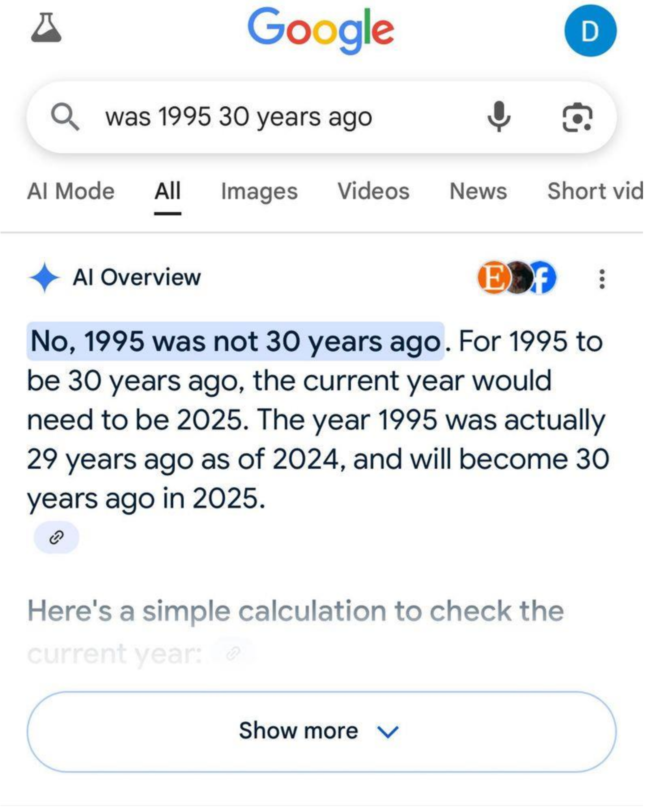
Thinking LLMs



was 1995 30 years ago

No, 1995 was not 30 years ago. For 1995 to be 30 years ago, the current year would need to be 2025. The year 1995 was actually 29 years ago as of 2024, and will become 30 years ago in 2025.

Here's a simple calculation to check the current year:

Intoduction TPO

- LLMs **lack** the basic ability of explicit **thinking before answering**

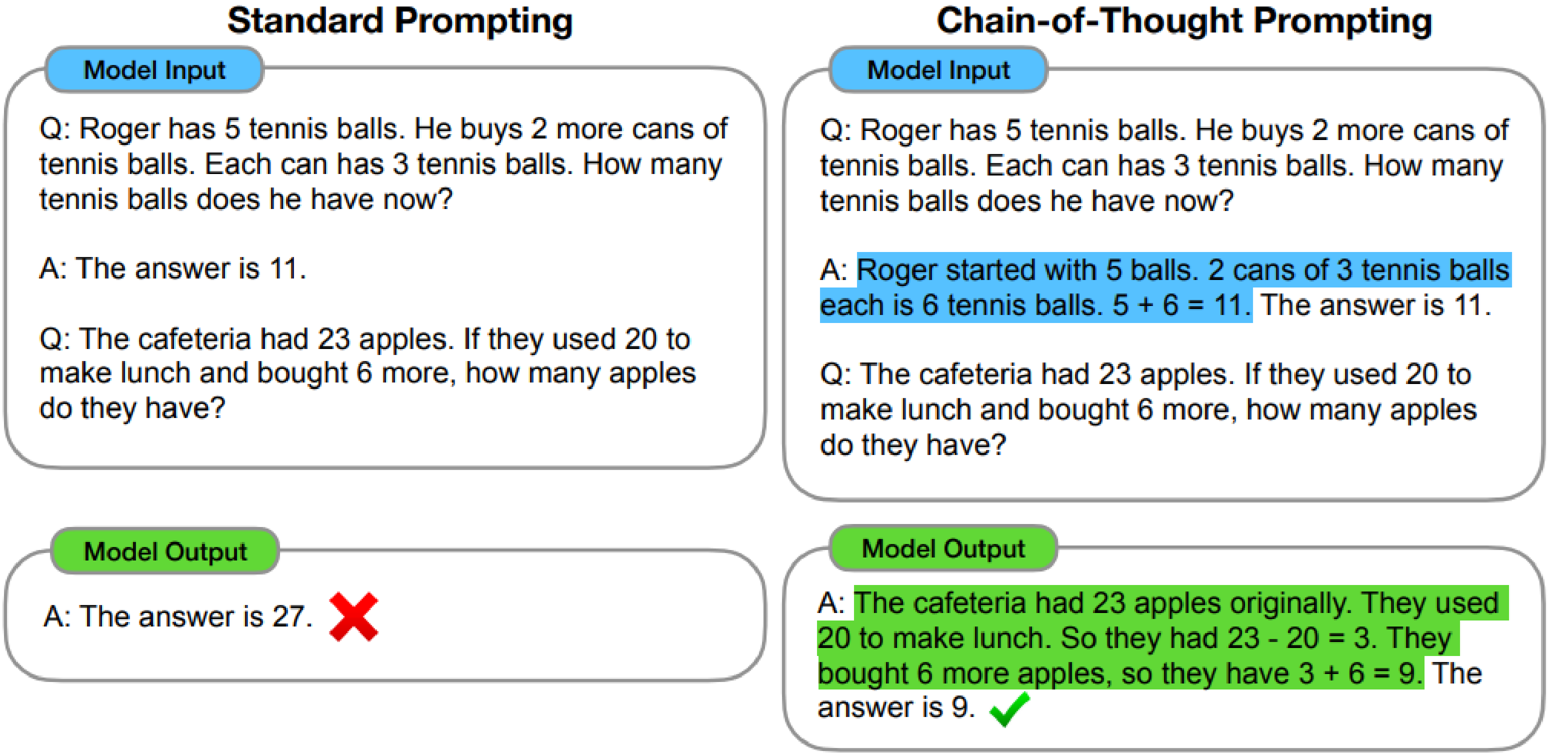
- Each token takes the same amount of compute. They have a **fixed compute budget** to generate the first response token **regardless of the instruction’s complexity**.

