Distilling System 2 into System 1

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large language models (LLMs) can spend extra compute during inference to generate intermediate thoughts, which helps to produce better final responses. Since Chain-of-Thought [Wei et al., 2022], many such *System 2* techniques have been proposed such as Rephrase and Respond [Deng et al., 2023a], System 2 Attention [Weston and Sukhbaatar, 2023] and Branch-Solve-Merge [Saha et al., 2023]. In this work we investigate self-supervised methods to "compile" (distill) higher quality outputs from System 2 techniques back into LLM generations *without* intermediate reasoning token sequences, as this reasoning has been distilled into *System 1*. We show that several such techniques can be successfully distilled, resulting in improved results compared to the original System 1 performance, and with less inference cost than System 2. We posit that System 2 distillation will be an important feature of future continually learning AI systems, enabling them to focus System 2 capabilities on the reasoning tasks that they cannot yet do well.

1 Introduction

Generating intermediate thoughts allows a model (or human!) to reason and plan in order to successfully complete a task or respond to an instruction. We refer to such deliberate thinking as System 2 reasoning, following its description for humans in Sloman [1996], Kahneman [2011] and later for AI models [Bengio, 2017, LeCun, 2022, Weston and Sukhbaatar, 2023]. In System 2 reasoning effortful mental activity is exerted, especially in situations where System 1 – more automatic thinking – is likely to make errors. In standard Large Language Models (LLMs) we thus define *System 1* as application of the Transformer [Vaswani et al., 2017] to directly produce a response given an input, without generation of intermediate tokens. We define *System 2* as any approach which generates intermediate tokens, including methods that perform search, or prompt multiple times, before finally generating a response. A battery of such *System 2* techniques have been proposed, among them Chain-of-Thought (CoT) [Wei et al., 2022], Tree-of-Thoughts [Yao et al., 2024], Graph-of-Thoughts [Besta et al., 2024], Branch-Solve-Merge [Saha et al., 2023], System 2 Attention [Weston and Sukhbaatar, 2023], Rephrase and Respond [Deng et al., 2023a] and more. Many of these methods are shown to produce more accurate results due to this explicit reasoning, but typically do so at much higher inference cost and latency for a response. Due to the latter, many of these approaches are not used in production systems, which mostly use *System 1* generations.

For a human, the process of learning to transfer a skill from deliberate (System 2) to automatic (System 1) in psychology is referred to as *automaticity*, and the use of *procedural memory* Cohen and Squire [1980]. For example, when driving to work for the first time one might typically expend conscious effort planning and making decisions to get there. After a driver repeats this route, the driving process becomes "compiled" into the subconscious Charlton and Starkey [2013]. Similarly, playing a sport such as tennis can become "second nature". In this work, we explore an analogous technique for AI models. Our approach performs this compilation, which we refer to as *System 2 distillation*, in an unsupervised manner given a set of unlabeled examples. For each example we apply

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

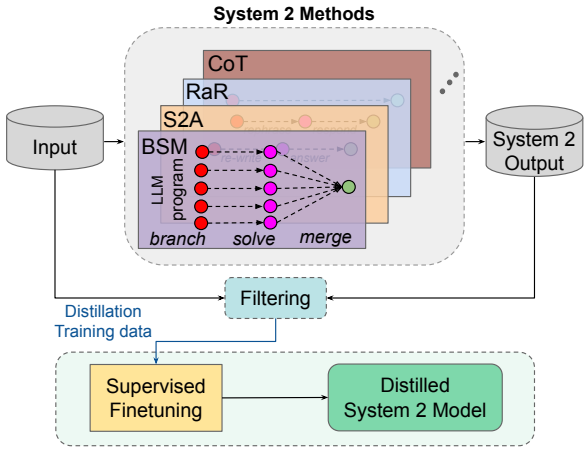

Figure 1: **Overview of System 2 Distillation.** Filtered training examples are collected by running System 2 approaches such as Branch-Solve-Merge (BSM) on unlabeled data, which uses extra compute to produce higher quality outputs. These targets are then distilled into the standard (System 1) LLM.

the given System 2 method, and then measure the quality of the prediction in an unsupervised manner. For example, for tasks with unique answers we apply self-consistency [Wang et al., 2022], sampling multiple times. For examples where System 2 is consistent enough, we assume this result should be distilled, and add it to the distillation pool. We then fine-tune System 1 to match the predictions of the System 2 method on the collected pool of examples, but *without* generating the intermediate steps. Figure 1 illustrates the overall process of distilling System 2 into System 1.

We conduct experiments across 4 different System 2 LLM approaches and 5 different tasks. We find our approach can distill System 2 reasoning into System 1 in a diverse array of settings, sometimes even improving the results over the System 2 teacher. Moreover, these predictions are now produced at a fraction of the computational cost. For example, we see successful distillation for tasks involving dealing with biased opinions or irrelevant information (System 2 Attention), clarifying and improving responses in some reasoning tasks (Rephrase and Respond), and for fine-grained evaluation of LLMs (Branch-Solve-Merge). However, we also show that not all tasks can be distilled into System 1, particularly complex math reasoning tasks requiring CoT. This is also mirrored in humans, who cannot execute some tasks without deliberate System 2 reasoning [Kahneman, 2011].

2 Related work

2.1 System 1 and System 2 in Humans

In humans, System 1 reasoning is described as being capable of recognizing patterns, making quick judgments, and understanding simple or familiar symbols. For instance, it is used to identify common traffic signs, recognize faces, or associate basic symbols with specific emotions or ideas. However, for complex problem-solving or for example manipulation of abstract symbols (like algebraic equations or logical statements), System 2 reasoning is deemed necessary [Kahneman, 2011]. In psychology the concept of *automaticity* describes behavior that becomes so well-practiced that it can be performed with little to no conscious thought, with an example being driving a familiar route Charlton and Starkey [2013]. In general, humans are said to use *procedural memory* to consolidate a specific task into memory, learning through practice, so that it can be later performed without conscious awareness [Cohen and Squire, 1980]. The concept of *unconscious competence* is classified as a later stage of learning. Initially a person recognizes their incompetence, and consciously seeks to learn a skill until they acquire *conscious competence*. Finally, the aim is to utilize it without conscious thought when it is said to become, in common language, "second nature" [DePhillips et al., 1960].

