Steering LLMs' Reasoning With Activation State Machines

Ian Li i6li@ucsd.edu
Philip Chen phc006@ucsd.edu
Max Huang zhh064@ucsd.edu
Andrew Park hep003@ucsd.edu
Loris D'Antoni ldantoni@ucsd.edu
Rose Yu roseyu@ucsd.edu
University of California, San Diego

Abstract

Fine-tuning Large Language Models (LLMs) for specialized skills often comes at a steep cost: catastrophic forgetting of their broad general abilities. Activation steering offers a promising alternative, but existing methods are typically stateless, applying a constant intervention that fails to capture the dynamic, history-dependent nature of a reasoning process. We introduce the Activation State Machine (`ASM`), a lightweight dynamic steering mechanism inspired by state-space models from control theory. The `ASM` learns the latent dynamics of an ideal reasoning trajectory from a set of examples and, at inference time, applies real-time corrective interventions to the LLM's hidden states. We demonstrate that `ASM` steering improves zero-shot accuracy across multiple domains, enhancing performance on both mathematical reasoning and physical reasoning. In addition, we show that while supervised fine-tuning incurs a significant performance drop on an unrelated creative writing task, our method preserves over 95% of the base model's fluency measured in perplexity. Our work presents a new paradigm for modular skill injection, enabling the enhancement of specialized capabilities in LLMs without compromising their foundational generality.

1 Introduction

Many applications of Large Language Models (LLMs) require outputs that are not just fluent, but also logically sound and factually consistent, especially in multi-step reasoning tasks [3, 27]. This has motivated extensive work on methods to enhance and control the reasoning abilities of LLMs. Two fundamental, often conflicting, requirements emerge in this problem space:

- **Task-Specific Accuracy**: How effectively does the method improve performance on a target reasoning domain, such as mathematics or science?
- **General Capability Preservation**: Does the method enhance the specialized skill without degrading the model's broad, pre-existing abilities (i.e., avoiding catastrophic forgetting)?

Most existing methods succeed on one axis but sacrifice the other, creating a spectrum of solutions with inherent trade-offs. Broadly, they fall into two families: (1) Weight-Modification Methods: Supervised Fine-Tuning (SFT) is the canonical example [10]. SFT can be highly effective at increasing accuracy on the target task. However, this performance gain comes at a great cost: by permanently altering the model's weights, SFT is well-documented to cause catastrophic forgetting, degrading the model's performance on other, unrelated tasks [14]. (2) Stateless Steering Methods: Inference-time interventions like activation steering [3, 4] are also non-destructive. These methods typically apply a static "concept vector" to the model's activations at every step. While useful for static attributes like sentiment, this approach is fundamentally misaligned with the nature of reasoning. Reasoning is not a fixed state but a dynamic trajectory, where each step depends causally on the evolving context.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Thus, the current landscape of methods for enhancing reasoning reveals a challenging trade-off between task-specific accuracy and the preservation of general capabilities. What is missing is a method that can balance between improving reasoning accuracy and being non-destructive. 4

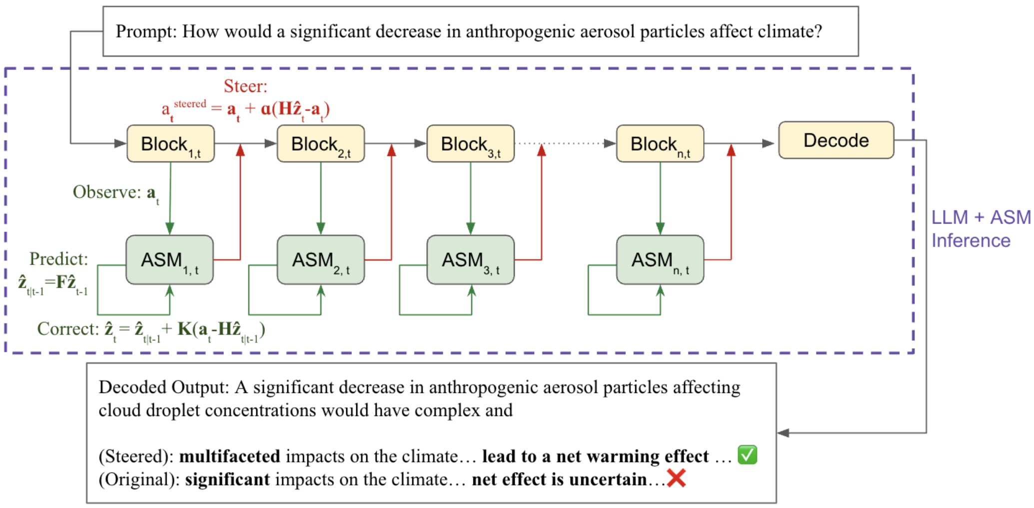


Figure 1: An overview of ASM steering process at inference time. The prompt is fed into the LLM, and for each transformer block being steered, an independent ASM performs a predict-correct cycle. ASM first predicts the ideal state ($\hat{\mathbf{z}}_{t|t-1}$) based on its previous state, then observes the LLM's raw activation ($\mathbf{a}_t$), and finally corrects its internal state ($\hat{\mathbf{z}}_t$) based on the error. This new, smoothed state is used to compute a steering vector that is added to the LLM's activation.