- One way to increase the compute budget for harder instructions is to **allow LLMs to think internally before outputting** an response. This is similar to humans who will take more time and think before answering complex questions.

- Chain-of-Thought (CoT) (Wei et al., 2022) is a widely used prompting technique that elicits such behavior.

https://arxiv.org/html/2410.10630v1

CoT few-shot



Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Figure 0

https://arxiv.org/pdf/2201.11903

CoT issues and data limitation

- However, the **usage of CoT** has been mostly **limited to math and reasoning** tasks. Meta-analysis by Sprague et al. (2024) found CoT methods to be unhelpful on tasks that do not involve math and logic.
- We argue that “**thinking**” should **have broad utility**.
- However it is challenging to train a model to think due to the **lack of supervised training data**
- **Pre-training data limited** in certain domains as **internal thoughts are often omitted** in human writing.

https://arxiv.org/html/2410.10630v1

TPO(Thought Preference Optimization)

- The goal of this paper is to **converting existing LLMs into Thinking LLMs** that work across a wide variety of tasks, **without any addition data**.
- The LLM produce an **output** sequence that can be **divided into thought and response parts**. The **thought part** is considered internal, and **not part of the response** shown to the user.
- We **optimize** this thought and response **output through iterative Reinforcement Learning** from AI Feedback (RLAIF) training
- We rely on a **standard judge model** that is trained to **evaluate responses only**, and implicitly judge the quality of the thoughts via the induced responses
- Surprisingly, we observe that **thinking** not only benefits topics like reasoning and problem solving, but also **leads to better performance on categories that are not typically considered in the reasoning domain**, such as general knowledge, marketing, and health. This opens up a new opportunity to develop Thinking LLMs aimed at general instruction following rather than specializing in more narrow technical fields.

https://arxiv.org/html/2410.10630v1

TPO training

- We start by prompting the LLM to generate thoughts before its response. After sampling different outputs, we feed the response parts to the judge model which determines the best and worst ones. Then we use the corresponding full outputs as chosen and rejected pairs for DPO optimization. We perform multiple iterations of this training.