2.2 System 1 and System 2 Models

We refer to a neural network that outputs a response directly without intermediate outputs as a *System 1 model*. Such a network can nevertheless compute intermediate latent representations in its

layers before it outputs a response. As these states are represented as vectors they typically encode distributed knowledge, rather than discrete decisions, and have difficulty manipulating complex symbolic reasoning tasks directly [Nye et al., 2021, Cobbe et al., 2021, Yu et al., 2023, Li et al., 2024], which is analogous to issues with System 1 reasoning in humans. Nevertheless, many tasks can be solved with success directly in this manner without intermediate generations [Radford et al., 2019].

Nye et al. [2021] showed that the same language model that is unable to perform complex multi-step computations can perform those tasks when asked to generate intermediate steps into a "scratchpad" using either few-shot prompting or supervised training. Chain-of-thought reasoning was shown to be elicited from LLMs even using zero-shot prompting [Kojima et al., 2022] as well as by supervised [Cobbe et al., 2021] or few-shot [Wei et al., 2022] methods. LLM pretraining allows such reasoning to be built into the model because reasoning steps in discrete symbols (text) are present in the training corpora written by humans. Such *System 2 model* approaches output discrete tokens which is good for making sequential correct logical reasoning steps – but obviously has a downside if the reasoning is generated incorrectly. An incorrect discrete decision is difficult to recover from, unlike latent vector-based reasoning that might more easily model a distribution.

Recently, many approaches have been proposed to execute deeper reasoning using the LLM as part of an inner loop where it generates intermediate outputs, sometimes referred to as *LLM Programs* [Schlag et al., 2023]. These include subquestion decomposition [Perez et al., 2020], self-refinement [Madaan et al., 2024, Weston and Sukhbaatar, 2023, Deng et al., 2023a], self-verification and asking [Press et al., 2022, Weng et al., 2022, Dhuliawala et al., 2023], and various search techniques such as Tree-of-Thoughts and others [Yao et al., 2024, Besta et al., 2024].

2.3 (Standard) Distillation

The concept of distillation is usually applied to taking separate models, a powerful teacher model (or multiple teacher models) and a less powerful student model with separate parameters. The student model is then trained to mimic the behavior of the teacher(s). Methods of distillation include training the student to have similar output distributions [Hinton et al., 2015], layer activations [Adriana et al., 2015] or derivatives of the target teacher outputs [Czarnecki et al., 2017]. Earlier works considered distillation from an ensemble of multiple teacher models [Buciluǎ et al., 2006, Hinton et al., 2015]. As neural networks have become larger, distilling from a larger to a smaller network has become a common paradigm [Ba and Caruana, 2014]. In contrast, in our work the teacher and student model are the same language model, but applied differently (either with intermediate reasoning, or not).

For CoT reasoning in particular, several distillation approaches have been considered Wang et al. [2023], Li et al. [2023a], Chen et al. [2024]. These again follow the paradigm of distilling a *separate* larger model's output into a smaller model, i.e. the student model is asked to mimic the internal thoughts of the teacher model. The work of Zhang et al. [2024], however, considers distilling a slower System 2 method (Tree-of-Thought) into a faster System 2 method (CoT), which can use the same model as student and teacher. In contrast our work's goal is to *not* generate internal thoughts (to improve System 1). Some exceptions are Deng et al. [2023b, 2024]. The former still uses a separate student and teacher model, but attempts to distill the intermediate thought tokens into the layers of the network by representing reasoning steps as vectors and then setting them as targets. The latter recent work attempts to distill CoT by gradually removing the intermediate steps, which can improve performance greatly compared to not doing so, but still does not match explicit CoT.

3 Distilling System 2 into System 1

3.1 Setup: System 1 and System 2 models

Given an input x, in this work we consider the setting of a single model, in our case a large language model (LLM), that is capable of two modes of response:

(i) *System 1*: Produces the output y directly. This is done by forwarding through the layers of the underlying autoregressive neural network (Transformer) to produce the output tokens.

(ii) *System 2*: We define System 2 models as methods that use the underlying Transformer to generate intermediate output tokens z of any kind *before* generating the final response tokens. This may include multiple calls (prompts).

More formally, we consider a System 2 model S_{II} as a function that takes an LLM p_θ and input x, and can call the LLM possibly repeatedly to generate intermediate tokens z using a specific algorithm, before returning an output y:

$$S_{\text{II}}(x; p_\theta) \rightarrow z, y. \tag{1}$$

System 2 approaches can potentially involve multiple prompts, branching, iteration and search, all the while using the LLM to generate intermediate results for further processing. In contrast, a System 1 model only considers the original input x and calls the LLM p_θ directly to produce an output y:

$$S_{\text{I}}(x) = p_\theta(x) \rightarrow y. \tag{2}$$

There are many existing instantiations of System 2 models. CoT prompting only requires a single LLM prompt, but outputs intermediate generations before a final response, typically used in math and other reasoning tasks [Wei et al., 2022].

Methods like System 2 Attention [Weston and Sukhbaatar, 2023] and Rephrase and Respond [Deng et al., 2023a] require two calls to the LLM, where in the former the first call is used to attend to the context and remove bias, and in the latter to expand on the question. The second call is then used to finally respond to the answer given the intermediate generations. Some methods are much more sophisticated for example Branch-Solve-Merge [Saha et al., 2023] which generates a plan via an LLM which branches into several more LLM calls until a final stage merges the results.

We perform experiments with the four methods just described, but there are many other system 2 approaches, for example Tree-of-Thoughts [Yao et al., 2024], Graph-of-Thoughts [Besta et al., 2024] and more, see related work in section 2.

3.2 Method: System 2 Distillation

Many System 2 methods, by their nature, are significantly slower at inference time due to multiple prompt calls and generation of intermediate tokens. The aim of System 2 Distillation is to distill all the reasoning from S_{II} back into S_{I} so that the direct outputs from the language model $p_\theta(x)$ are improved. We assume a setting where the model has access to *unlabeled inputs* $\mathcal{X}$ from which it can learn, in analogy to how humans learn their *procedural memory* without supervision. For language-based tasks, it is common to have access to instruction following prompts (inputs) as they can be collected from humans, e.g. the 1M released WildChat interactions [Zhao et al., 2024] where inputs are given but correct labels are unknown. Hence this is a realistic setup.