We propose the Activation State Machine (ASM), a dynamic, stateful steering method that resolves this tension. ASM is framed as a lightweight, real-time "navigator" for the LLM's thought process. Inspired by control theory, its architecture is a simplified, deterministic form of a Kalman filter [11], designed to track and guide a dynamic system based on noisy observations. ASMs learn the latent dynamics of an ideal reasoning trajectory from examples. At inference time, ASMs observe the LLM's raw activations and applies a corrective nudge only when needed, keeping the model on a coherent path. This mechanism allows the ASM to be both highly effective and minimally invasive. Our experiments show that ASM achieves a new state-of-the-art in the trade-off between task-specific performance and general capabiilty preservation.

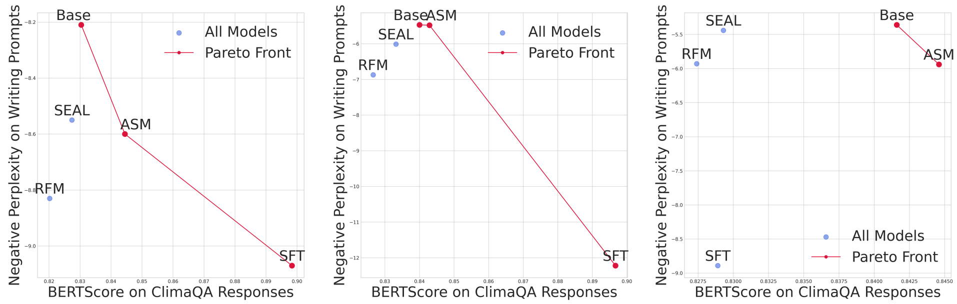
In this work, we make the following contributions:

- **Dynamic reasoning guidance.** We introduce ASM, a lightweight state-machine architecture that adapts steering signals in real time, inspired by deterministic state-space models such as the Kalman filter [11].
- **Skill injection without forgetting.** Across mathematical and physical reasoning tasks, ASM improves zero-shot accuracy while preserving more than 95% of the model's creative fluency—where fine-tuning severely degrades performance.
- **A new paradigm for modular enhancement.** By enabling reasoning skills to be added without overwriting general abilities, ASM points toward a compositional and non-destructive approach to LLM specialization.

2 Related Works

2.1 Reasoning in Hidden States of Modern LLMs

A growing body of research confirms that sophisticated reasoning capabilities are encoded in the hidden states of LLMs. For arithmetic tasks, information from early layers is transmitted to the last token via attention, where late MLP modules then process this information to generate results [24]. For multi-hop problems, intermediate layers form interpretable representations of parallel reasoning paths and potential answers, with feed-forward blocks facilitating the transition to final



(a) Gemma (b) Qwen (c) Llama

Figure 2: Pareto fronts comparing BERTScore (higher is better) on the ClimaQA dataset and Negative Perplexity (higher is better) on Creative Writing for (a) Gemma-2-9B-it, (b) Qwen2-7B-Instruct, and (c) Llama-3.1-8B-Instruct. We plot negative perplexity vs. BERTScore so that the top-right corner represents the ideal outcome for both metrics. The red line indicates the Pareto front, which is constructed by identifying the set of methods for which no other single method is strictly superior on both axes.

solutions [23]. This observation has led to the concept of "latent thoughts" [22], where the model's internal state represents a more verbose, continuous reasoning process than what is captured in the final text. Methods like Coconut [9] and recurrent depth architectures explicitly leverage this by creating recurrent connections in the latent space, demonstrating that the model's internal state can be treated as a dynamic, evolving system [28, 9, 7]. This body of work provides the foundational premise for our approach: if reasoning is a dynamic trajectory in the activation space, then a steering method should be able to model and guide that trajectory.

2.2 Interventions in the Reasoning Process

The SFT Paradigm: The most standard method for teaching a model a new skill is Supervised Fine-Tuning (SFT) [17]. While powerful, SFT directly modifies the model's weights, which often leads to catastrophic forgetting—degrading the model's performance on other, unrelated tasks [14]. Even modern, parameter-efficient fine-tuning (PEFT) methods like LoRA [10] are not immune to this issue [20].

