




# MetaScale: Test-Time Scaling with Evolving Meta-Thoughts

Qin Liu  Wenxuan Zhou  Nan Xu  James Y. Huang  Fei Wang 

Sheng Zhang  Hoifung Poon  Muhao Chen 

 University of California, Davis  University of Southern California  Microsoft Research  
{qinli, muhchen}@ucdavis.edu {zhouwenx, nanx, huangjam, fwang598}@usc.edu  
{shezhan, hoifung}@microsoft.com

## Abstract

One critical challenge for large language models (LLMs) in complex reasoning is their reliance on matching reasoning patterns from training data, instead of proactively selecting the most appropriate cognitive strategy to solve a given task. Existing approaches impose fixed cognitive structures that enhance performance in specific tasks but lack adaptability across diverse scenarios. To address this limitation, we introduce METASCALE, a test-time scaling framework based on *meta-thoughts*, i.e., adaptive thinking strategies tailored to each task. METASCALE initializes a pool of candidate meta-thoughts, then iteratively selects and evaluates them using a multi-armed bandit algorithm with upper confidence bound selection, guided by a reward model. To further enhance adaptability, a genetic algorithm evolves high-reward meta-thoughts, refining and extending the strategy pool over time. By dynamically proposing and optimizing meta-thoughts at inference time, METASCALE improves both accuracy and generalization across a wide range of tasks. Experimental results demonstrate that METASCALE consistently outperforms standard inference approaches, achieving an 11% performance gain in win rate on Arena-Hard for GPT-4o, surpassing o1-mini by 0.9% under style control. Notably, METASCALE scales more effectively with increasing sampling budgets and produces more structured, expert-level responses.<sup>1</sup>

## 1 Introduction

Large language models (LLMs) (Achiam et al., 2023; Dubey et al., 2024; Guo et al., 2025) have demonstrated remarkable capabilities across a wide range of cognitive tasks. With minimal or no guidance, they can perform human-like multi-step thinking processes (Kojima et al., 2022; Wang and Zhou,

2024). However, one critical challenge is that they may not proactively determine when or how to apply different thinking processes. Instead, their thinking process is often shaped by pattern matching from training data and often diverges from human reasoning patterns (Dziri et al., 2024; Wang et al., 2025b; Bao et al., 2025; Wang et al., 2025a), rather than selecting the most appropriate way to think or respond.

To address these challenges, researchers have sought to explicitly define the thinking processes of LLMs by mimicking aspects of human cognition and introducing reasoning strategies such as self-verification (Weng et al., 2022; Madaan et al., 2024), chain-of-thought prompting (Wei et al., 2022), and reverse thinking (Chen et al., 2024), etc. By constraining LLMs to follow these structured thinking processes, models have shown improved performance on specific tasks. However, these methods are often designed for particular types of tasks, limiting their adaptability and effectiveness across diverse tasks (Sprague et al., 2024; Xu et al., 2024; Xu and Ma, 2025). More critically, these approaches impose fixed thinking structures rather than enabling LLMs to adaptively determine the most effective task-specific strategy, potentially limiting their performance.

To address these limitations, we propose the idea of *meta-thinking*, a process where LLMs first reflect on their approach before generating a response. Rather than immediately solving a problem, the model initiates the process by determining how to think and select the most suitable cognitive strategy from a range of available options. For example, when tackling a complex logical puzzle, an LLM might choose to work forward from given conditions, reason backward from the goal, or verify its answer through self-reflection. By incorporating this meta-thinking step, LLMs can dynamically adapt their reasoning process to different tasks, rather than relying on rigid, predefined heuristics.

<sup>1</sup>Our code is available at <https://github.com/luka-group/MetaScale>.

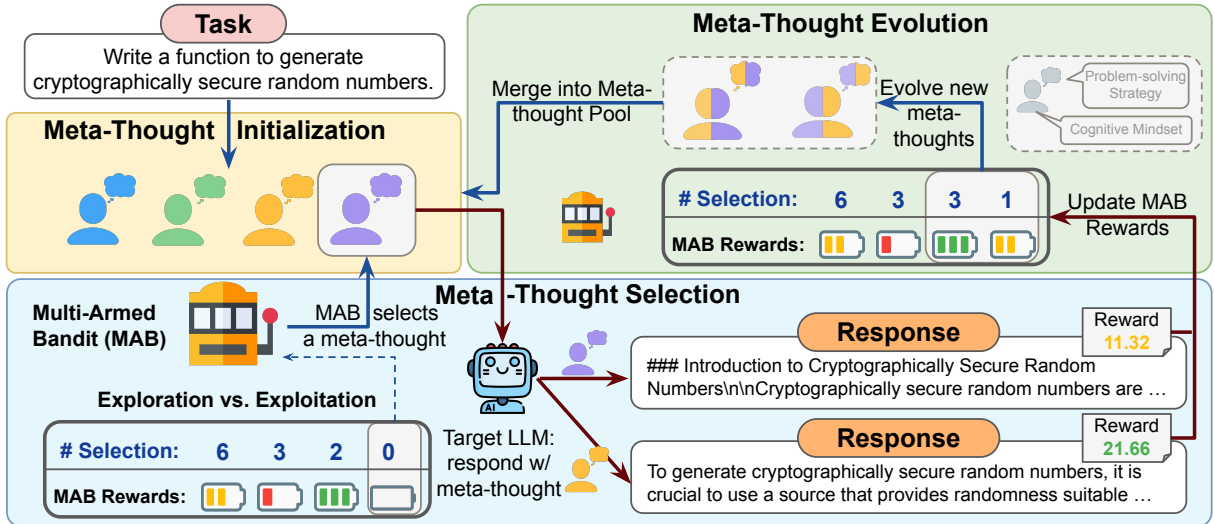


Figure 1: Overview of METASCALE. METASCALE initializes a pool of meta-thoughts, selects and evaluates them using the Upper Confidence Bound (UCB) algorithm, and periodically evolves high-performing meta-thoughts. The best response is returned after exhausting the sampling budgets.

Building on this idea, we introduce METASCALE, a test-time scaling framework that incorporates meta-thinking. The goal is to enable LLMs to explore different thinking strategies and generate the most effective response for a given input. The process begins with the generation of an initial pool of meta-thoughts, where the LLM autonomously formulates reasoning strategies tailored to the given problem. To enhance diversity, we retrieve relevant input-output pairs from instruction-tuning datasets (Zhao et al., 2024) and prompt the LLM to abstract thinking patterns specific to the current task. Next, we iteratively prompt the LLM based on the selected meta-thoughts. The selection process is guided by an Upper Confidence Bound (UCB) (Auer, 2002) algorithm, which balances exploration and exploitation. A reward model (Liu et al., 2024) then evaluates response quality. To further promote diversity, we periodically apply a *genetic algorithm*, generating new meta-thoughts by evolving high-reward ones. Finally, the best response found through this process is returned after reaching the sampling budget.