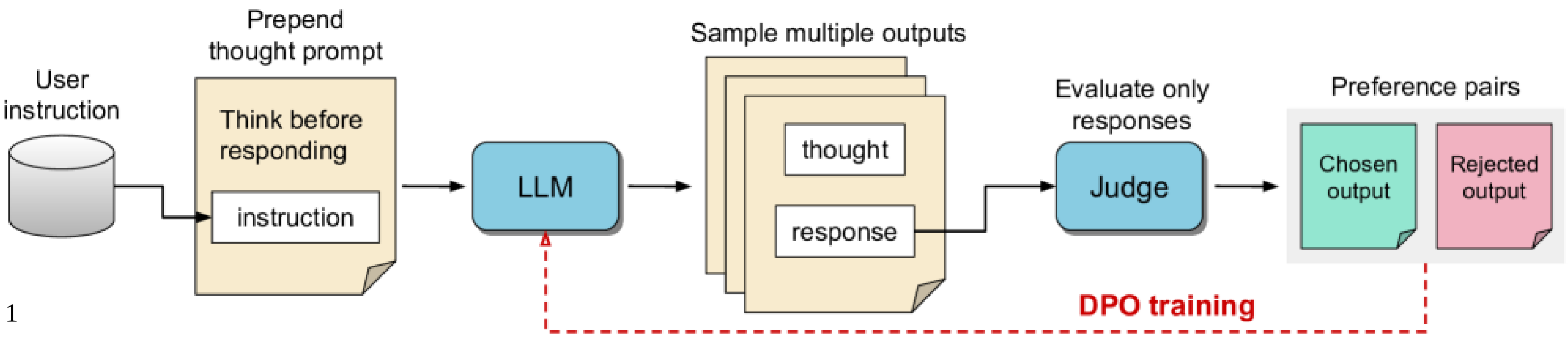
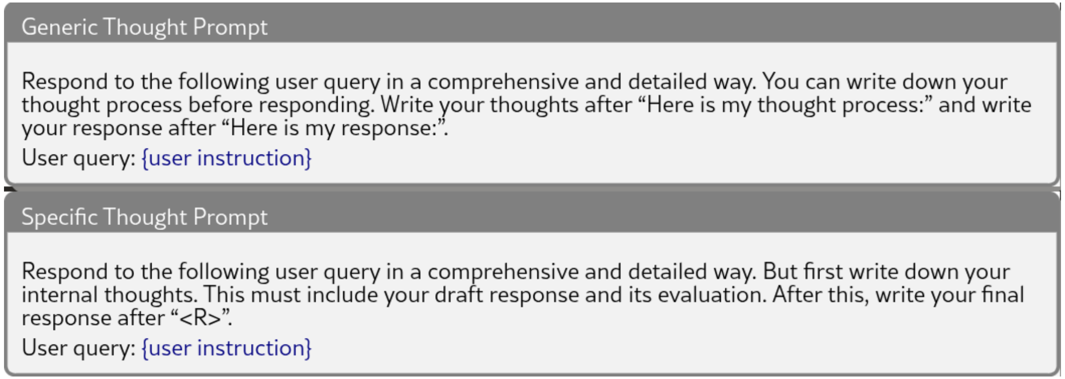


Figure 1

https://arxiv.org/html/2410.10630v1

Prompt examples


Generic Thought Prompt

Respond to the following user query in a comprehensive and detailed way. You can write down your thought process before responding. Write your thoughts after “Here is my thought process:” and write your response after “Here is my response:”.

User query: {user instruction}

Specific Thought Prompt

Respond to the following user query in a comprehensive and detailed way. But first write down your internal thoughts. This must include your draft response and its evaluation. After this, write your final response after “<R>”.

User query: {user instruction}

Figure 2

- The initial thoughts are important as they act as a starting point.

- The first thought prompt given in Figure 2 (top) is more generic and **leaves it up to the model what the thoughts will contain**.

- Specific thought prompt, given in Figure 2 (bottom), that **specifies that the thought should contain** a draft response and its evaluation.

- This **differentiates our outputs from CoT** where the **reasoning steps become part of the overall response**, sometimes without **being a clear distinction**. While it might be **useful in certain cases like solving math problems**, in general the **user expects to receive** a **response without excessive** reasoning **steps**.

