.827/. doi:10.18653/v1/2024.findings-acl.827.

[34] F. Ranaldi, A. Zugarini, L. Ranaldi, F. M. Zanzotto, Protoknowledge shapes behaviour of llms in downstream tasks: Memorization and generalization with knowledge graphs, 2025. URL: https://arxiv.org/abs/2505.15501. arXiv:2505.15501.

A. SAGE Instruction Template

#Role
You are an experienced expert skilled in multilingual mathematical reasoning problems.

#Task
You are presented with a mathematical reasoning problem in a given language. Follow the steps below rigorously to formalise and solve it.

#Instructions

1) Formalisation (Language-Agnostic): Identify and define the key mathematical components of the problem, such as variables, functions, operations, and constraints. Structure these components in an abstract manner to ensure a clear and precise formulation. *Label this step as* *<formalisation>....</formalisation>*

2) Reasoning Execution: Solve the problem systematically by breaking it into logical steps. Clearly justify each step using natural language explanations while maintaining logical rigor. Express the final answer in the same language as the input query. *Label this step as* *<reasoning>....</reasoning>*

Final Answer: Present the extracted answer in a concise format, marked as "**The answer is: [num]**" in the same language as the query. *Label this step as* *<answer>....</answer>*

#Question
{**question**}

Table 4
The SAGE instructs the model to abstract problem components and deliver step-wise reasoning paths that lead the model to solve multilingual tasks. Following [11] we propose principled reasoning framework based on structured step-wise passages to reach the final solution.

B. Annotations Pipeline

We use SAGE to generate synthetic demonstrations for training smaller LLMs. We use GPT-4o as an annotator and use the annotations to warm-up the models with the proposed methodologies. We then conduct a complete Self-training phase. Moreover, we conduct the Self-training by using self-generated data (generated by the trained models themselves). We define these configurations 'Full'-Self-training. In both cases, the demonstrations are generated by prompting the models using instructions detailed in Appendix A. However, while GPT-4o follows the instructions well (in fact, we did not find any significant issues), the other models generate outcomes that include errors. To handle this, we evaluated the quality of the generated demonstrations by filtering out inaccurate examples to get a gold instruction set. In particular, we removed all inaccurate answers (outputs that do not match the exact target string metric). Then, we control if the demonstrations follow correctly the steps indicated in our prompt (see Table 4) using GPT-4o-mini and the prompt in Appendix **??**.

C. Evaluation Metrics

We used a double-check to assess the accuracy of the responses delivered in the different experiments. In the first step, we used an exact-match heuristic. However, since some experiments required a more accurate response check, we used GPT-4o-mini as a judge.

D. Models and Hyperparameters

Hyperparameters In §3.2, we described the standard Self-training setting. However, we have proposed different experimental settings. In the Self-training experimental setting, we conducted three iterations as proposed in [12, 14]. In the SFT-only and RL-only settings, we used warm-up and four epochs and 8000 steps, respectively. We conducted this study after the pilot experiments shown in the previous sections.

E. Models Vesions

Model	Version
Llama3-8(-instruct)	meta-llama/Meta-Llama-3-8B-Instruct
Phi-3(-mini-instruct)	microsoft/Phi-3-mini-4k-instruct
DeepSeekMath-7B	deepseek-ai/deepseek-math-7b-instruct
GPT-4o	gpt-4o-2024-08-06
GPT-4o-mini	gpt-4o-mini-2024-07-18

Table 5
List the versions of the models proposed in this work, which can be found on huggingface.co. We used all the default configurations proposed in the repositories for each model.

F. Data Composition

As evaluation sets, we use the tasks introduced in §3.3. These tasks are used to assess the performance of LLMs, but they do not have reserved sets for evaluation and training. Therefore, to produce a training set, we split mSVAMP into training and testing. Table 6 shows the instances of each dataset in training and testing. To ensure the languages are perfectly balanced, we translated 350 samples from English to Telugu (language non-present in mSVAMP). This subset was used for training purposes only.

Task	Total	Test	Train. Set	# dim
mGSM	$0.5k$	$0.5k$	No	No
mGSM-Symbolic	$0.5k$	$0.5k$	No	No
mSVAMP	$2k$	$0.5k$	Yes	$1k$

Table 6
Training and evaluation data. *($1k$ is equal to 1000).

The data are perfectly balanced between the languages in the proposed tasks. However, as described in Appendix B, the qualities of the annotations are not perfect. Behind filtering the annotations, we obtained a reduced dataset. To have fair, balanced subsets, we use 1k samples in total. We use 1k samples when instructing the models for DPO and SFT. For the Self-training, we used as the initial subset (§2.2) 60% of the filtered samples balanced between all languages.

G. Number of Iterations

Following pilot experiments, we set the number of iterations of self-tuning at three. Figure 7 shows the performance trend by increasing the number of iterations, epochs and steps after warm-up (wup).

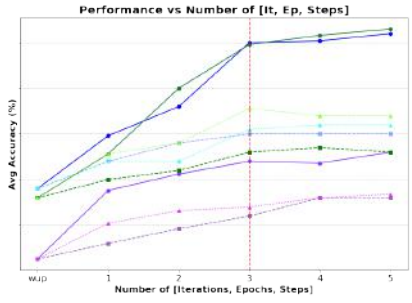

Table 7
Average accuracies on mGSM.