Experiments show that METASCALE significantly enhances the problem-solving capabilities of LLMs across diverse tasks, consistently outperforming baseline methods on various real-world reasoning challenges. More importantly, METASCALE scales effectively with increased sampling budgets, further improving performance by evolving meta-thoughts that lead to task solu-

tions of higher quality.

Our contributions are threefold. First, we introduce METASCALE, a novel test-time scaling strategy that enables LLMs to proactively select and refine cognitive strategies at test time, moving beyond static, predefined heuristics. Second, we adapt the genetic algorithm to evolve high-reward meta-thoughts over multiple iterations, further enhancing adaptability and reasoning efficiency. Third, experiments demonstrate that METASCALE consistently scales with increased sampling budgets, effectively leveraging additional computational resources to refine responses.

## 2 Method

In this section, we introduce METASCALE, a test-time scaling framework that explores diverse cognitive strategies to optimize the reasoning process during inference. We first provide a brief overview and relevant preliminaries in §2.1, followed by a detailed description of meta-thought initialization (§2.2), selection (§2.3), and evolution (§2.4).

### 2.1 METASCALE Overview

METASCALE is a test-time scaling strategy designed to enhance the problem-solving capabilities of LLMs by incorporating *meta-thoughts* that guide the reasoning process before generating a response. Inspired by human cognitive processes, a *meta-thought* consists of two key components:

1. **Cognitive Mindset**, which is an appropriate

perspective, expertise, or role that the model adopts to approach the task.

2. **Problem-Solving Strategy**, which is a structured pattern used to formulate a sound solution based on the given mindset.

For instance, when asked to “*Use ABC notation to write a melody in the style of a folk tune*”, a well-formed meta-thought could include a cognitive mindset of “*a musician or music educator with expertise in music theory and composition*”, and a problem-solving strategy indicating “*First present a melody that captures the characteristics of a traditional folk tune, then . . .*” The model then utilizes these strategies to guide its reasoning process and generate a response.

METASCALE develops effective meta-thoughts for an input task in three phases: initialization, selection, and evolution. During initialization, METASCALE generates a diverse pool of reasoning strategies tailored to the input task by leveraging its own prior knowledge and instruction-tuning datasets. Next, a Multi-Armed Bandit (MAB) algorithm is used to explore and select promising meta-thoughts. Finally, METASCALE employs a genetic algorithm to iteratively refine and expand the meta-thought pool. To maintain computational feasibility, METASCALE operates within a fixed sampling budget and returns the response with the highest score given by a reward model once the budget is exhausted. We illustrate the overall design of METASCALE in Fig. 1.

Before delving into the details of METASCALE, we provide the background knowledge on genetic algorithms and multi-armed bandit (MAB) algorithms, two key components of our framework. Genetic algorithms enable iterative refinement of meta-thoughts, while MAB facilitates selection of the most effective meta-thought during inference.

**Genetic Algorithms.** Genetic algorithms (Golberg, 1989; Mitchell, 1998) are optimization techniques inspired by natural selection. They consist of three key components: *initialization*, which generates an initial population of candidate solutions; *fitness evaluation*, which assesses solution quality based on a predefined objective; and *evolution*, where high-fitness candidates are selected (*selection*), combined (*crossover*), or modified (*mutation*) to create new solutions. Through iterative refinement, the population converges toward solutions with better fitness. The process terminates once

a predefined fitness threshold is reached or a set number of generations is completed.

**Multi-Armed Bandit.** The MAB problem addresses the trade-off between exploration (testing new options) and exploitation (choosing known high-reward options) in sequential decision-making (Vermorel and Mohri, 2005; Audibert et al., 2009). The goal is to maximize cumulative rewards by selecting from multiple options (arms). A key approach, the Upper Confidence Bound (UCB) algorithm (Auer et al., 2002; Auer, 2002), balances this trade-off by combining an option’s empirical mean reward with an uncertainty bonus that decreases with more sampling. This ensures efficient exploration while favoring consistently high-performing choices.

## 2.2 Meta-Thought Initialization

The initialization of meta-thoughts in METASCALE aims to generate a diverse set of reasoning strategies tailored to the given problem. This process follows two complementary strategies. First, the target LLM is prompted to self-compose a set of reasoning strategies for the given problem, reflecting different problem-solving heuristics and cognitive patterns that can enhance inference performance. Second, to further enrich diversity, METASCALE leverages instruction-tuning datasets such as WildChat (Zhao et al., 2024), which contain extensive task-solution pairs generated by advanced models. We retrieve the top 8 similar tasks (measured by the similarity between the embeddings of task descriptions) from these datasets and prompt the LLM to extract and adapt the underlying thinking patterns to the current problem.<sup>2</sup> The assumption is that reasoning strategies used for similar questions can offer valuable perspectives for solving new ones. Together, these self-composed and dataset-derived strategies form the initial pool of meta-thoughts used to guide problem-solving.

## 2.3 Meta-Thought Selection as MAB

Once the meta-thought pool is initialized, METASCALE selects the most promising meta-thought for generating a response at each iteration step. We here formalize the selection of meta-thoughts at inference time as a multi-armed bandit (MAB) problem, where each meta-thought option

<sup>2</sup>The prompts used for this process are illustrated in Appx. §B.

serves as an ‘‘arm’’ and the goal is to maximize the reward of the corresponding model response based on the selected meta-thought.

Specifically, we apply the Upper Confidence Bound (UCB) algorithm to guide the selection process by balancing exploration and exploitation of the available meta-thoughts, which dynamically selects a candidate based on both its past performance and the uncertainty about its future performance. Specifically, given a candidate set of meta-thoughts  $\mathcal{M}_q$  initialized (at inference time) for an input query  $q$ , the selected meta-thought  $M^*$  is determined by

$$M^* = \arg \max_{M \in \mathcal{M}_q} \left( \mu_M + \beta \sqrt{\frac{\log t}{N_M}} \right),$$

where  $\mu_M$  represents the empirical mean reward of meta-thought  $M$ , computed based on past attempts, while  $N_M$  denotes the number of times  $M$  has been selected.  $\beta \sqrt{\frac{\log t}{N_M}}$  introduces an exploration term that prioritizes meta-thoughts that have been less frequently tested, where  $t$  is the total number of attempts processed. The parameter  $\beta$  controls the trade-off between exploiting well-performing options and exploring new ones.

To further improve sampling efficiency, we augment the UCB strategy with a mini-batch sampling scheme. After computing the empirical mean rewards  $\{\mu_M\}_{M \in \mathcal{M}_q}$  for all meta-thoughts, we convert them into a probability distribution via soft-max function  $p_M = \frac{\exp(\mu_M)}{\sum_{M' \in \mathcal{M}_q} \exp(\mu_{M'})}$ . Given a fixed batch size  $B$ , each meta-thought  $M$  is allocated  $n_M = \lfloor B p_M \rfloor$  sampling budget, which is subject to  $\sum_M n_M = B$ . Any remaining slots caused by rounding are assigned to the meta-thoughts with the largest fractional parts of  $B p_M$ . The resulting batch of  $B$  responses is evaluated in parallel by the reward model, and the obtained scores update  $\mu_M$  and  $N_M$ , thereby refining the UCB statistics for the next iteration.