- Hiding the thought part allows it to take many forms that are usually not interesting to the user: making mistakes, drafting responses and evaluating them, trying to understand the question better.

https://arxiv.org/html/2410.10630v1

Training approach

- Unlike conventional RLAIF, we will **not feed the whole model output to the judge**.
- The **judge can only see the response part** of the outputs, so the **thought part cannot influence its judgement**.
- We **chose this approach** for several reasons:
    - First, **A lack of a judge model that is capable of evaluating internal thoughts**. Building such a judge is inherently challenging because it is hard to collect human thoughts.
    - Even if such data was collected, it is **not clear if human-written thoughts will be equally useful** for LLMs.
    - Secondly, the **ultimate goal is to provide better responses** to the user. Thus, it might be better to **optimize the final objective** instead of relying on an auxiliary objective that might not align well.

https://arxiv.org/html/2410.10630v1

Training

- Ewo training instruction sets: the **synthetic instruction set generated by few-shot prompting Llama**-2-70B, and **UltraFeedback** - a more curated instruction set that **contains human instructions**.

- M_0 - Our training starts with a seed model, that is instruction-tuned to directly respond to the user instruction.

- X_i - dataset of user instructions.

- P - thought prompt

- For each input, we sample k<K outputs

$$\mathcal{M}_t(p + x_i) \rightarrow \{z_i^k, y_i^k\}$$

- $\mathcal{J}$ - Judge model for scoring. For pointwise judge models that take a single response and output a scalar score, the process is simple

$$\mathcal{J}(x_i, y_i^k) \rightarrow s_i^k \in \mathbb{R}$$

- Once we have all pairwise winners, we convert those to individual pointwise scores sik using ELO scoring

$$\text{Pair} = \{p + x_i \rightarrow z_i^c + y_i^c \; ; \; p + x_i \rightarrow z_i^r + y_i^r\} \quad \text{where} \quad c = \text{argmax}_k s_i^k \quad r = \text{argmin}_k s_i^k$$

- Using this process, the model can learn which thought led to a better response.

https://arxiv.org/html/2410.10630v1

DPO recap

- Once we have built preference pairs, **we use them with the DPO loss to train the current model** M_t. This gives us a new model M_{t+1} that will be used for the next training iteration.

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]$$

https://arxiv.org/pdf/2305.18290

Training

- It is known that some judge models tend to favor longer responses (Dubois et al., 2024; Yuan et al., 2024a). Let us define a normalization function:

$$N(l_i^k) = \big(l_i^k - \mathrm{mean}_k(l_i^k)\big) / \mathrm{std}_k(l_i^k)$$

- We recompute the scores by penalizing longer responses

$$s_i^k \leftarrow N(s_i^k) - \rho N(l_i^k)$$

The hyper-parameter ρ controls the strength of the length-control mechanism. Note we normalize both the score and the length to align them into a similar scale.

https://arxiv.org/html/2410.10630v1

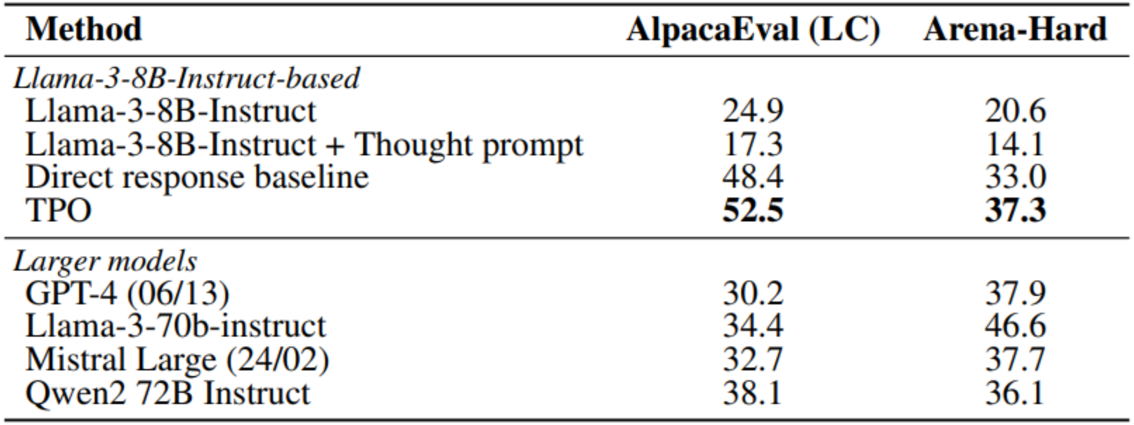
Evaluation

- As a **baseline**, we train the **same seed model that outputs responses directly without any thinking** (note, this can still perform CoT as a part of the response due to its initial instruction training). We train this baseline in the exactly same way, using the same judge, data and loss. This allows us to **directly measure the effect of the thoughts** on response quality.