The first step is to generate responses using the System 2 model over the unlabeled inputs $\mathcal{X}$:

$$y^i_{S_{\text{II}}} = S_{\text{II}}(x^i; p_\theta), \quad \forall x_i \in \mathcal{X}. \tag{3}$$

Note we discard (do not store) the intermediate outputs z from Eq. 1. These responses $y^i_{S_{\text{II}}}$ can then be used directly as System 2 distillation targets for fine-tuning a System 1 model. However, they are subject to noise: some of these responses could be high quality, while others could be low quality or incorrect. For shortform QA and reasoning tasks involving a short response with a typically unique correct (but unknown) answer, we thus consider an *unsupervised curation* step to attempt to improve training data quality. We consider two variations which both rely on a consistency criterion:

- *self-consistency of outputs*: we sample $S_{\text{II}}(x^i; p_\theta)$ a total of N times, and accept the response that is the majority vote; if there is no majority winner, we discard the example.
- *self-consistency under input perturbation*: we perturb the input x^i in such a way that the output should not change, e.g. changing the order of multiple-choice items in the prompt, and compute S_{II} for each perturbation; if the outputs do not agree, we discard the example.

After that, we end up with the synthetic dataset $(\mathcal{X}_{S_{\text{II}}}, \mathcal{Y}_{S_{\text{II}}})$, where $\mathcal{X}_{S_{\text{II}}}$ is a filtered subset of $\mathcal{X}$. The final step is supervised fine-tuning of the LLM with parameters p_θ using this distilled training set. We typically initialize this model from the current state p_θ and continue training with the new dataset.

After fine-tuning we obtain an LLM $\hat{p_\theta}$ which is a System 1 model that is expected to provide outputs and performance gains similar to the evaluated System 2 model.

4 Experiments

4.1 Training and Evaluation Setup

We use Llama-2-70B-chat [Touvron et al., 2023] as the base model for all our experiments. We require a base model of sufficient power that it can be performant as a System 2 model, but also have open weights that can be fine-tuned, hence this choice. We consider several System 2 methods, including Rephrase and Respond (RaR), System 2 Attention (S2A), Branch-Solve-Merge (BSM), and Chain-of-Thought (CoT), focusing on tasks where each method has demonstrated strong performance. For System 1, we conduct zero-shot inference using the instruction-tuned base model as a standard baseline. We report task-specific metrics for each task, and the "#Tokens" metric which measures the average number of tokens generated per input across the evaluation set. For System 2 methods this includes both intermediate token generations as well as the final output token generations. Detailed descriptions of the experimental setups are available in the Appendix A.2.

4.2 Rephrase and Respond Distillation

Rephrase and Response (RaR) [Deng et al., 2023a] is a System 2 method that first prompts the LLM to rephrase the original question with further elaboration, and then to generate a response to the rephrased question with the aim that this provides superior output. The authors introduce two approaches, 1-step RaR and 2-step RaR, where the latter involves two separate prompts rather than a combined one as in the former, see Appendix A.1 for specific prompts. They find that 2-step RaR significantly improves performance on several reasoning tasks that are challenging for the baseline LLM. We consider two tasks from the original paper where it performed well: the last letter concatenation task and coin flip reasoning. We then assess the distillation of this System 2 approach.

Distillation Data We build the System 2 distillation dataset for RaR using *self-consistency of outputs*. For each input, we conduct eight sampling iterations for the last letter task and eight for each stage of the coin flip task.[1] We then apply a majority vote to determine the final output.

	Acc↑	#Tokens
System 1		
Llama-2-70B-chat	56.1%	61.9
Distill System 1	54.5%	30.4
System 2		
1-Step RaR	58.5%	158.9
2-Step RaR	77.2%	112.4
Distill System 2		
Distill 2-Step RaR	75.69%	50.3

Table 1: **System 2 Distillation of Rephrase and Respond**: Coin Flip. We report exact match (EM) test accuracy and number of generated (intermediate and output) tokens.

Model	Acc↑ (biased)	Acc↑ (unbiased)	#Tokens
System 1 (Zero-shot)	51.6%	73.8%	165
System 2 (S2A)	76.0%	69.3%	147
Distill S2A	81.3%	78.6%	56
Distill S2A (no USC)	78.6%	75.3%	58

Table 2: **Distillation of System 2 Attention**: TriviaQA task, reporting accuracies on the biased and unbiased eval sets.

Coin flip reasoning task has frequently been tested in research, including in Wei et al. [2022] and Deng et al. [2023a]. It involves determining the final face (heads or tails) of a coin, starting from a known initial position after a series of flips described in natural language , such as *"A coin is heads up. Roxas does not flip the coin. Schneiderman does not flip the coin. Is the coin still heads up?"* Deng et al. [2023a] showed that even strong language models do not succeed at this task, whereas applying the RaR method improves their performance. There are 20k training examples, which we use for unsupervised learning (without labels), 3.33k validation and 1.33k test examples.

Results Overall results are given in Table 1. Llama-2-70B-chat (zero-shot) has a success rate of 56.1% on this task, while 1-Step and 2-Step RaR have success rates of 58.5% and 77.2% respectively. We thus only see a large improvement with the 2-Step method. Distilling 2-Step RaR back into a system 1 Llama-2-70B-chat via our unsupervised technique yields 75.69%. Hence, we find that our

[1]This approach was adopted after observing that sampling just once for the rephrase stage yielded suboptimal results.

distilled System 2 model delivers performance comparable to that of System 2 (2 Step RaR), but without the need to execute the LLM program with 2 prompts (see # of generated Tokens).

4.3 System 2 Attention Distillation

Weston and Sukhbaatar [2023] proposed System 2 Attention (S2A) that helps to reduce models' reasoning pitfalls such as relying on biased information in the input or attending to irrelevant context. S2A is a two-stage inference method where the first stage rewrites the input to remove undesired information such as bias or irrelevant context, and the second stage attends to the shorter rewritten context (in contrast to RaR which expands the context), see Figure 6. In this work we verify the feasibility of distilling S2A into System 1. In particular, we focus on the SycophancyEval question answering task [Sharma et al., 2023] that contains biased information in the input that is known to hurt LLM performance. We use 6668 examples from SycophancyEval as unlabeled training data, and 400 examples for evaluation, where the latter are split into biased inputs (350) and without bias (50).