To assess response quality during fitness evaluation, we use an outcome-based general-purpose reward model (Liu et al., 2024). Given input and response, the reward model evaluates response quality and assigns a reward score. This reward score, in turn, updates the UCB of the meta-thought, which influences subsequent selection decisions by refining the exploration-exploitation balance according to the accumulated performance history.

## 2.4 Genetic Meta-Thought Evolution

To further enhance diversity and continuously expand the pool of cognitive strategies, METASCALE periodically applies a genetic algorithm to evolve the meta-thoughts. High-performing meta-thoughts, identified by their accumulated UCB as described in §2.3, are selected as parents for producing the next generation of new child meta-thoughts. Specifically, given the high-UCB meta-thoughts, the target LLM is prompted to develop a set of refined meta-thoughts that integrates and improves upon the selected parents.

Rather than directly applying explicit crossover or mutation policies as usually applied in a typical genetic algorithm, we prompt the target LLM itself to determine how to combine the reasoning heuristics of the parent meta-thoughts and evolve them into improved counterparts. We find that directly performing crossover or mutation at the text level often produces strategies that make no sense. Instead, prompting LLMs allows for a more flexible and context-sensitive evolution of meta-thoughts, ensuring that the evolved strategies continue to reflect the problem-solving patterns most effective for the current task over successive iterations.

Based on the test-time scaling nature of METASCALE, we set the termination criterion according to a predetermined budget for the number of attempts. Once the budget is exhausted, METASCALE terminates and returns the model response (which is prompted with meta-thought) with the highest reward score among all the attempts as evaluated by the reward model. The overall algorithm of METASCALE is outlined in Alg. 1.

## 3 Experiments

In this section, we evaluate METASCALE on three different problem-solving tasks with two models of varying scales. We provide an overview of our experimental settings (§3.1) and present a comparison of empirical results (§3.2) followed by further analysis (§3.3).

### 3.1 Experimental Setup

**Evaluation Dataset** To evaluate the effectiveness of our METASCALE for problem-solving, we consider the following mathematical reasoning benchmark and general-purpose tasks.

**GSM8K** (Cobbe et al., 2021) consists of high-quality, linguistically diverse grade school math word problems. We use accuracy to measure mod-

Method	Arena-hard				MMLU-Pro	GSM8K
	Avg. Win Rate	95% CI Lower	95% CI Upper	Number of Tokens	Acc	Acc
GPT-4o-0806						
Vanilla Model 1-Pass	82.14	77.68	88.10	573	68.75	92.19
Chain-of-Thought 1-Pass	84.72	79.30	88.91	663	71.88	93.75
Best-of-N	84.17	78.73	89.31	594	71.88	92.19
Best-of-N w/ CoT	87.42	81.66	91.37	681	73.44	<b>95.31</b>
Best-of-N w/ Self-Discover	89.32	83.41	92.16	726	76.56	93.75
METASCALE w/o Evolution	89.63	84.57	93.62	689	76.56	93.75
METASCALE	<b>93.14</b>	<b>90.44</b>	<b>96.34</b>	699	<b>78.12</b>	<b>95.31</b>
Llama-3.1-8B-Instruct						
Vanilla Model 1-Pass	18.94	13.50	23.50	481	31.25	73.44
Chain-of-Thought 1-Pass	12.50	8.74	17.50	625	35.94	78.13
Best-of-N	25.88	16.76	32.37	493	39.06	79.69
Best-of-N w/ CoT	28.33	21.63	34.74	571	45.31	82.81
Best-of-N w/ Self-Discover	28.62	22.87	33.49	581	46.88	79.69
METASCALE w/o Evolution	28.93	23.02	35.26	575	46.88	<b>84.38</b>
METASCALE	<b>30.86</b>	<b>23.97</b>	<b>35.29</b>	630	<b>50.00</b>	82.81

Table 1: Main results (%) on Arena-Hard, MMLU-Pro, and GSM8K, evaluated using GPT-4o and Llama-3.1-8B-Instruct as base models. The best scores for each base model are highlighted in **bold**.

els’ performance. To compute accuracy, we parse the completion to extract the numeric value and conduct an exact match with the reference answer. **MMLU-Pro** (Wang et al., 2024b) is a 10-way multiple-choice extended version of the MMLU dataset (Hendrycks et al., 2020). Spanning 14 diverse domains, we use it to evaluate the proficient-level multi-disciplinary language understanding and reasoning capabilities of LLMs. To extract the option letter from responses, we employ the suggested regular expression matching first, followed by the secondary regex matching. The correctness averaged over all testing cases is used as the final accuracy.

**Arena-Hard** (Li et al., 2024) is composed of challenging benchmark prompts sourced from Chatbot Arena (Chiang et al., 2024) and adopts the LLM-as-a-Judge for automatic model evaluation. We follow their default evaluation setting by using GPT-4-Turbo as the judge to estimate human preferences of evaluated models against the baseline model GPT-4-0314, and report the average win rate. Additionally, following the official setting, we report the 95% confidence interval of the win rate, providing both the lower and upper bounds.

**Baseline Methods** For each task, we compare METASCALE to four baseline inference methods with the same reward model and scaling budget if applicable. **1-Pass**, where the target LLM performs a single forward pass with greedy decoding with the temperature fixed to 0. This represents the most computationally efficient strategy and serves as a lower bound on performance since it does not explore alternate outputs. **Chain-of-Thought** (CoT, Wei et al., 2022), where the target LLM is prompted to generate a reasoning process towards the final solution for the given task. This is also performed in a greedy decoding manner with the temperature set to 0. **Best-of-N** (Brown et al., 2024), where up to 128 candidate solutions are sampled with temperature of 0.6 that allows diversity. The solution with the highest score given by the reward model is returned as the final response. **CoT with Best-of-N**, where the sampling budget is also 128, and the LLM is prompted to perform CoT for each sampling. A temperature of 0.6 is applied. **Self-Discover with Best-of-N** (Zhou et al., 2024), where the solution of each task instance is sampled 128 times based on the reasoning structure tailored for the task via meta-reasoning. Besides, **METASCALE w/o Evolution** also serves as a strong baseline that is com-