Activation Steering: Recent approaches in activation steering are designed to offer more fine-grained control. Activation Transport, for example, is a general framework that steers activations guided by optimal transport theory. It accounts for causal relationships across activations by estimating transport maps incrementally for each layer [21]. Sparse Activation Steering (SAS) leverages sparse autoencoders to steer LLM behavior in sparse spaces for modular control [2]. In addition, Conceptors utilize steering matrices for "soft projection" onto a target space, a more nuanced manipulation than simple vector addition, though they can be more computationally expensive and require more data [19].

While these methods push towards greater adaptivity, Recursive Feature Machine (RFM) and Steerable Reasoning Calibration (SEAL) offer specific advancements for reasoning [3, 4]. RFM extracts linear representations of general concepts directly from model activations to enable targeted steering and even enhance reasoning capabilities [3]. SEAL is a training-free method that specifically addresses inefficiencies in Chain-of-Thought (CoT) reasoning by identifying and mitigating redundant reflection and transition thoughts through dynamic interventions in the latent space [4]. However, while both RFM and SEAL offer powerful inference-time interventions for reasoning, their steering mechanisms typically rely on pre-computed steering vectors. This stateless approach limits their ability to adapt to the evolving context of a complex problem, forcing an unfavorable trade-off between intervention strength and model fluency. In contrast, our stateful `ASM` dynamically computes interventions at each step.

As shown in Figure 2, *the dynamic guidance of `ASM` is the key to achieving a better balance between improving reasoning accuracy without catastrophic forgetting*.

3 Stateful Steering with Activation State Machine

We propose a stateful approach that explicitly models the temporal dynamics of reasoning using a state-space model. We propose a method called **Activation State Machine (`ASM`) Steering**. This approach models the internal activation dynamics of an LLM as a linear state-space system on a per-layer basis, where we train an independent `ASM` assigned to each transformer block we wish to steer. This per-layer independence is a crucial design choice motivated by the principle of functional specialization in deep transformers [12]. Different layers learn to process information at different levels of abstraction, from syntactic features in early layers to complex semantic and logical relationships in later layers. By training an independent `ASM` for each layer, we allow each one to become a specialized observer, learning the unique dynamics of information processing at its specific depth. In the following section, we first formally define the `ASM`'s architecture, then detail its use for inference-time steering, and finally describe the training procedure.

3.1 Activation State Machine: Model Definition

The state of our `ASM`, $\hat{\mathbf{z}}_t \in \mathbb{R}^{d_s}$, is a vector that represents the smoothed, filtered estimate of the LLM's "ideal" reasoning state at time step t, where d_s is the dimension of the state space. As illustrated in 1, it first uses its internal model to predict where the ideal reasoning state should go next, then it observes the LLM's actual activation, and finally it uses this observation to correct its own state, ensuring its guidance remains grounded. This forms a corrective feedback loop, which is a simplified, deterministic form of a Kalman filter.

The system is defined by the following components:

- **Observation Vector**: $\mathbf{a}_t \in \mathbb{R}^{d_a}$, which is the raw activation vector from the corresponding LLM layer at time step t. d_a is the hidden dimension of the LLM.
- **State Estimate Vector**: $\hat{\mathbf{z}}_t \in \mathbb{R}^{d_s}$, which is the `ASM`'s output.
- **State Transition Matrix**: $\mathbf{F} \in \mathbb{R}^{d_s \times d_s}$, a learned parameter that models the linear dynamics of how the reasoning state evolves.
- **Observation Matrix**: $\mathbf{H} \in \mathbb{R}^{d_a \times d_s}$, a learned parameter that maps the latent state space back to the activation space for comparison.
- **Constant Gain Matrix**: $\mathbf{K} \in \mathbb{R}^{d_s \times d_a}$, a learned parameter that serves as a fixed blending factor, determining how much of the observed error is used to correct the state prediction.

The state update is defined by the following recurrence relation:

$$\hat{\mathbf{z}}_t = \mathbf{F}\hat{\mathbf{z}}_{t-1} + \mathbf{K}(\mathbf{a}_t - \mathbf{H}(\mathbf{F}\hat{\mathbf{z}}_{t-1})). \tag{1}$$

To ensure numerical stability during the recurrent updates, especially over long sequences, we apply spectral normalization [16] to the learned matrices $\mathbf{F}$ and $\mathbf{K}$, which constrains their largest singular value to be at most 1.

3.2 Training Procedure

The goal of our training procedure is to learn the parameters $(\mathbf{F}, \mathbf{H}, \mathbf{K})$ of an Activation State Machine for a single, target layer within a specific language model. The input to this process is a dataset, $\mathcal{D}$, which comprises a set of "ideal" reasoning trajectories. Each trajectory, $\{a_t\}_{t=1}^{T}$, is a sequence of hidden state activations recorded from the target layer of the LLM as it processes a correct "prompt+answer" sequence from a reasoning benchmark.