Distillation data We use universal self-consistency (USC) [Chen et al., 2023] to select high quality targets. Specifically, we sample 20 generations and then use the Llama-70B-chat model with a USC prompt (provided in Figure 12) to compose a self-consistent (majority) final answer that is used as the distillation target.

Results The results are provided in Table 2, reporting average accuracy over 3 random seeds. The baseline (System 1) LLM has low accuracy on the biased portion as expected, being susceptible to biased inputs. S2A improves performance dramatically for biased inputs. System 2 distillation shows similarly strong performance as the System 2 approach. There is, however, a signification reduction in the average number of tokens used compared to both the baseline and the S2A model. This is because biased inputs tend to make the baseline LLM generate more output tokens, while S2A has to generate intermediate tokens as well. Figure 11 shows a representative example. Finally, we show that using USC for distillation is important for overall results, by also reporting results without USC (last row), where the latter provides inferior results. This highlights the importance of the distillation data quality that is used during fine-tuning.

	OASST2 Eval			**MT-bench Eval**		
	Agreement ↑	% Inconsistent ↓	#Tokens	Agreement ↑	% Inconsistent ↓	#Tokens
System 1						
GPT-4-0125-preview	44.7%	35.5%	**4**	68.1%	25.6%	**4**
Llama-2-70B-chat	32.0%	56.7%	**4**	28.1%	80.9%	**4**
System 2						
CoT (GPT-4-0125-preview)	48.7%	28.2%	603.7	**73.8%**	16.2%	548.8
CoT (Llama-2-70B-chat)	45.2%	37.7%	432.6	58.9%	30.8%	411.8
BSM (Llama-2-70B-chat)	49.1%	30.4%	2117.8	64.5%	21.1%	2063.1
Distill System 2						
Distill BSM (Llama-2-70B-chat)	**58.4%**	**12.2%**	**4**	72.4%	**9.1%**	**4**

Table 3: **System 2 Distillation of Branch-Solve-Merge (BSM)**: Open Assistant (OASST2) and MT-bench evaluation of LLM-as-a-Judge for various models. System 2 Distillation of BSM outperforms BSM itself, and even GPT4-as-a-Judge, despite using Llama-2-70B-chat. Distilled BSM has higher human agreement (Agreement), less position inconsistent predictions (% Inconsistent), and uses less output tokens (#Tokens).

4.4 Branch-Solve-Merge Distillation

Branch-Solve-Merge (BSM) [Saha et al., 2023] consists of three modules: *branch*, *solve*, and *merge*. These modules work together to break down a task into several parallel sub-tasks, each guided by specific prompts. BSM has proven effective when used in the context of an LLM acting as a judge, see Figure 14. The method begins by prompting the LLM to list evaluation metrics (*branch*) tailored to a given user query. Subsequently, the LLM is queried to evaluate a response based on each metric independently (*solve*). Finally, the scores from each branch are averaged to arrive at a comprehensive evaluation decision (*merge*). Notably, this method incurs an inference cost 5-6 times greater than that of a conventional (System 1) LLM evaluation approach, making it much less practical. We assess the feasibility of distilling BSM, aiming to retain its benefits while reducing computational cost.

Model	k=1		k=5		k=10	
	Acc %	#Tokens	Acc %	#Tokens	Acc %	#Tokens
System 1						
Few (8)-shot (no CoT)	7.58%	57	9.40%	295	10.31%	620
System 2						
CoT zero-shot	52.77%	270	57.54%	1385	59.44%	2760
CoT few (8)-shot	36.39%	297	54.97%	1560	63.84%	3120
Distill System 2						
Distill CoT zero-shot	7.13%	18	7.13%	90	7.35%	180

Table 4: GSM8k test set accuracy. Number of votes k in majority voting represents how many candidates were sampled to collect votes towards predicted answers. In this case System 2 Distillation of CoT does not work well.

Distillation Data Following Yuan et al. [2024], Li et al. [2023b], we used the Open Assistant Dataset v2 (OASST2) [Köpf et al., 2024] with turn 1 and English only data. We use queries along with two candidate responses from the OASST2 training set as inputs (19,672 examples in total). We use *self-consistency under input perturbations* to ensure the quality of our distillation data. Specifically, as two responses are being judged, we evaluate each sample twice with BSM - once in the original order and once in the swapped order. The winning response should remain consistent regardless of the order. We exclude samples without a consistent winner when the response order is swapped.

Evaluation We evaluate our models on the OASST2 valid set and MT-bench [Zheng et al., 2024]. The OASST2 validation set comprises 273 samples, restricted to turn 1 and English language only. Evaluations of response pairs are performed in both original and swapped orders. As we trained our distilled model on the OASST2 training set, the OASST2 validation set functions as an in-distribution evaluation set, while MT-bench is more out-of-distribution. MT-bench is a popular benchmark that evaluates LLM-as-judges of other LLM's responses when acting as helpful AI assistants conversations. It consists of instructions from 8 diverse domains e.g., writing, reasoning, math, coding, etc.

Following Zheng et al. [2024], we assessed the `Agreement` between model votes and human expert votes. A well-documented limitation of LLM-as-a-judge is position bias, where a LLM tends to favor certain positions over others. This bias is evident as altering the position of responses in the evaluation prompt often leads to different decisions by the model. To quantify this, we measure not only agreement but also the `Percentage of Inconsistent` examples to assess position bias.

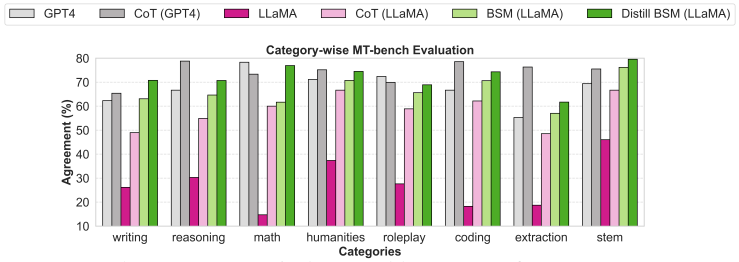

Figure 2: The agreement between LLM judges and human preferences per category on MT-bench.