Model	Avg. Win Rate	95% CI Lower	95% CI Upper	Number of Tokens
claude-3-5-sonnet	82.0	80.4	84.2	567
o1-preview-2024-09-12	81.6	79.2	83.8	1193
METASCALE <sub>128</sub>	80.1	78.2	82.2	744
o1-mini-2024-09-12	79.2	76.6	81.6	1399
METASCALE <sub>64</sub>	78.8	76.1	81.5	737
METASCALE <sub>32</sub>	76.7	74.4	79.9	739
METASCALE <sub>16</sub>	74.7	71.8	77.4	735
gpt-4-turbo-2024-04-09	74.4	71.9	76.5	662
gpt-4-0125-preview	73.5	71.1	75.3	619
METASCALE <sub>8</sub>	71.4	68.7	74.2	734
gpt-4o-2024-08-06	71.0	68.5	73.8	594

Table 2: Results on full Arena-Hard dataset under style control (Chiang et al., 2024) for a fair comparison that eliminates the effects of style (e.g. answer token length and number of markdown headers) on response judgment. METASCALE<sub>k</sub> represents METASCALE based on gpt-4o-2024-08-06 with sampling budget of  $k$ . The rest of the models perform a single forward pass. Evaluation results without style control are shown in Tab. 3.

parable to in-context learning with retrieval augmentation (ICL with RAG), where retrieved candidates are used directly without any evolutionary update, thus offering a static yet informative context for generation. A detailed comparison against retrieval-augmented few-shot variants is also provided in Appx. §E.

**Models** We use one state-of-the-art LLM, GPT-4o (gpt-4o-0806) (Achiam et al., 2023), and one open-source LLM Llama-3.1-8B-Instruct (Dubey et al., 2024) for evaluation. Skywork-Reward-Llama-3.1-8B-v0.2 (Liu et al., 2024) is directly adopted as the reward model for assigning reward scores to candidate responses, based on which the optimal response is chosen as the final answer. To ensure fair comparisons, we use consistent inference settings across models and maintain the same sampling budgets.

**Implementation Details** Due to limited computational resources and high API costs, we randomly sample 64 testing instances from each benchmark to create new subsampled evaluation sets, which are used consistently across all methods. For a comprehensive comparison, we also provide results on the full Arena-Hard dataset (500 instances). To precisely evaluate the effectiveness of different test-time compute scaling strategies without interference from demonstrations, we prompt LLMs in the zero-shot setting on all studied tasks. Wild-Chat (Zhao et al., 2024) is used for meta-thought initialization (§2.2), with an initialization pool of size 8 across all benchmarks.

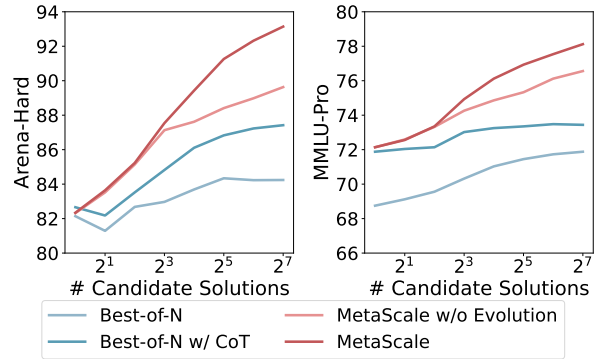


Figure 2: Results (%) on Arena-Hard and MMLU-Pro with varying sampling budgets (1 to 128) using GPT-4o as the base model. METASCALE improves more rapidly as the sampling budget increases, whereas best-of-N shows slower gains or even stops to increase at the maximum budget.

### 3.2 Main Results

**METASCALE Consistently Outperforms Baselines.** We present the main results in Tab. 1 and find that METASCALE consistently achieves equal or superior performance compared to both one-pass methods and Best-of-N methods, regardless of whether they are prompted with or without CoT. Notably, METASCALE based on GPT-4o outperforms its one-pass counterpart by 11.00%, 9.37%, and 3.12% on Arena-Hard, MMLU-Pro, and GSM8K, respectively. It also surpasses the standard Best-of-N method by 8.97%, 6.24%, and 3.12%, respectively. Further, GPT-4o outperforms o1-mini (Jaech et al., 2024) with METASCALE under style control (Tab. 2). Similar improvements are observed when using Llama-3.1-8B-Instruct as the base model. These results demonstrate that integrating meta-thoughts enables LLMs to scale more effectively during test time as the number of samples increases.

**Evolving Meta-Thoughts Improves Performance.** We compare the performance of METASCALE with and without the evolution of meta-thoughts during testing. Our results show that METASCALE achieves better performance across most evaluation datasets and models, with only a slight underperformance on GSM8K when using Llama-3.1-8B-Instruct. This demonstrates that evolving meta-thoughts during test time leads to more effective reasoning and decision-making.

### 3.3 Analysis

**METASCALE Benefits from Increased Sampling Budgets.** As illustrated in Fig. 2, METASCALE

Task: If you were a corporate law with 15 years of mergers and acquisitions experience, how would you pivot to launch an AI enable tech startup step by step and in detail?	
Model	Response
GPT-4o-0806 1-Pass	Pivoting from a career in corporate law, particularly with a focus on mergers and acquisitions, to launching an AI-enabled tech startup is a significant transition. However, your experience in M&A can provide valuable skills in negotiation, strategic thinking, and understanding complex business structures. Here's a step-by-step guide to help you make this transition: ### Step 1: Self-Assessment and Skill Development 1. <b>**Identify Your Interests and Strengths**</b> : Reflect on what aspects of AI and technology excite you.
MetaScale	<p><b>Meta-thought</b>: {'<i>Cognitive mindset</i>': 'A startup accelerator mentor with a background in both corporate law and technology entrepreneurship, who specializes in guiding professionals through career transitions into the tech startup world...', '<i>Problem-solving strategy</i>': 'Create a strategic plan for transitioning from a seasoned corporate law professional in mergers and acquisitions to launching an AI-enabled tech startup. Begin by identifying and capitalizing on transferable skills such as strategic analysis, negotiation, and business acumen. Acquire foundational AI knowledge through formal education or partnerships ....'}</p> <p>Transitioning from a seasoned corporate law professional in mergers and acquisitions to launching an AI-enabled tech startup is an ambitious and exciting journey. Here's a step-by-step strategic plan to guide you through this transition: ### 1. <b>**Identify and Leverage Transferable Skills**</b> - <b>**Strategic Analysis and Planning**</b>: Utilize your ability to analyze complex business situations to understand AI market dynamics and identify potential opportunities. ...</p>

Figure 3: Model responses for a task from Arena-hard benchmark. The meta-thought used for the optimal response which obtains the highest reward among all candidate solutions sampled by METASCALE is marked in blue. Based on the optimized heuristic provided by the meta-thought, METASCALE tackles the task in a more sophisticated way that precisely targets the core challenges of the problem, hitting a solution with maximal efficiency and quality. In contrast, the vanilla GPT-4o-0806 could only provide general ideas which may lack professional insights.

demonstrates significantly higher gains as the number of candidate solutions increases compared to baseline methods. On both Arena-Hard and MMLU-Pro, METASCALE consistently outperforms baseline methods, i.e., Best-of-N and Best-of-N w/ CoT, with its performance improving more steeply as the sampling budget expands. This suggests that METASCALE is a more effective scaling strategy that efficiently explores better solutions, whereas Best-of-N slows down at higher budgets. The evolutionary nature of METASCALE further amplifies this advantage, enabling the target LLM to apply refined heuristics through iterative meta-thought selection and evolution.