- For evaluation, we use **two benchmarks**: **AlpacaEval 2** (Dubois et al., 2024) and **Arena-Hard** (Li et al., 2024). Both benchmarks perform auto-evaluation using gpt-4-1106-preview as a judge.

- The highest win rate our model **TPO achieves is 52.5%, which is +4.1% better than the direct baseline**

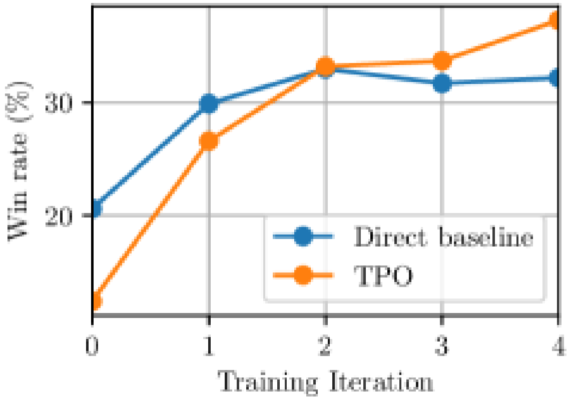
- It is also a **+27.6% incre** d GPT-4 Turbo

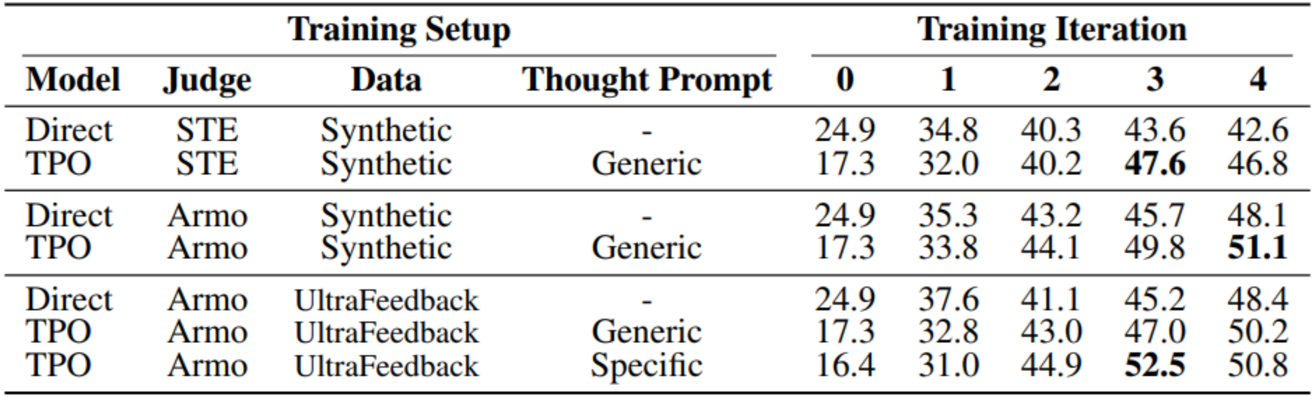


Method	AlpacaEval (LC)	Arena-Hard
Llama-3-8B-Instruct-based		
Llama-3-8B-Instruct	24.9	20.6
Llama-3-8B-Instruct + Thought prompt	17.3	14.1
Direct response baseline	48.4	33.0
TPO	**52.5**	**37.3**
Larger models		
GPT-4 (06/13)	30.2	37.9
Llama-3-70b-instruct	34.4	46.6
Mistral Large (24/02)	32.7	37.7
Qwen2 72B Instruct	38.1	36.1