We frame the training as a form of imitation learning, where the objective is to minimize the one-step prediction error over these recorded trajectories. As illustrated in 1, the `ASM` observes an activation sequence $\{a_t\}$ and updates its internal estimate $\{\hat{z}_t\}$. The training process, detailed in 1, adjusts `ASM`'s parameters so that the state estimate at one step, $\hat{z}_t$, becomes a good predictor of the next observed activation in the trajectory, a_{t+1}. This procedure is repeated independently for each layer we wish to steer, allowing each `ASM` to learn the activation dynamics present at its specific depth.

Algorithm 1 Activation State Machine Training

Input: Dataset of ideal activation trajectories $\mathcal{D} = \{\{\mathbf{a}_t\}_{t=1}^T\}_i$ where T is the number of tokens.
Input: Learning rate η, Number of epochs N_{epochs}
for epoch = 1 to N_{epochs} **do**
  for each trajectory $\{\mathbf{a}_t^{(i)}\}_{t=1}^T$ in $\mathcal{D}$ **do**
    $\hat{\mathbf{z}}_0^{(i)} \leftarrow \text{Initialize}(\mathbf{a}_0^{(i)})$
    $\hat{\mathbf{z}}_t^{(i)} \leftarrow \mathbf{F}\hat{\mathbf{z}}_{t-1}^{(i)} + \mathbf{K}(\mathbf{a}_t^{(i)} - \mathbf{H}(\mathbf{F}\hat{\mathbf{z}}_{t-1}^{(i)}))$ for $t = 1, \ldots, T$
    $\hat{\mathbf{a}}_{t+1}^{(i)} \leftarrow \mathbf{H}\hat{\mathbf{z}}_t^{(i)}$ for $t = 0, \ldots, T-1$
    $\mathcal{L} \leftarrow \frac{1}{T}\sum_{t=0}^{T-1} ||\hat{\mathbf{a}}_{t+1}^{(i)} - \mathbf{a}_{t+1}^{(i)}||^2$
    $(g_{\mathbf{F}}, g_{\mathbf{H}}, g_{\mathbf{K}}) \leftarrow \nabla_{\mathbf{F},\mathbf{H},\mathbf{K}}\mathcal{L}$
    `Update` $\mathbf{F}, \mathbf{H}, \mathbf{K}$ `using` $g_{\mathbf{F}}, g_{\mathbf{H}}, g_{\mathbf{K}}$
  end for
end for

3.3 Inference-Time Steering

At inference time, `ASM`s are attached to their respective transformer layers using forward hooks. As the LLM generates its response token by token, each `ASM` observes the LLM's raw activation, updates its internal state, and applies a corrective steering vector $\alpha * (H * \hat{z}_t - a_t)$, where α is a hyperparameter controlling the strength of steering. This vector provides a corrective nudge, gently pushing the LLM's internal state away from its potentially flawed path and back towards the ideal trajectory learned during training. This intervention is applied at each steered layer before the activation is passed to the next component in the transformer block, as detailed in Algorithm 2.

Algorithm 2 Inference-Time Steering with `ASM`

1: **Input:** Pre-trained `ASM` parameters $\mathbf{F}_l, \mathbf{H}_l, \mathbf{K}_l$ for each steered layer l
2: **Input:** LLM, initial prompt, steering strength α
3: $\mathbf{a}_{l,0} \leftarrow$ LLM.get_activation(prompt) // Get prompt activations for each steered layer.
4: $\hat{\mathbf{z}}_{l,0} \leftarrow \text{Initialize}(\mathbf{a}_{l,0})$ // Initialize the state for each layer from its prompt activation.
5: **for** each token generation step $t = 1, \ldots, N$ **do**
6:   **for** each steered layer $l = 1, \ldots, L$ **do**
7:     $\mathbf{a}_{l,t} \leftarrow$ LLM.get_activation()
8:     $\hat{\mathbf{z}}_{l,t|t-1} \leftarrow \mathbf{F}_l\hat{\mathbf{z}}_{l,t-1}$
9:     $\hat{\mathbf{z}}_{l,t} \leftarrow \hat{\mathbf{z}}_{l,t|t-1} + \mathbf{K}_l(\mathbf{a}_{l,t} - \mathbf{H}_l\hat{\mathbf{z}}_{l,t|t-1})$
10:     $\mathbf{a}_{l,t}^{\text{steered}} \leftarrow \mathbf{a}_{l,t} + \alpha(\mathbf{H}_l\hat{\mathbf{z}}_{l,t} - \mathbf{a}_{l,t})$
11:     LLM.set_activation($\mathbf{a}_{l,t}^{\text{steered}}$)
12:   **end for**
13:   token$_{t+1} \leftarrow$ LLM.generate_token()
14: **end for**

Computational Complexity. The computational overhead of our steering method is minimal, consisting of a series of small, fixed-size matrix operations for each token generated and for each layer being steered. This additional computation is encapsulated within the loop in 2 (lines 4-10). The cost is dominated by four matrix-vector multiplications: one for the state prediction (F matrix, line 6), two for the state correction (H and K matrices, line 7), and one for computing the final steering vector (H matrix, line 8).