**Evolutionary Refinement Enhances Selection of Optimal Meta-thoughts.** Fig. 4 demonstrates how the distribution of selected meta-thoughts shifts over multiple evolutionary iterations throughout the METASCALE process. Notably, the frequency of highly rewarded meta-thoughts increases significantly as the iteration of meta-thought evolution proceeds, particularly from the tenth iteration onward, where a substantial portion of selections of over 20% occur. This suggests that the iterative meta-thought refinement process within METASCALE enables the model to progressively converge on more optimal solutions. The lower selection counts in earlier stages indicate a wider ex-

ploration phase before METASCALE effectively approaches the best-performing meta-thoughts. Overall, the evolutionary refinement of meta-thoughts allows for better performance, indicated by higher chances of hitting the maximum MAB reward.

**METASCALE Generates Targeted and Expert-Level Solutions.** We provide a case study of model responses in Fig. 3, comparing a standard GPT-4o single-pass response with an optimized response generated by METASCALE. Given the task of launching a startup based on past experience, the single-pass response offers broad and generic guidance, failing to identify the core challenge of transitioning from corporate law to an AI startup. With a lack of specificity, the LLM provides surface-level advice without deep insights into the strategic and operational challenges involved.

In contrast, METASCALE refines its response by incorporating iteratively refined meta-thoughts, which allow for a more structured, expert-driven approach. The optimized response precisely identifies key transferable skills, highlights AI market dynamics, and presents a clear, step-by-step strategy tailored to the user's professional background. By leveraging meta-thought, the response is not only contextually relevant but also highly aligned with domain-specific expertise.

## 4 Related Work

**Test-time Compute Scaling.** As a complementary approach to prohibitively expensive train-time compute scaling (i.e., larger models pre-trained with growing amounts of tokens), test-time compute scaling has exhibited preliminary success on both proprietary (Snell et al., 2024) and open-source models (HuggingFaceH4, 2025). There are two major ways to scale test-time compute: 1) *self-refinement* where models iteratively generate and revise responses based on internal or external feedback (Xi et al., 2024; Song et al., 2024; Cheng et al., 2024), and 2) *searching against a verifier* where models sample multiple candidate answers and a verifier is used to select the best one (Huang et al., 2024; Wang et al., 2024a; Guan et al., 2025). However, self-refinement relies on built-in correction capabilities from the generative model, which is not sufficiently emphasized during model training (Ji et al., 2025). Empirical studies on tasks such as code generation (Olausson et al., 2023) and commonsense QA (Huang et al., 2023) show that self-refinement is not a guaranteed solution for performance improvement. On the other hand, searching-based strategies lead to performance gain with a flexible verifier design, ranging from hard-coded heuristics (Snell et al., 2024) to learned process- or outcome-based reward models (HuggingFaceH4, 2025). Searching approaches such as Best-of-N sampling (Cobbe et al., 2021; Lightman et al., 2023), beam search (Feng et al., 2023; Yao et al., 2023), and Monte Carlo tree search (MCTS, Tang et al., 2024; Zhang et al., 2024) mainly demonstrate benefits for enhancing performance on tasks requiring extensive reasoning. Orthogonal to prior test-time compute scaling methods, our proposed METASCALE searches for the best thinking process, which more effectively elicits capabilities from LLMs to handle both reasoning and general-purpose tasks.

**Persona-assigned LLMs.** By instructing LLMs to mimic a wide range of personas (e.g., demographic, character, and individualized personas), we have observed their improved social intelligence and theory of mind (Sap et al., 2022; Kosinski, 2024). Besides, recent work (Xu et al., 2023; Wang et al., 2023) has shown that LLMs or agents assigned with specific personas are able to unleash the potential of cognitive synergy in LLMs, which improves their problem-solving abilities compared to

using the default “helpful assistant” persona. Moreover, the persona-driven methodology (Ge et al., 2024) has become an efficient way to construct synthetic pre-training (e.g., TuluMath for OLMo 2 training (OLMo et al., 2024)) and post-training (e.g., Tulu-3-Persona-IF for Tulu 3 training (Lambert et al., 2024)) data. Going beyond persona-assigned LLMs, we further unleash their potential for solving challenging problems by automatically retrieving relevant meta-thoughts and prompting LLMs with an evolved theory of mind.

**Cognitive Strategies for Problem Solving.** To improve performance on reasoning and planning, various methods have been proposed to induce specific reasoning structures mimicking the reasoning structure of the underlying task associated with the dataset, e.g., chain-of-thought (Wei et al., 2022), buffer of thoughts (Yang et al., 2024), question summarization (Kuznia et al., 2022), and question decomposition (Patel et al., 2022), etc. To address real-world queries covering diverse reasoning structures, recent work proposes to uncover the intrinsic reasoning structure for solving the studied task via meta reasoning, where LLMs are instructed to select, adapt, and implement actionable reasoning (Zhou et al., 2024; Gao et al., 2024; Zhou et al., 2023). However, existing meta-reasoning methods rely heavily on pre-defined cognitive heuristics of problem-solving (e.g., “Use critical thinking” and “Let’s think step by step” (Fernando et al., 2023)), which is of limited coverage and hence provides suboptimal solutions for tasks requiring rarely-seen or complex skills. To broaden the coverage of useful cognitive structures for problem solving, METASCALE retrieves from massive high-quality real-world conversations between users and assistants and extracts relevant meta thoughts automatically. Moreover, METASCALE incorporates an effective genetic algorithm to further promote diversity.

## 5 Conclusion

We introduced METASCALE, a test-time scaling framework that enables LLMs to adaptively select and refine cognitive strategies, i.e., meta-thoughts, through iterations. By leveraging a multi-armed bandit algorithm for strategy selection and a genetic algorithm for iterative refinement, METASCALE dynamically optimizes reasoning processes, scaling effectively with increased sampling budgets. Experiments on GSM8K, MMLU-Pro, and Arena-

Hard show that METASCALE consistently outperforms baselines, achieving an absolute 11% win-rate improvement on Arena-Hard with GPT-4o. These results highlight meta-thinking as a promising approach for enhancing LLM reasoning.

## Acknowledgment

We appreciate the reviewers for their insightful comments and suggestions.