https://arxiv.org/html/2410.10630v1

Alpaca Eval Results

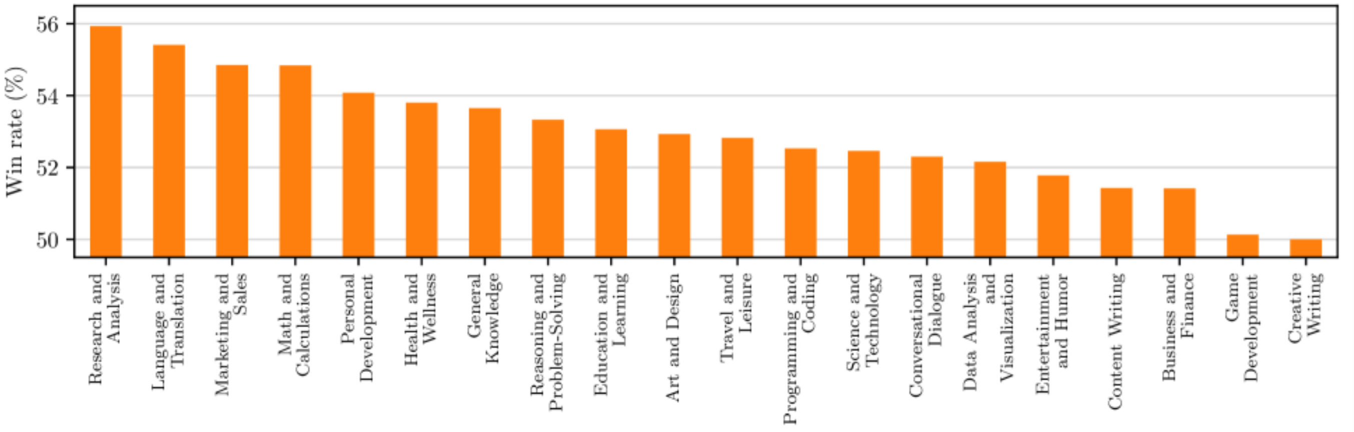




Training Setup				Training Iteration				
Model	**Judge**	**Data**	**Thought Prompt**	**0**	**1**	**2**	**3**	**4**
Direct	STE	Synthetic	-	24.9	34.8	40.3	43.6	42.6
TPO	STE	Synthetic	Generic	17.3	32.0	40.2	**47.6**	46.8
Direct	Armo	Synthetic	-	24.9	35.3	43.2	45.7	48.1
TPO	Armo	Synthetic	Generic	17.3	33.8	44.1	49.8	**51.1**
Direct	Armo	UltraFeedback	-	24.9	37.6	41.1	45.2	48.4
TPO	Armo	UltraFeedback	Generic	17.3	32.8	43.0	47.0	50.2
TPO	Armo	UltraFeedback	Specific	16.4	31.0	44.9	**52.5**	50.8