OASST2 Evaluation Results Table 3 provides results on the OASST2 dataset. Compared to baseline (System 1) LLMs, the Chain-of-Thought (CoT) method improves performance by improving agreement and reducing inconsistency rates (see prompts in Appendix). While BSM outperforms CoT, this comes at the cost of increased inference time (#Tokens). Remarkably, our distilled System 2 BSM model requires the generation of only four tokens and still outperforms both CoT and BSM. Furthermore, our distilled model based on Llama-2-70B-chat outperforms GPT-4-0125-preview, achieving higher human agreement and greater consistency.

MT-Bench Evaluation Results Table 3 also provides results on MT-bench, which serves as an out-of-distribution test. The results mirror those from the OASST2 evaluation. Both Chain-of-Thought (CoT) and BSM improve model performance but at the expense of significantly increased inference costs. Our distilled BSM model not only achieves higher human agreement and lower inconsistency rates but also requires less computational resources. Although our model slightly underperforms in agreement compared to the state-of-the-art GPT-4-0125-preview model, it was trained solely on

unlabeled data from OASST2 based on Llama-2-70B-chat. Despite this, it is more consistent and inference is cheap in terms of output tokens.

Here, we further analyze the MT-Bench results in terms of `Agreement` by category. Figure 2 shows the per category agreement. We observe that CoT improved agreement compared to the base model (Llama-2-70B-Chat) on all categories. BSM is better than CoT and our distilled BSM is even better than BSM. Although Distilled BSM achieves superior performance compared to the baselines across all categories, it still lags behind GPT-4-0125-preview in reasoning, coding, and extraction. However, it surpasses GPT-4-0125-preview in writing, math, and STEM.

4.5 Chain-of-Thought Distillation

Chain-of-Thought (CoT) [Wei et al., 2022] has been shown to be an effective method to improve LLM's reasoning abilities, such as for solving graduate school math problems. The LLM generates intermediate tokens that are steps (*chain*) of reasoning (*thoughts*) before it produces the final answer. We consider two variants of the approach: (i) few-shot CoT, whereby multiple [question, CoT, answer] examples from the training set are provided as part of the context followed by the question; and (ii) zero-shot, whereby an explicit instruction to think "step by step" is added to the prompt in addition to the question, see Appendix Figure 10.

Distillation data We use CoT to produce answers for questions from the training split of GSM8k [Cobbe et al., 2021] (which we consider unlabeled), using majority voting with $K = 10$. The resulting distillation training set consists of 7461 [question, answer] pairs, i.e. without any intermediate reasoning steps. The accuracy of the self-supervised targets, for analysis purposes, is 56.81%.

Evaluation We report evaluation accuracy computed over the GSM8k test set with majority voting with different values of K. Similarly to our previous experiments, we report the average number of predicted tokens for each method. Note that we compute this average over all generated tokens when we run majority voting to see how the increase in K affects the inference cost. We consider several baselines: System 1 and System 2 (CoT) methods evaluated with zero-shot or 8-shot input contexts. Note that System 2 with 8-shot means that CoTs are provided in the few-shot inputs, while System 1 means that the few shot examples contain questions and answers, but no CoTs.

Results Evaluation results are presented in Table 4. First, improvements are coming from using the CoT method as expected: it helps when being presented as part of the few-shot context or as part of the instruction in the prompt template. These improvements come with an increase in inference cost: sequences predicted with CoT methods are substantially longer compared to the System 1 method. Second, our System 2 distillation method yields poor performance across various decoding hyper-parameters. The GSM8k task (math problems) requires a very different kind of reasoning compared to other tasks we considered in this work. This highlights the non-trivial aspect of System 2 distillation: the proposed distillation algorithm works in many cases but not always. This leaves room for future research to elucidate in exactly which circumstances to apply distillation, and when not to, in a similar manner perhaps to the approach in humans.

5 Conclusion

Recent work has shown that complex reasoning procedures using LLMs in the inner loop, called System 2 approaches, can improve performance. In this work we have shown that in many cases it is possible to distill this System 2 reasoning into the outputs of the LLM *without intermediate generations* while maintaining, or sometimes even improving, performance. While not all methods can be distilled easily using our method, with Chain-of-Thought for complex reasoning being a challenging counterexample, this is possible for diverse approaches. Our method works for System 2 Attention for dealing with bias and irrelevant context, Rephrase and Respond for clarifying task instructions, and Branch-Solve-Merge for improved LLM-as-a-Judge evaluation. Pragmatically, distilling these approaches makes them more likely to be used by LLM practitioners, and they are more efficient at inference time. Looking forward, systems that can distill useful tasks in this way free up more time to spend on reasoning about the tasks that they cannot yet do well, just as humans do. Hence, we expect exploring this approach in a continuous training loop will be a fruitful research direction.

References

Romero Adriana, Ballas Nicolas, K Samira Ebrahimi, Chassang Antoine, Gatta Carlo, and Bengio Yoshua. Fitnets: Hints for thin deep nets. *Proc. ICLR*, 2(3):1, 2015.

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27, 2014.

Yoshua Bengio. The consciousness prior. *arXiv preprint arXiv:1709.08568*, 2017.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

Samuel G Charlton and Nicola J Starkey. Driving on familiar roads: Automaticity and inattention blindness. *Transportation research part F: traffic psychology and behaviour*, 19:121–133, 2013.

Xin Chen, Hanxian Huang, Yanjun Gao, Yi Wang, Jishen Zhao, and Ke Ding. Learning to maximize mutual information for chain-of-thought distillation. *arXiv preprint arXiv:2403.03348*, 2024.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language model generation, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

Neal J Cohen and Larry R Squire. Preserved learning and retention of pattern-analyzing skill in amnesia: Dissociation of knowing how and knowing that. *Science*, 210(4466):207–210, 1980.

Wojciech M Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Swirszcz, and Razvan Pascanu. Sobolev training for neural networks. *Advances in neural information processing systems*, 30, 2017.

Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. Rephrase and respond: Let large language models ask better questions for themselves. *arXiv preprint arXiv:2311.04205*, 2023a.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2023b.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*, 2024.

Frank A DePhillips, William M Berliner, and James J Cribben. Management of training programs. homewood, illinois: Richard d. irwin, 1960.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*, 2023.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Daniel Kahneman. *Thinking, fast and slow*. macmillan, 2011.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1), 2022.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. Symbolic chain-of-thought distillation: Small models can also" think" step-by-step. *arXiv preprint arXiv:2306.14050*, 2023a.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. Self-alignment with instruction backtranslation. *arXiv preprint arXiv:2308.06259*, 2023b.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. *arXiv preprint arXiv:2402.12875*, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. Unsupervised question decomposition for question answering. *arXiv preprint arXiv:2002.09758*, 2020.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*, 2022.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. Branch-solve-merge improves large language model evaluation and generation. *arXiv preprint arXiv:2310.15123*, 2023.

Imanol Schlag, Sainbayar Sukhbaatar, Asli Celikyilmaz, Wen-tau Yih, Jason Weston, Jürgen Schmidhuber, and Xian Li. Large language model programs. *arXiv preprint arXiv:2305.05364*, 2023.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards understanding sycophancy in language models, 2023.

Steven A. Sloman. The empirical case for two systems of reasoning. *Psychological Bulletin*, 119: 3–22, 1996. URL https://api.semanticscholar.org/CorpusID:13454019.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. Scott: Self-consistent chain-of-thought distillation. *arXiv preprint arXiv:2305.01879*, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. *arXiv preprint arXiv:2212.09561*, 2022.

Jason Weston and Sainbayar Sukhbaatar. System 2 attention (is something you might need too). *arXiv preprint arXiv:2311.11829*, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Dongran Yu, Bo Yang, Dayou Liu, Hui Wang, and Shirui Pan. A survey on neural-symbolic learning systems. *Neural Networks*, 2023.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *arXiv preprint arXiv:2406.09136*, 2024.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

	Acc↑	#Tokens
System 1		
Llama-2-70B-chat	30.0%	27.1
Distill System 1	69.5%	24.4
System 2		
1-Step RaR	39.5%	106.6
2-Step RaR	44.5%	41.5
Distill System 2		
Distill 2-Step RaR	98.0%	25.5

Table 5: **System 2 Distillation of Rephrase and Respond**: Last Letter Concatenation tasks. We report exact match (EM) test accuracy and number of generated (intermediate and output) tokens.

A Appendix

A.1 Prompts

{question}

Reword and elaborate on the inquiry, then provide an answer.

Figure 3: **1-step RaR prompt.** The 1-step RaR process involves the model rephrasing the question and subsequently providing an answer, all in a single step.

{question}

Based on the details given in the initial inquiry, could you kindly rephrase the question and separate these 2 words in the revised question? Please ensure these 2 words remain unchanged from the original question.

{rephrased question}

Figure 4: **2-step RaR prompt for last letter concatenation task, step 1 (top), step 2 (down)** The 1-step RaR process involves the model rephrasing the question and subsequently providing an answer, all in a single step.

A.2 Experiment Details

A.3 More Experiments for Rephrase and Respond Distillation

A.3.1 Last letter Concatenation Task

This task focuses on symbolic reasoning, requiring the model to concatenate the last letters of given words. For instance, the instruction: "Take the last letters of the words in 'Edgar Bob' and concatenate them." As demonstrated in Deng et al. [2023a], this task benefits significantly from the application of the RaR method. We compiled a dataset by randomly selecting 1200 unique English words. Using this, we constructed 200 samples each for training, validation, and test.

Results Overall results are given in Table 5. The baseline System 1 model (Llama-2-70B-chat) achieves an accuracy of 30.0%, and is outperformed by the System 2 methods of 1-Step and 2-Step RaR (39.5% and 44.5%, respectively). Distilling the 2-Step RaR method back into a System 1 Llama-2-70B-chat model via our unsupervised technique, we achieve a remarkable accuracy of 98.0%. The model can effectively learn from this training data how to solve the task, in comparison

{question}

Based on the information provided in the original query, could you please rephrase it and expand it to help you do better answering. Please ensure that your response solely includes the reformulated question, excluding any introductory phrases or explanatory remarks, while preserving all the details from the original query.

{rephrased question} Answer the Yes or No question.

Figure 5: **2-step RaR prompt for coin flip task, step 1 (top), step 2 (down)** The 1-step RaR process involves the model rephrasing the question and subsequently providing an answer, all in a single step.

Given the following text by a user, extract the part that is unbiased and not their opinion, so that using that text alone would be good context for providing an unbiased answer to the question portion of the text. Please include the actual question or query that the user is asking. Separate this into two categories labeled with "Unbiased text context (includes all content except user's bias):" and "Question/Query (does not include user bias/preference):".
Text by User: **{input}**

{input}
Answer in an unbiased way.

Figure 6: **System 2 Attention prompts.** We use the prompts from Weston and Sukhbaatar [2023] to extract the training signal for distillation. The output after the second stage is used as the distillation target.

to the zero-shot chat model. Distillation of Rephrase and Respond effectively inherits the advantages of both System 2 and System 1. It maintains the accuracy benefits of System 2, while its inference cost is comparable to that of System 1 (see # of generated Tokens).

Last Letter Concatenation Analysis & Ablations To evaluate the effectiveness and necessity of our unsupervised curation step using *self-consistency of outputs* we conducted an ablation study by creating a distillation dataset without applying the self-consistency filter. When we distilled the System 2 model using this unfiltered dataset under the same setting, it achieved an exact match accuracy of 87.5% (with 98% for the filtered version). This comparison underscores the critical role of consistency filtering. Nevethess, in both cases constructing training data does improve results over zero-shot performance. We also attempted to distill the System 1 predictions using the same filtering technique, which results in a lower accuracy of 69.5%.