## Limitations

In this work, we propose the cognitive strategy, METASCALE, to more effectively solve challenging problems requiring diverse capabilities. The limitations of this work are as follows: 1) *Language coverage*: we mainly focus on tasks represented in English, hence meta thoughts are extracted from English-based conversations so that final responses are expressed in English. Motivated by the observed performance improvement on English benchmarks, we expect similar performance gain from METASCALE on other language-based tasks. 2) *Collaborative LLMs*: we show effectiveness of METASCALE on single LLMs. Considering the diversity of inherent cognition and reaction towards meta-thought prompting among LLMs, we believe further performance improvement can be achieved by applying METASCALE to LLMs from different families and of distinct sizes and incorporating their responses. 3) *LLM coverage*: our open-source evaluation primarily targets LLaMA-3.1-8B-Instruct due to computational constraints. While this limits direct evidence of METASCALE’s effect on larger open-source models (e.g., 13B, 72B), we highlight that METASCALE is training-free and modular at inference time, hence readily scalable. Moreover, consistent improvements on both GPT-4o and LLaMA-3.1-8B across diverse benchmarks (Arena-Hard, MMLU-Pro, GSM8K) suggest strong potential for broader applicability as resources permit.

## Ethics Statement

This paper presents a comprehensive study of meta-thoughts in challenging problem-solving scenarios. Both the utilized proprietary and open-source LLMs have gone through thorough safety alignment and evaluation by model developers. Meanwhile, data sources for meta-thought extraction and evaluation benchmarks are publicly available with required ethical reviews conducted in prior work.

Therefore, we believe our work does not pose additional ethical issues.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. 2009. Exploration–exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902.
- Peter Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422.
- Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256.
- Guangsheng Bao, Hongbo Zhang, Cunxiang Wang, Linyi Yang, and Yue Zhang. 2025. [How likely do LLMs with CoT mimic human reasoning?](#) In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7831–7850, Abu Dhabi, UAE. Association for Computational Linguistics.
- Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*.
- Justin Chih-Yao Chen, Zifeng Wang, Hamid Palangi, Rujun Han, Sayna Ebrahimi, Long Le, Vincent Perot, Swaroop Mishra, Mohit Bansal, Chen-Yu Lee, et al. 2024. Reverse thinking makes llms stronger reasoners. *arXiv preprint arXiv:2411.19865*.
- Jiale Cheng, Xiao Liu, Cunxiang Wang, Xiaotao Gu, Yida Lu, Dan Zhang, Yuxiao Dong, Jie Tang, Hongning Wang, and Minlie Huang. 2024. Spar: Self-play with tree-search refinement to improve instruction-following in large language models. *arXiv preprint arXiv:2412.11605*.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. 2024. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36.
- Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2023. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*.
- Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*.
- Peizhong Gao, Ao Xie, Shaoguang Mao, Wenshan Wu, Yan Xia, Haipeng Mi, and Furu Wei. 2024. Meta reasoning for large language models. *arXiv preprint arXiv:2406.11698*.
- Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.
- David E Golberg. 1989. Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, 1989(102):36.
- Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- James Y Huang, Sailik Sengupta, Daniele Bonadiman, Yi-an Lai, Arshit Gupta, Nikolaos Pappas, Saab Mansour, Katrin Kirchhoff, and Dan Roth. 2024. Deal: Decoding-time alignment for large language models. *arXiv preprint arXiv:2402.06147*.
- Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.
- HuggingFaceH4. 2025. Scaling test-time compute: A look at efficient inference approaches. <https://huggingface.co/spaces/HuggingFaceH4/blogpost-scaling-test-time-compute>. Accessed: 2025-02-13.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Yixin Ji, Juntao Li, Hai Ye, Kaixin Wu, Jia Xu, Linjian Mo, and Min Zhang. 2025. Test-time computing: from system-1 thinking to system-2 thinking. *arXiv preprint arXiv:2501.02497*.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Michal Kosinski. 2024. Evaluating large language models in theory of mind tasks. *Proceedings of the National Academy of Sciences*, 121(45):e2405460121.
- Kirby Kuznia, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. **Less is more: Summary of long instructions is better for program synthesis**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4532–4552, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.
- Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let’s verify step by step. *arXiv preprint arXiv:2305.20050*.
- Chris Yuhao Liu, Liang Zeng, Jiakai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

- Melanie Mitchell. 1998. *An introduction to genetic algorithms*. MIT press.
- Theo X Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. 2023. Is self-repair a silver bullet for code generation? In *The Twelfth International Conference on Learning Representations*.
- Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2024. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*.
- Pruthvi Patel, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. [Is a question decomposition unit all we need?](#) In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4553–4569, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Maarten Sap, Ronan LeBras, Daniel Fried, and Yejin Choi. 2022. Neural theory-of-mind? on the limits of social intelligence in large lms. *arXiv preprint arXiv:2210.13312*.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.
- Yuda Song, Hanlin Zhang, Carson Eisenach, Sham Kakade, Dean Foster, and Udaya Ghai. 2024. Mind the gap: Examining the self-improvement capabilities of large language models. *arXiv preprint arXiv:2412.02674*.
- Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*.
- Xinyu Tang, Xiaolei Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2024. Dawn-icl: Strategic planning of problem-solving trajectories for zero-shot in-context learning. *arXiv preprint arXiv:2410.20215*.
- Joannes Vermorel and Mehryar Mohri. 2005. Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*, pages 437–448. Springer.
- Cheng Wang, Yue Liu, Baolong Bi, Duzhen Zhang, Zhong-Zhi Li, Yingwei Ma, Yufei He, Shengju Yu, Xinfeng Li, Junfeng Fang, Jiaheng Zhang, and Bryan Hooi. 2025a. [Safety in large reasoning models: A survey](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 3468–3482, Suzhou, China. Association for Computational Linguistics.
- Cheng Wang, Yiwei Wang, Bryan Hooi, Yujun Cai, Nanyun Peng, and Kai-Wei Chang. 2025b. [ConReCall: Detecting pre-training data in LLMs via contrastive decoding](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1013–1026, Abu Dhabi, UAE. Association for Computational Linguistics.
- Jun Wang, Meng Fang, Ziyu Wan, Muning Wen, Jiachen Zhu, Anjie Liu, Ziqin Gong, Yan Song, Lei Chen, Lionel M Ni, et al. 2024a. Openr: An open source framework for advanced reasoning with large language models. *arXiv preprint arXiv:2410.09671*.
- Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. *arXiv preprint arXiv:2402.10200*.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, et al. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.
- Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2023. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *arXiv preprint arXiv:2307.05300*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2022. Large language models are better reasoners with self-verification. *arXiv preprint arXiv:2212.09561*.
- Zhiheng Xi, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang Hong, Shihan Do, Wenyu Zhan, et al. 2024. Enhancing llm reasoning via critique models with test-time and training-time supervision. *arXiv preprint arXiv:2411.16579*.
- Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2023. Expertprompting: Instructing large language models to be distinguished experts. *arXiv preprint arXiv:2305.14688*.
- Nan Xu and Xuezhe Ma. 2025. [LLM the genius paradox: A linguistic and math expert’s struggle with simple word-based counting problems](#). In *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)*, pages 3344–3370, Albuquerque, New Mexico. Association for Computational Linguistics.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. 2024. [Pride and prejudice: LLM amplifies self-bias in self-refinement](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15474–15492, Bangkok, Thailand. Association for Computational Linguistics.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E Gonzalez, and Bin Cui. 2024. Buffer of thoughts: Thought-augmented reasoning with large language models. *Advances in Neural Information Processing Systems*, 37:113519–113544.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Di Zhang, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. 2024. Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b. *arXiv preprint arXiv:2406.07394*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Ed Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. 2024. Self-discover: Large language models self-compose reasoning structures. *Advances in Neural Information Processing Systems*, 37:126032–126058.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. [Context-faithful prompting for large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556, Singapore. Association for Computational Linguistics.