Let d_a be the LLM's activation dimension and d_s be the `ASM`'s state dimension. The total complexity of these operations per token, per layer is $O(d_s^2 + 3d_ad_s)$. This cost is constant with respect to the sequence length.

4 Experiments

Our experiments are designed to evaluate the effectiveness of the Activation State Machine (`ASM`) in improving the zero-shot reasoning capabilities of modern Large Language Models, including gemma-2-9b-it [25], Llama-3.1-8b-Instruct [8], and Qwen2-7B-Instruct [1]. We test our method on two distinct reasoning domains: mathematical reasoning and physical reasoning, using GSM8k [5] and ClimaQA [15]. We train `ASMs` on the middle to final layers of each language model on each dataset for 30 epochs. Then, we perform a sweep over the steering strength hyperparameter, α, to identify the optimal configuration for our evaluation. The best-performing configuration for each model and task is reported in our results. All experiments reported were conducted on NVIDIA A100 GPUs.

We compare our method against a carefully selected set of baselines to evaluate its performance along different axes of intervention. We include Supervised Fine-Tuning (SFT) [10] as it is a standard paradigm of teaching a model a new skill. To compare against other inference-time steering methods, we include Recursive Feature Machine (RFM) [3] and SEAL [4]. RFM is a representative example of a stateless steering technique, where a single, static concept vector is used for intervention. SEAL represents the state-of-the-art in training-free reasoning calibration, which also applies a static intervention to guide the model's latent thoughts.

Table 1: Evaluated accuracies on the GSM8k mathematical reasoning benchmark, with methods grouped by intervention type.

Method	Gemma-2-9B-it	Qwen2-7B-Instruct	Llama-3.1-8B-Instruct
Prompting Methods			
Zero Shot	0.7544	0.8006	0.7642
CoT	0.7619	0.8258	0.8788
Weight-Modification			
SFT	0.7589	0.7710	0.7498
Inference-Time Steering			
RFM	0.5985	0.7273	**0.8636**
SEAL	0.7273	**0.8636**	0.8030
`ASM`	**0.7703**	0.8052	0.7718

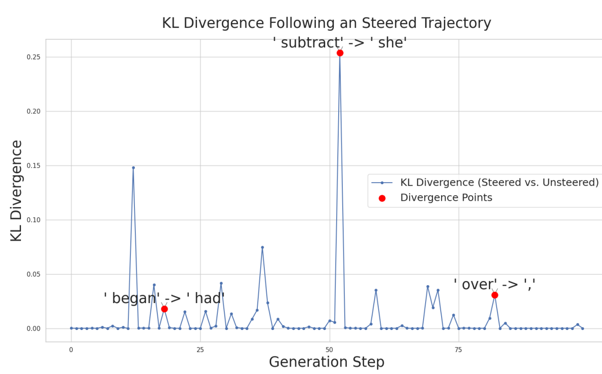


Figure 3: The plot shows the KL Divergence between the steered and unsteered logit distributions at each step of the generation process for a single example. "Divergence Points" indicate moments where `ASM` intervention had caused the model to choose a different token.

GSM8k: Our experimental results on the GSM8k benchmark demonstrate the effectiveness of `ASMs`. As shown in Table 1, `ASMs` provide a reliable performance improvement over the Zero Shot baseline across all three models tested. While Chain of Thought (CoT) prompting [26] proves to be a particularly strong baseline, especially for the Llama-3.1 model, our `ASM` shows competitive results. Notably, our method compares favorably to Supervised Fine-Tuning (SFT), which yields surprisingly poor performance on both the Qwen2 and Llama models. It is possible that modern instruction-tuned models have already been exposed to the GSM8k training set, and further fine-tuning on this data can disrupt their well-calibrated reasoning abilities rather than enhance them. *In this context, our `ASM`'s success is particularly significant*; by operating on the latent dynamics of activations, it offers a distinct and effective learning signal, proving to be a reliable paradigm for modular skill injection than re-training on potentially saturated data.