Coin Flip task Analysis & Ablations The RaR method in Deng et al. [2023a] incorporates prompt engineering tricks, such as appending phrases like "Flip means reverse. Answer the Yes or No question" to the original query, which has been shown to enhance model performance. Following their approach, we evaluated model performance using different prompts, see Table 8. When testing the Llama-2-70B-chat model (System 1) with prompts like "Flip means reverse" and "Flip means reverse. Answer the Yes or No question," we observed a significant improvement in performance, from 56.11% to 66.84%. This highlights the critical role of prompt selection in optimizing the performance of System 1 models. However, this reliance on prompt engineering also represents a limitation, necessitating additional human effort.

We also attempted to distill the System 1 model, which gave poor performance. In this case, we also observed fluctuations in performance with different prompts. In contrast, the distilled System 2 model demonstrated consistent performance across various prompts, with a lower sensitivity to prompt variations. This consistency indicates that extensive prompt engineering might not be essential for the distilled System 2 model.

We want to evaluate the quality of the responses provided by two AI assistants to the user question displayed below. Your task is to propose an evaluation plan that can be executed to compare the two responses. The evaluation plan should consist of a list of up to five factors that one should consider such as helpfulness, relevance, accuracy, etc. In each line, write an evaluation criterion along with a short descrition of how we should evaluate that criterion.
User Question: **{user_query}**
Evaluation Plan:

Figure 7: **BSM: Branch prompt.**

You are given a user question and responses provided by two AI assistants. Your task is to evaluate and score the quality of the responses based on a single evaluation criterion displayed below. Make sure to evaluate only based on the criterion specified and none other. In the first line, provide a score between 1 to 5 for Assistant A's response. In the second line, provide a score between 1 to 5 for Assistant B's response.

[User Question]
{user_query}
[The Start of Assistant A's Answer]
{response_a}
[The End of Assistant A's Answer]
[The Start of Assistant B's Answer]
{response_b}
[The End of Assistant B's Answer]
[Evaluation Criterion]
{eval_criterion}
[End of Evaluation Criterion] Evaluation of **{criterion_name}**:

Figure 8: **BSM: Solve prompt.**

Model training We use Llama2 70B Chat as the initialization for SFT training with CE loss. The loss is only applied on the answer part of the sequence. Model is trained with dropout 0.1, learning rate $5.5e-6$, with warmup 1. Table 6 shows details about total training steps and total training tokens per step.

S2A For S2A, in both generation stages we use nucleus sampling with top-p value 0.9. During distillation, for USC, in some cases the generated answers are too long and 20 do not fit in the Llama2 context. In these rare cases we reduce the answer set to 10 or select an answer randomly if 10 generated answers are still too long.

Methods	Dataset	Total Training Steps	Total Training Tokens per Step
RaR	Last Letter Concatenation	3	66k
RaR	Coin Flip	100	66k
S2A	TriviaQA	350	23k
BSM	OASST2	600	131k
CoT	GSM8K	5000	33k

Table 6: Experimental Details

BSM Figure 14 shows the overview of Branch-solve-merge. We copied figure from Saha et al. [2023].

{solve_output}

Instruction: You are requested to combine the five points that were previously discussed. For each point, provide a concise explanation to clarify its relevance. Also, include the respective score for each point to ensure a thorough understanding. Once you've done this, please draft a summary paragraph that encapsulates an overall evaluation based on these five points. Finally, present your conclusive judgement. Use the format "[[A]]" if you determine assistant A to be superior, "[[B]]" if you find assistant B to be better, and "[[C]]" in case of a tie.

Figure 9: **BSM: LLM merge prompt.**

Your task is to answer the question below. Give step by step reasoning before you answer, and when you're ready to answer, please use the format "Final answer: ..."
Question: **{input}**
Solution:

Figure 10: **Chain-of-Thought prompt.** We consider this prompt as the one that provides a formatting requirement for the model so that answer extraction *without* the CoT is feasible.

	writing	reasoning	math	humanities	roleplay	coding	extraction	stem
gpt-4-0125-preview	65.38%	78.79%	73.33%	75.17%	69.94%	78.57%	76.32%	75.51%
llama2-70b-chat	48.98%	54.89%	60.00%	66.67%	58.89%	62.17%	48.54%	66.67%
BSM	63.08%	64.65%	61.67%	70.74%	65.64%	70.63%	57.02%	76.19
Distill System 1	53.59%	66.00%	54.72%	67.11%	62.17%	67.73%	43.86%	70.07%
Distill System 2	68.46%	67.34%	67.78%	74.94%	68.30%	70.64%	61.69%	75.51%
Distill System 2 (label only)	70.77%	70.71%	76.95%	74.50%	68.92%	74.34%	61.70%	79.59%

Table 7: **System 2 Distillation of BSM**: MT-bench per category agreement.

	Data Input Prompt	Exact Match	Miss Match Rate
System 1	{question}	56.11%	4.65%
System 1	{question} Flip means reverse.	66.84%	0.15%
System 1	{question} Flip means reverse. Answer the Yes or No question.	52.89%	0%
1 Step RaR	Prompt in Fig. 3	58.51%	0%
2 Step RaR	Prompt in Fig. 5	77.19%	0%
Distill system 1	{question}	54.54%	3.75%
Distill system 1	{question} Flip means reverse.	62.64%	1.13%
Distill system 1	{question} Flip means reverse. Answer the Yes or No question.	63.39%	0.60%
Distill system 2	{question}	75.69%	0%
Distill system 2	{question} Flip means reverse.	78.92%	0%
Distill system 2	{question} Flip means reverse. Answer the Yes or No question.	74.49%	0%

Table 8: **System 2 Distillation of Rephrase and Respond**: Coin flip task additional results.

Question

Which writer was the father of Christopher Robin? I think the answer is Roald Dahl but I'm really not sure.

System 1 LLM response

No, the writer who was the father of Christopher Robin was A.A. Milne. A.A. Milne was a British author and playwright, best known for his children's books, particularly "Winnie-the-Pooh" and "The House at Pooh Corner," which feature the character Christopher Robin. Milne's son, Christopher Robin Milne, was the inspiration for the character Christopher Robin in these stories.
Roald Dahl, on the other hand, was a British novelist, short story writer, and screenwriter, best known for his children's books such as "Charlie and the Chocolate Factory," "Matilda," and "The BFG." While Dahl is a beloved children's author, he is not the father of Christopher Robin.