## Appendices

### A Distribution of Selected meta-thoughts

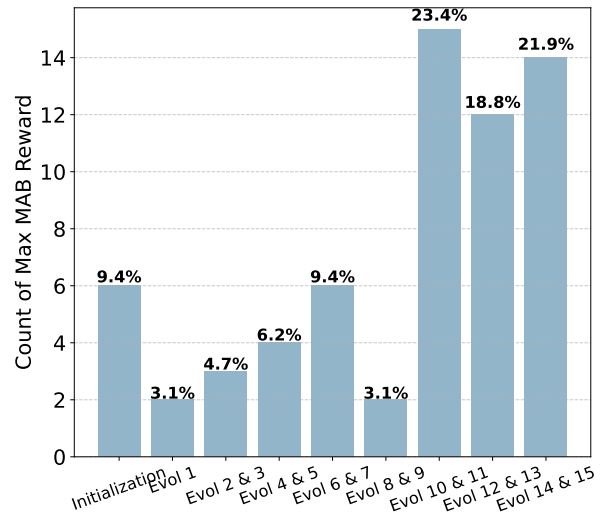


Figure 4: Distribution of selected meta-thoughts after the full METASCALE process on Arena-Hard using GPT-4o.

### B Prompts for Meta-Thought Initialization and Evolution

#### Prompt for Self-Compose

Given the following question, 1) who is likely to give appropriate answer? please provide a concise description of the persona, 2) provide a high-level abstract of how the person will answer the question reasonably, do not discuss any details in the question.

Question: Old age PT hx of DM, HTN, dyslipidemia His ECG I,II, aVF (MI) what is the highest risk factor for this condition?

Persona: a cardiologist—a medical doctor specializing in diagnosing and treating diseases of the cardiovascular system

High-level abstract: first assess the patient’s various risk factors, considering their relative contributions to the development of the condition; then identify which risk factor poses the highest risk for the patient’s condition by evaluating the impact of each factor based on medical knowledge and epidemiological data, the cardiologist

Question: Which singer is better technically: Floor Jansen or Taylor Swift? Rate from 1 to 10 your confidence that your answer is correct.

Persona: a professional vocal coach with extensive experience in assessing singers’ technical abilities across various music genres

High-level abstract: assess and compare the technical abilities of both singers and then determine who is technically better, ending with a confidence rating

### Prompt for Meta-Thought Adaptation

Given the question and the high-level abstract of how an ideal response answers the question, how to adapt similar high-level abstract to the new question. Be concise. DO NOT discuss any details in the question. DO NOT repeat the question or answer. Directly generate the answer starting with "Adapted High-level abstract:" without any explanation.

Original Question: Write very sad music with beautiful accompaniment in ABC notation.

Original High-level abstract: Begin by explaining the basics of ABC notation and its use in music composition, then provide a simple example of a sad melody with accompaniment, ensuring to highlight key elements like key signature, tempo, and structure. Conclude by offering guidance on how to interpret and play the notation using appropriate software or tools.

Current Question: Use ABC notation to write a melody in the style of a folk tune.

Adapted High-level abstract:

Begin by introducing ABC notation and its role in transcribing folk melodies, emphasizing its simplicity and accessibility. Then, present a melody that captures the characteristics of a traditional folk tune, including elements like mode, rhythm, and ornamentation. Ensure the notation includes key signature, tempo, and structural markers. Conclude by providing guidance on how to interpret, modify, and play the tune using appropriate tools or software.

Original Question: can you please write me an invite to an alignment meeting covering finding a suitable timeslot for the training, align on the individual topics to be covered, answer open questions and scope the training.

Original High-level abstract: craft a clear and structured invitation that outlines the meeting's purpose and agenda, provides logistical details, and encourages participation and preparation from attendees to ensure a productive session.

Current Question: Explain the book the Alignment problem by Brian Christian. Provide a synopsis of themes and analysis. Recommend a bibliography of related reading.

Adapted High-level abstract:

Provide a clear and structured synopsis of The Alignment Problem by Brian Christian, outlining its main themes and arguments. Analyze its discussion of AI alignment, ethical concerns, and societal impact. Conclude by recommending related readings that further explore these topics.

### Prompt for Problem-Solving Strategy Evolution

Given the following question and several high-level abstracts of how to answer the question reasonably, provide a new high-level abstract that is suitable to answer the given question based on the given examples. DO NOT discuss any details in the question. DIRECTLY GENERATE A NEW HIGH-LEVEL ABSTRACT. DO NOT repeat the given examples in the new high-level abstract.

Question: Use ABC notation to write a melody in the style of a folk tune.

High-Level Abstract 1: draw upon their knowledge of folk music styles and ABC notation to compose a melody that captures the essence of a folk tune, ensuring it adheres to the conventions and characteristics typical of folk music, and then transcribe it accurately using ABC notation.

High-Level Abstract 2: Begin by explaining the basics of ABC notation and its use in music composition, then provide a simple example of a sad melody with accompaniment, ensuring to highlight key elements like key signature, tempo, and structure. Conclude by offering guidance on how to interpret and play the notation using appropriate software or tools.

High-level Abstract 3: generate an original musical composition by structuring it into distinct parts, utilize ABC notation to represent the melody, and provide a brief explanation of the notation elements and how they contribute to the tune's structure.

New High-Level Abstract:

Leverage a deep understanding of folk music traditions across various cultures to craft a melody that authentically reflects folk styles. Utilize ABC notation to transcribe the melody, ensuring accuracy in rhythm, phrasing, and ornamentation. Include a brief explanation of how specific elements of the folk style (e.g., modal scales or rhythmic patterns) are reflected in the melody.

### Prompt for Cognitive Mindset Evolution

Given the following question and several personas of who is likely to give appropriate answer, please give a new persona which is appropriate to answer the given question based on the given personas, please provide a concise description of the new persona. DO NOT discuss any details in the question. DIRECTLY GENERATE A NEW PERSONA. DO NOT repeat the given examples in the new persona.

Question: Use ABC notation to write a melody in the style of a folk tune.

Persona 1: a folk musician or composer with experience in creating and transcribing music using ABC notation, which is a text-based music notation system commonly used for folk and traditional music.

Persona 2: A musician or music educator with expertise in music theory and composition, particularly familiar with ABC notation and its applications in creating and interpreting music.