https://arxiv.org/html/2410.10630v1

Fine-grained Evaluation

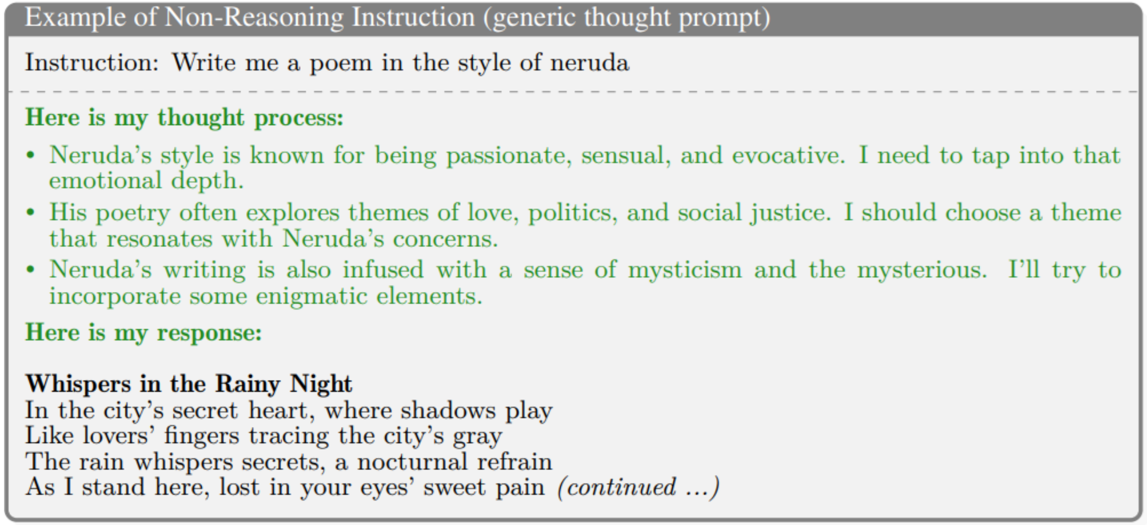


- UltrafeedBack by categories
- Compare responses generated from our **TPO model against** responses from the **direct baseline model**.
- We use the evaluation prompt from Arena-Hard and GPT4 (gpt-4-1106) as a judge
- We observe that **non-reasoning categories obtain large gains through thinking**. This includes language and translation, marketing and health.

https://arxiv.org/html/2410.10630v1

Benefits on non-reasoning task

- While writing a poem is not typically viewed as a reasoning task, it can **benefit from better planning and understanding of the instruction**.



Example of Non-Reasoning Instruction (generic thought prompt)

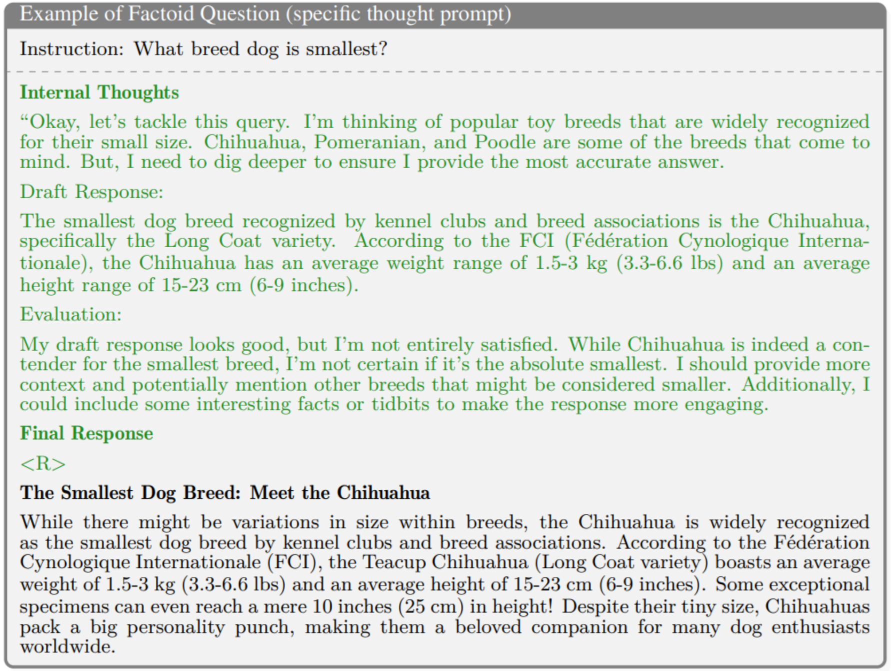
Instruction: Write me a poem in the style of neruda

Here is my thought process:

- Neruda's style is known for being passionate, sensual, and evocative. I need to tap into that emotional depth.
- His poetry often explores themes of love, politics, and social justice. I should choose a theme that resonates with Neruda's concerns.
- Neruda's writing is also infused with a sense of mysticism and the mysterious. I'll try to incorporate some enigmatic elements.

Here is my response:

Whispers in the Rainy Night
In the city's secret heart, where shadows play
Like lovers' fingers tracing the city's gray
The rain whispers secrets, a nocturnal refrain
As I stand here, lost in your eyes' sweet pain *(continued ...)*



Example of Factoid Question (specific thought prompt)

Instruction: What breed dog is smallest?

Internal Thoughts

"Okay, let's tackle this query. I'm thinking of popular toy breeds that are widely recognized for their small size. Chihuahua, Pomeranian, and Poodle are some of the breeds that come to mind. But, I need to dig deeper to ensure I provide the most accurate answer.

Draft Response:

The smallest dog breed recognized by kennel clubs and breed associations is the Chihuahua, specifically the Long Coat variety. According to the FCI (Fédération Cynologique Internationale), the Chihuahua has an average weight range of 1.5-3 kg (3.3-6.6 lbs) and an average height range of 15-23 cm (6-9 inches).