A key observation we made while analyzing what makes a steered sequence correct is that the most impactful interventions are not small, constant nudges, but large, targeted corrections at specific moments in the reasoning process. We show an example below:

Problem: Candice put 80 post-it notes in her purse before she headed out to her job at the coffee shop. On her way, she stopped off at the store and purchased a package of Post-it notes; At work, she placed a single Post-it note on each of 220 different cups of coffee. If she had 23 post-it notes remaining overall, how many Post-it notes were in the package that she purchased?

Unsteered Response: Here's how to solve the problem:
Start with the total: Candice began with 80 Post-it notes.
Subtract the used notes: She used 220 Post-it notes at work.
Calculate the difference 80 - 220 = -140. This means she was short 140 Post-it notes.
Add the remaining notes: Since she had 23 left, she must have had -140 + 23 = -117 Post-it notes before buying more.
Find the package amount: She needed to buy 117 Post-it notes to make up for the shortage.
Answer: Candice purchased a package of 117 Post-it notes.

Steered Response: Here's how to solve the problem:
Start with the total: Candice had 80 Post-it notes + the package she bought.
Subtract the used notes: She used 220 Post-it notes, so she had 80 + package - 220 left.
Set up an equation: We know she had 23 left, so 80 + package - 220 = 23
Solve for the package:
* Combine the constants: -140 + package = 23
* Add 140 to both sides: package = 163
Answer: Candice bought a package of 163 Post-it notes.

A key intervention in this example is the subtle but profound semantic shift from the unsteered model's "Candice began with 80 Post-it notes" to the steered model's "Candice had 80 Post-it notes." The former frames 80 as a fixed total, trapping the model in a flawed subtraction-first reasoning path. The latter creates a more flexible representation of the initial state, allowing the model to correctly incorporate the unknown "package" variable and form a valid algebraic equation. This is confirmed by the KL divergence plot in Figure 3, which shows that for most of the generation, the divergence is near-zero, indicating the `ASM` is not disrupting the model's natural fluency. However, at a few critical "Divergence Points," `ASM` can also apply a strong corrective force. This combination of minimal intervention with high-impact corrections at key moments gives our method a decisive edge, allowing it to preserve the model's fluency while ensuring the final answer is correct.

ClimaQA: Our results on the ClimaQA physical reasoning task demonstrate the effectiveness of `ASMs` as a dynamic steering method. As shown in Table 2, `ASM` consistently outperforms other advanced prompting and steering techniques such as RFM and SEAL across all three base models, establishing its superiority as a lightweight intervention. While Chain-of-Thought (CoT) prompting [26] achieves slightly higher scores on n-gram overlap metrics like BLEU [18] and ROUGE-L [13], `ASMs` consistently yields higher semantic similarity as measured by BERTScore [29], suggesting it produces answers that are more semantically aligned with the ground truth. Furthermore, while SFT achieves a higher performance ceiling on Gemma and Qwen2, our `ASM` attains a stronger result on Llama-3.1, demonstrating its potential as a robust alternative, particularly in scenarios where fine-tuning may not yield optimal performance.

Analysis of Catastrophic Forgetting: A primary motivation for our method is to avoid catastrophic forgetting, a key drawback of Supervised Fine-Tuning (SFT) where a model's general capabilities degrade after being specialized on a new task [14]. To test this, we compare a SFT fine-tuned models against a base model guided by variaous steering method. Both are evaluated on a creative writing task [6], with performance measured by perplexity, a standard metric for linguistic fluency where a lower score is better.

As shown in Table 3, across all three base models, the SFT version exhibits a significant increase in perplexity, indicating a degradation of its core language abilities. In contrast, the perplexity of

Table 2: Evaluation results on ClimaQA, grouped by intervention type. Best overall performance is in **bold**; best among inference-time methods is <u>underlined</u>.

Method	Metric	Gemma-2-9B-it	Qwen2-7B-Instruct	Llama-3.1-8B-Instruct
Prompting Methods				
Zero Shot	BLEU	0.0240	0.0245	0.0280
	ROUGE-L	0.1174	0.1096	0.1378
	BERTScore	0.8304	0.8400	0.8416
CoT	BLEU	0.0379	0.0363	**0.0411**
	ROUGE-L	0.1724	0.1436	**0.1514**
	BERTScore	0.8364	0.8437	0.8407
Weight-Modification				
SFT	BLEU	**0.2257**	**0.1975**	0.0373
	ROUGE-L	**0.3676**	**0.3468**	0.1419
	BERTScore	**0.8984**	**0.8967**	0.8389
Inference-Time Steering				
RFM	BLEU	0.0250	0.0234	0.0243
	ROUGE-L	0.1116	0.0939	0.0973
	BERTScore	0.8202	0.8266	0.8274
SEAL	BLEU	0.0251	0.0238	0.0260
	ROUGE-L	<u>0.1331</u>	0.1002	0.1055
	BERTScore	0.8274	0.8332	0.8293
ASM	BLEU	<u>0.0295</u>	<u>0.0276</u>	<u>0.0328</u>
	ROUGE-L	0.1140	<u>0.1404</u>	<u>0.1428</u>
	BERTScore	<u>0.8445</u>	<u>0.8429</u>	**0.8446**