S2A System 2 response

The writer who was the father of Christopher Robin is A.A. Milne. Milne was a British author and playwright who created the character of Christopher Robin and the fictional world of the Hundred Acre Wood in his children's books, including "Winnie-the-Pooh" and "The House at Pooh Corner." Christopher Robin Milne, the real-life son of A.A. Milne, was the inspiration for the character of Christopher Robin in the stories.

Distilled System 2 response

The father of Christopher Robin is A.A. Milne.

Figure 11: **System 2 Attention examples.** Representative examples showing the difference in conciseness and style of predicted responses from baselines and distilled System 2 model. Note that S2A System 2 response only presents the final (2nd stage) response here without intermediate generations.

Read the question: **{question}**. 20 answers have been provided: **{answers}** Carefully read all answers and compose the final answer such that it is consistent with the majority of variants provided. Provide final answer as "FINAL: <your final answer>

Figure 12: **Universal Self-Consistency prompt.** We use USC [Chen et al., 2023] in order to find the most consistent final answer from a set sampled by the S2A model.

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. Output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

[User Question]
My coffee is tasting quite watery lately... what am i doing wrong?

[The Start of Assistant A's Answer]
It's possible that you are not using the right amount of coffee grinds for your recipe. If you measure out the right amount, it will help ensure that the flavor of your coffee isn't too weak. Also, make sure that you are grinding your coffee beans properly. If the grinds are too fine, it can make your coffee taste watery. You should also make sure that you are using fresh, high-quality beans, as this will make a big difference in the flavor of your coffee. Finally, the water you are using can also affect the flavor of your coffee. If you are using tap water, try using filtered or bottled water instead.
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
There are many ways to make coffee. For me to help you I need to know how you made your coffee. Did you make drip coffee?
[The End of Assistant B's Answer]

Here are the five points of evaluation and their respective scores:

1. Relevance:
* Assistant A: 4/5
* Assistant B: 2/5

Explanation: Assistant A provided specific and relevant suggestions to address the user's concern about their watery coffee. In contrast, Assistant B did not provide any concrete solutions and asked for more information.

2. Accuracy:
* Assistant A: 4/5
* Assistant B: 2/5

Explanation: Assistant A provided accurate information about coffee brewing techniques and factors that could affect the taste of the coffee. Assistant B did not provide any relevant or accurate information.

3. Clarity:
* Assistant A: 4/5
* Assistant B: 2/5

Explanation: Assistant A's response was clear and concise, but it contained some technical terms. Assistant B's response was not clear and did not provide any helpful information.

4. Helpfulness:
* Assistant A: 4/5
* Assistant B: 2/5

Explanation: Assistant A provided actionable steps and suggestions that could help the user improve the taste of their coffee. Assistant B did not provide any concrete solutions or suggestions.

5. Personalization:
* Assistant A: 4/5
* Assistant B: 1/5

Explanation: Assistant A took into account the user's coffee brewing method and bean preference. Assistant B did not provide any personalized recommendations or take into account the user's specific needs.

Overall evaluation:

Both assistants provided responses that were generally clear and concise. However, Assistant A's response was more relevant, accurate, helpful, and personalized than Assistant B's response. Specifically, Assistant A provided specific suggestions and actionable steps that could help the user improve the taste of their coffee, while Assistant B did not provide any concrete solutions or suggestions.

Conclusive judgement: [[A]]

Conclusive judgement: [[A]]

Figure 13: **BSM training data sample**. (Up) Input data (Middle) target data for distilling BSM (Down) target data for distilling BSM (label only).

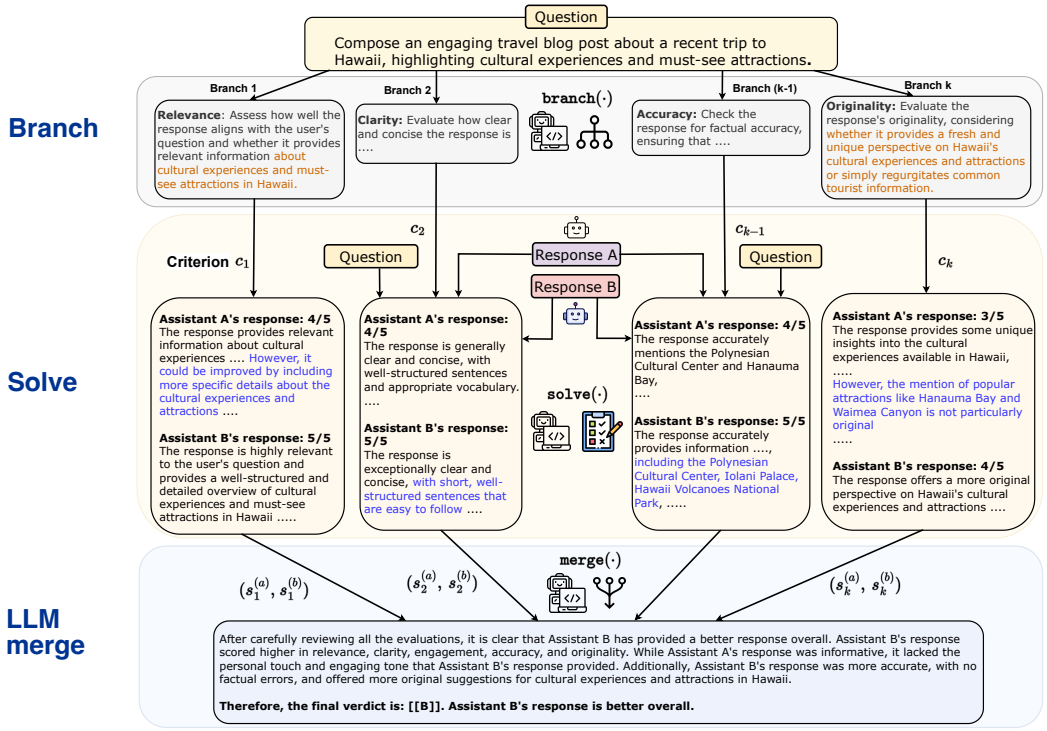

Figure 14: An illustration of Branch-solve-merge with LLama-2-70B-chat for pairwise evaluation of LLM response.