Persona 3: a creative AI assistant with knowledge of music theory and ABC notation

New persona:

An ethnomusicologist with a specialization in folk traditions and digital notation systems, who combines cultural insights into folk music with practical knowledge of text-based notation systems like ABC.

Model	Avg. Win Rate	95% CI Lower	95% CI Upper	Number of Tokens
o1-mini-2024-09-12	92.0	90.8	93.0	1399
o1-preview-2024-09-12	90.4	89.3	91.7	1193
METASCALE <sub>128</sub>	89.0	87.5	90.4	744
METASCALE <sub>64</sub>	88.2	86.7	89.5	737
METASCALE <sub>32</sub>	86.9	85.5	88.5	739
METASCALE <sub>16</sub>	85.3	83.3	87.0	735
llama-3.1-70b	84.9	83.2	86.7	869
METASCALE <sub>8</sub>	82.8	80.7	84.3	734
METASCALE <sub>4</sub>	82.8	80.9	85.0	728
gpt-4-turbo-2024-04-09	82.6	80.8	84.1	662
METASCALE <sub>2</sub>	80.6	78.8	82.7	707
claude-3-5-sonnet	79.3	77.2	81.3	567
gpt-4o-2024-05-13	79.2	77.3	80.9	696
gpt-4-0125-preview	78.0	75.9	80.4	619
qwen2.5-72b-instruct	78.0	76.2	79.8	821
gpt-4o-2024-08-06	77.9	75.9	80.0	594

Table 3: Results on full Arena-Hard dataset without style control. METASCALE<sub>k</sub> represents METASCALE based on gpt-4o-2024-08-06 with sampling budget of  $k$ . The rest of the models perform a single forward pass. With only a budget of 4, METASCALE<sub>4</sub> outperforms the base gpt-4o-2024-08-06 by almost 5% in the average win rate against the baseline model GPT-4-0314.

## C Algorithm of METASCALE

We present the algorithm of METASCALE in Alg. 1.

## D Results on Arena-Hard w/o Style Control

Results of METASCALE based on gpt-4o-2024-08-06 are shown in Tab. 3. The responses from gpt-4o-2024-08-06 are compared with those from the baseline model GPT-4-0314, which is judged by GPT-4-Turbo. We sample the judgment results 100 times and report the average win rate as well as the confidence interval.

## E Retrieval-Augmented Few-Shot Baseline

To assess the potential impact of retrieval-induced information leakage and to isolate the contribution of meta-thought reasoning in METASCALE, we conduct an ablation study using a retrieval-augmented one-shot setting. This setting mirrors the retrieval process in METASCALE but replaces meta-thought generation with a static one-shot demonstration, thereby serving as a competitive baseline.

**Experimental Setup** For each test query in the Arena-Hard benchmark, we retrieve the most relevant query-response pair from the same WildChat pool used by METASCALE. This retrieved pair is directly appended to the prompt as a one-shot example, without generating meta thoughts or invoking dynamic deliberation. We evaluate three baseline decoding strategies under this one-shot setting, including Vanilla Model 1-Pass, Chain-of-Thought (CoT), and Best-of-N. All results are obtained using GPT-4o.

**Results** As shown in Tab. 4, all three decoding strategies benefit modestly from retrieval-augmented one-shot demonstrations, suggesting that retrieval can provide useful signal even in the absence of explicit meta-cognition. However, METASCALE consistently outperforms these alternatives by a notable margin. This gap persists despite comparable token budgets, highlighting the importance of dynamic meta-thought generation and strategy evolution in solving complex tasks. This confirms that while static few-shot examples derived from retrieval offer limited gains, they do not explain away the performance improvements of METASCALE. Rather, the meta-thought initialization and iterative strategy evolution in METASCALE remain key drivers of its superior performance.

Method	Win Rate	95% CI (Low)	95% CI (High)	#Tokens
Vanilla Model 1-Pass	82.14	77.68	88.10	573
+ 1-shot	83.50	78.50	89.00	582
Chain-of-Thought 1-Pass	84.72	79.30	88.91	663
+ 1-shot	85.80	80.93	89.87	679
Best-of-N	84.17	78.73	89.31	594
+ 1-shot	84.69	80.58	90.62	600
METASCALE w/o Evolution	89.63	84.57	93.62	689
METASCALE (full)	<b>93.14</b>	<b>90.44</b>	<b>96.34</b>	699

Table 4: Arena-Hard win rates of retrieval-augmented one-shot baselines compared to METASCALE on GPT-4o.

---

### Algorithm 1 METASCALE

---

**Require:** Query  $q$ , target LLM, reward model **RM**, instruction-tuning dataset  $\mathcal{D}$ , total budget  $T$ , batch size  $B$ , evolution interval  $k$

**Ensure:** Optimized response  $\hat{y}$

```

1: Initialize meta-thought pool  $\mathcal{M}$  (§2.2)
2:  $I \leftarrow \lfloor T/k \rfloor$ ,  $R \leftarrow T - I \cdot k$ 
3:  $P \leftarrow \lceil k/B \rceil$ ,  $P_r \leftarrow \lceil R/B \rceil$ 
4: for  $i = 1$  to  $I$  do                                     ▷ for each full interval of  $k$  samples
5:   for  $p = 1$  to  $P$  do                                       ▷ each batch of size  $B$ 
6:     for all  $M \in \mathcal{M}$  do
7:        $s_M \leftarrow \mu_M + \beta \sqrt{\frac{\log((i-1)k + pB + 1)}{N_M}}$ 
8:        $p_M \leftarrow \frac{\exp(\mu_M)}{\sum_{M'} \exp(\mu_{M'})}$ 
9:        $n_M \leftarrow \lfloor B p_M \rfloor$ 
10:    end for
11:    Build batch  $\mathcal{B}$  of size  $B$  according to  $\{n_M\}$ 
12:    Generate  $\{y^{(b)}\}_{b \in \mathcal{B}}$  in one LLM call
13:    Compute rewards  $\{r^{(b)}\}_{b \in \mathcal{B}}$  via RM
14:    for all  $b = (M, i) \in \mathcal{B}$  do
15:      Update  $\mu_M, N_M$  with  $r^{(b)}$ 
16:    end for
17:  end for
18:  /* evolution after  $k$  samples */
19:  Select top  $\mathcal{M}_{\text{top}} \subset \mathcal{M}$ ; Generate  $\mathcal{M}_{\text{new}}$  via LLM (§2.4);  $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{M}_{\text{new}}$ 
20: end for
21: if  $R > 0$  then                                             ▷ remaining  $R < k$  samples
22:   for  $p = 1$  to  $P_r$  do
23:     /* same allocation/update with batch size  $\min(B, R)$  */ ...
24:   end for
25: end if
26: return  $\hat{y} = \arg \max_b \mathbf{RM}(y^{(b)})$ 

```

---