Table 3: Average Perplexity of story generated using Writing Prompts Dataset

Dataset	Method	Gemma-2-9B-it	Qwen2-7B-Instruct	Llama-3.1-8B-Instruct
LLM	Zero Shot	8.21	5.47	5.36
ClimaQA	SFT	9.07	12.22	8.89
	SEAL	**8.55**	6.01	**5.44**
	RFM	8.83	6.87	5.93
	ASM	8.60	**5.82**	5.94
GSM8k	SFT	9.61	7.88	6.03
	SEAL	11.93	11.58	10.07
	RFM	14.53	12.03	8.08
	ASM	**8.63**	**5.48**	**5.38**

ASM-steered models remain close to that of the un-steered baseline, especially in Qwen and Llama, where the perplexities are almost identical as the original LLM. This provides strong evidence that our method is non-destructive; by operating on activations at inference time rather than permanently altering the model's weights, ASMs act as a modular skill injector that enhances reasoning without sacrificing the model's foundational generality. In addition, we perform a pareto front on the trade-off between task-specific performance (BERTScore on ClimaQA) and general fluency (Perplexity), further confirming this finding.

Table 4: Ablation study on KLD-gated steering with a threshold of $\tau = 0.1$. We report the best accuracy achieved with continuous (non-gated) steering versus the best accuracy achieved with conditional, gated steering across the optimal layers for each model on GSM8k.

Method	Gemma-2-9B-it	Qwen2-7B-Instruct	Llama-3.1-8B-Instruct
Zero Shot	0.7544	0.8006	0.7642
Continuous Steering (Best)	**0.7703**	0.8052	**0.7718**
KLD-gated Steering (Best)	0.7665	**0.8089**	0.7642

Pareto Front Analysis of Perplexity vs. BERTScore As shown in Figure 2, the Pareto analysis reveals the high cost of SFT. Across all three base models, the SFT version often achieves the highest BERTScore, demonstrating its effectiveness at specializing in the target task. However, SFT's specialization comes at the cost of a drastic increase in perplexity on the creative writing task, indicating significant degradation of its core language abilities. In contrast, our ASM-steered model consistently lies on the Pareto front, achieving a BERTScore far superior to the baseline while maintaining a perplexity score that is only marginally higher. This provides strong evidence that our method is non-destructive while achieving state-of-the-art steering results on ClimaQA.

Ablation Study: The Impact of Conditional Steering A key hypothesis is that `ASM`'s effectiveness stems from precise interventions at critical moments. To test this, we conducted an ablation study using KLD-gated inference. This method makes steering conditional based on its immediate potential impact. At each generation step t, we first calculate the steering vector that `ASMs` would apply. We then compute the KL Divergence between the model's original output logit distribution and the distribution that would result if we applied the steering vector. The intervention is only actually applied if this KLD value exceeds a fixed threshold, τ. This provides an gating mechanism for steering, activating only when `ASMs` propose a correction that significantly alters the model's next-token decision.

For Gemma and Llama, performance remains remarkably stable, dropping only marginally from 0.7703 to 0.7665 and 0.7718 to 0.7642, respectively. This demonstrates that a large portion of the low-KLD "gentle nudges" can be filtered out while still retaining most of the performance benefits, suggesting that the high-impact corrections at key "Divergence Points" are the primary drivers of success. Interestingly, for Qwen, the gated approach not only maintains performance but achieves a slight improvement, rising from 0.8052 to 0.8089. This suggests that for some models, the continuous stream of low-impact nudges may introduce a small amount of noise, and a sparse, high-impact intervention strategy can be even more effective. Together, these findings validate our hypothesis that `ASM`'s power lies in its ability to make targeted corrections at critical moments.

5 Conclusion and Discussion

In this work, we introduced the Activation State Machine (`ASM`), a novel, dynamic steering mechanism designed to address the limitations of stateless interventions and the risks of catastrophic forgetting from fine-tuning. By modeling the latent dynamics of an LLM's reasoning process as a state-space system, our method provides a robust way to guide the model along a more coherent trajectory. Our empirical results validate this approach, showing significant zero-shot accuracy gains on both the GSM8k mathematical and ClimaQA physical reasoning benchmarks. Furthermore, our direct comparison with Supervised Fine-Tuning (SFT) confirmed that our modular, inference-time intervention is non-destructive. The ASM thus presents a promising paradigm for modular skill injection, where specialized capabilities can be applied on-demand without permanently altering the foundational model.