Evaluation:

My draft response looks good, but I'm not entirely satisfied. While Chihuahua is indeed a contender for the smallest breed, I'm not certain if it's the absolute smallest. I should provide more context and potentially mention other breeds that might be considered smaller. Additionally, I could include some interesting facts or tidbits to make the response more engaging.

Final Response

<R>

The Smallest Dog Breed: Meet the Chihuahua

While there might be variations in size within breeds, the Chihuahua is widely recognized as the smallest dog breed by kennel clubs and breed associations. According to the Fédération Cynologique Internationale (FCI), the Teacup Chihuahua (Long Coat variety) boasts an average weight of 1.5-3 kg (3.3-6.6 lbs) and an average height of 15-23 cm (6-9 inches). Some exceptional specimens can even reach a mere 10 inches (25 cm) in height! Despite their tiny size, Chihuahuas pack a big personality punch, making them a beloved companion for many dog enthusiasts worldwide.

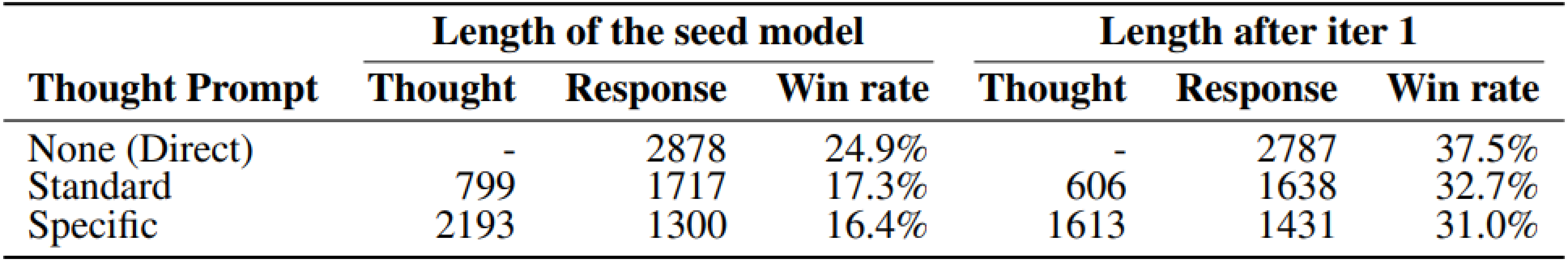
https://arxiv.org/html/2410.10630v1

Thought lengths

- The **seed model performs poorly** when it is asked to generate thoughts compared to directly generating a response, However, **after one iteration** of training, we see the **gap is shrinking**.

- Between the **two thought prompt types**, there is **not much difference** in terms of **win rate**

- The specific thought prompt has the longest thought generations. Although we **did not directly supervise the thought process**, we found the **model learns to shorten and condense the thought** throughout the training.



	Length of the seed model			Length after iter 1		
Thought Prompt	**Thought**	**Response**	**Win rate**	**Thought**	**Response**	**Win rate**
None (Direct)	-	2878	24.9%	-	2787	37.5%
Standard	799	1717	17.3%	606	1638	32.7%
Specific	2193	1300	16.4%	1613	1431	31.0%

https://arxiv.org/html/2410.10630v1

Math domain

- We evaluate model on the **GSM8K dataset** (Cobbe et al., 2021) that **contains grade-school math** word problems.

- We first test the seed model without a thought prompt, but observe it uses CoT anyway due to its instruct training, obtaining 79.2% accuracy. We thus append “Output only the number answer.” to the problems so that the model will not perform CoT in its response. It still performs surprisingly well in this answer-only mode, reaching 69.7% accuracy. However, on closer inspection we noticed that the seed model still performs CoT-like arithmetic operations like “16 - 3 - 4 = 9. 9 * 2 = 18” in its output.

- We then evaluate **direct models** trained. After training, this number drops to 51.3%. Performance is even lower for TPO models.

- However, we do see some examples where the TPO model is able to correct its mistake through reflection, while the direct model simply stops at the mistake

https://arxiv.org/html/2410.10630v1