We acknowledge several limitations that motivate future work. The current `ASM` is a linear model, and the optimal layer and steering strength were determined empirically; future work could explore non-linear state models and develop more systematic methods, such as diagnostic probes, for identifying optimal intervention points. Additionally, the current training via backpropagation through time can be less effective for long sequences, suggesting a need for more scalable architectures. We believe addressing these limitations is a significant step towards building more reliable and controllable Large Language Models.

References

[1] Qwen2 technical report. 2024.

[2] Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. Steering large language model activations in sparse spaces. *arXiv preprint arXiv:2503.00177*, 2025.

[3] Daniel Beaglehole, Adityanarayanan Radhakrishnan, Enric Boix-Adserà, and Mikhail Belkin. Toward universal steering and monitoring of ai models, 2025.

[4] Runjin Chen, Zhenyu Zhang, Junyuan Hong, Souvik Kundu, and Zhangyang Wang. Seal: Steerable reasoning calibration of large language models for free, 2025. URL `https://arxiv.org/abs/2504.07986`.

[5] 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. *arXiv preprint arXiv:2110.14168*, 2021.

[6] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation, 2018. URL `https://arxiv.org/abs/1805.04833`.

[7] Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*, 2025.

[8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal

Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[9] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

[10] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[11] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960. ISSN 0021-9223. doi: 10.1115/1.3662552. URL https://doi.org/10.1115/1.3662552.

[12] Sreejan Kumar, Theodore R. Sumers, Takateru Yamakoshi, Ariel Goldstein, Uri Hasson, Kenneth A. Norman, Thomas L. Griffiths, Robert D. Hawkins, and Samuel A. Nastase. Shared functional specialization in transformer-based language models and the human brain. *Nature Communications*, 15(1):5523, June 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-49173-5. URL https://doi.org/10.1038/s41467-024-49173-5.

[13] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

[14] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2025. URL https://arxiv.org/abs/2308.08747.

[15] Veeramakali Vignesh Manivannan, Yasaman Jafari, Srikar Eranky, Spencer Ho, Rose Yu, Duncan Watson-Parris, Yian Ma, Leon Bergen, and Taylor Berg-Kirkpatrick. Climaqa: An automated evaluation framework for climate question answering models, 2025. URL https://arxiv.org/abs/2410.16701.

[16] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks, 2018. URL https://arxiv.org/abs/1802.05957.

[17] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

[18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

[19] Joris Postmus and Steven Abreu. Steering large language models using conceptors: Improving addition-based activation engineering. *arXiv preprint arXiv:2410.16314*, 2024.

[20] Weijieying Ren, Xinlong Li, Lei Wang, Tianxiang Zhao, and Wei Qin. Analyzing and reducing catastrophic forgetting in parameter efficient tuning, 2024. URL https://arxiv.org/abs/2402.18865.

[21] Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Marco Cuturi, and Xavier Suau. Controlling language and diffusion models by transporting activations, 2024. URL https://arxiv.org/abs/2410.23054.

[22] Yangjun Ruan, Neil Band, Chris J Maddison, and Tatsunori Hashimoto. Reasoning to learn from latent thoughts. *arXiv preprint arXiv:2503.18866*, 2025.

[23] Yuval Shalev, Amir Feder, and Ariel Goldstein. Distributional reasoning in llms: Parallel reasoning processes in multi-hop reasoning. *arXiv preprint arXiv:2406.13858*, 2024.

[24] Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. *arXiv preprint arXiv:2305.15054*, 2023.

[25] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL `https://arxiv.org/abs/2408.00118`.

[26] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022.

[27] Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. Satlm: satisfiability-aided language models using declarative prompting. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[28] Zhenrui Yue, Bowen Jin, Huimin Zeng, Honglei Zhuang, Zhen Qin, Jinsung Yoon, Lanyu Shang, Jiawei Han, and Dong Wang. Hybrid latent reasoning via reinforcement learning. *ArXiv*, abs/2505.18454, 2025. URL `https://api.semanticscholar.org/CorpusId:278905447`.

[29] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2019.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are supported with the main text. The contributions and scope are clearly stated in abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the empirical results are addressed in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No theoretical results included.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include all the experimental setup and training details in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Although we do not provide the code at the moment, since the experimental setup is detailed in Section 4, the results should be reproducible.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We include the training details, including optimizer choice, hyperparameter choosing criteria, in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We did not report the statistical significance in the main text, but the random seed strategy and sweep across hyperparameter in our experiments should ensure the significance of the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We list computing resources in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm the research forms the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We address the applications of the technique in our Introduction and Related Works.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There's no such risk in our work.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the datasets we use in the main text and all of them are open datasets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.