Prototype-Based Dynamic Steering for Large Language Models

Ceyhun Efe Kayan **Li Zhang**
Drexel University
cek99,harry.zhang@drexel.edu

Abstract

Despite impressive breadth, LLMs still rely on explicit reasoning instructions or static, one-fits-all steering methods, leaving a gap for adaptive, instruction-free reasoning amplification. We present Prototype-Based Dynamic Steering (PDS), a test-time method that amplifies large language model (LLM) reasoning without adding or altering instructions. We introduce "reasoning prototypes" by clustering activation differences between Chain-of-Thought (CoT) and neutral prompts. At inference, an input's hidden state is projected onto these prototypes to form an instance-specific steering vector. Evaluated on GSM8K, AQuA-RAT, and BIG-Bench tasks, PDS consistently improves accuracy without fine-tuning or prompt engineering. Notably, the gains persist even when CoT is explicitly suppressed to improve cost-efficiency, indicating that the intervention strengthens latent reasoning processes rather than inducing a superficial behavioral shift. These results position dynamic, prototype-guided steering as a lightweight alternative to training-time approaches for enhancing LLM reasoning.

1 Introduction

The performance of Large Language Models (LLMs) for different tasks across various domains has been remarkable (Brown et al., 2020; Touvron et al., 2023). Their ability to generate human-like text, answer questions, and perform complex reasoning tasks has been proven many times. However, their black-box nature poses challenges for safety, reliability, and alignment. A critical research area is the development of methods for steering which is the ability to guide a model's output towards desired attributes (truthfulness, specific tones, jailbreak etc.) or away from undesired ones (toxicity, gender bias etc.) (Wang et al., 2025; Konen et al., 2024; Postmus and Abreu, 2025; Lamb et al., 2025).

Traditional methods for steering, such as fine-tuning, are computationally expensive, require large datasets and can cause catastrophic forgetting of other desired behaviors (Ouyang et al., 2022; Rafailov et al., 2024). A more recent and lightweight alternative is the use of steering vectors (Turner et al., 2023; Postmus and Abreu, 2025). These vectors, when added to internal activations at specific layers, can effectively shift the model's behavior in a targeted manner. With in-context learning, prior work typically constructs these vectors from the difference in activations elicited by carefully chosen pairs with contrast on a specific semantic attribute, such as sentiment, formality, or reasoning presence. These prompts are usually handcrafted or selected from curated datasets to contrast specific behavioral attributes, such as helpfulness, conscientiousness or politeness (Konen et al., 2024; Yang et al., 2025).

The most widely used technique for constructing steering vectors is **difference-of-means (DoM)**, which defines the steering vector as the difference between the means of the contrastive prompts from the training set. While effective, static techniques like difference-of-means apply the same intervention to every example in a test set regardless of context. A single difference-of-means vector captures an average behavioral shift but cannot adapt to the nuanced requirements of different problems within the same domain. For instance, different mathematical reasoning problems may benefit from different reasoning strategies, yet current methods apply identical steering across all inputs.

In this paper, we propose a robust and data-driven approach for constructing steering vectors during inference time, referred to as **Prototype-Based Dynamic Steering**. As described in Figure 1 instead of using static steering vectors, we learn a set of "prototype" vectors that represent different fine-grained target behaviors from the training set examples. In reasoning tasks specifically,

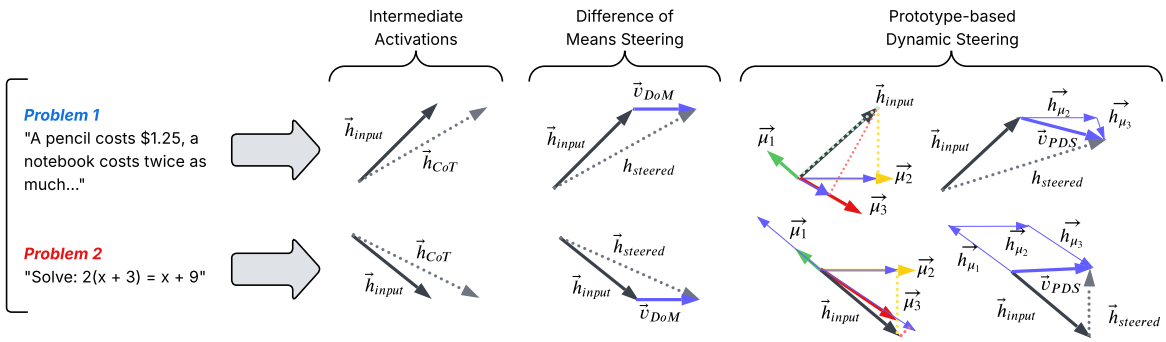

Figure 1: **Overview of technical differences between Difference-of-Means and proposed Prototype-Based Dynamic Steering**. Instead of using a single vector for each sample like DoM, PDS leverages the projections of input activations onto prototypes to compute the steering vector, which is then injected into the model's residual stream to enhance reasoning behavior.

those could be various strategies happening during models' step-by-step reasoning. We discover these prototypes by clustering activation differences between Chain-of-Thought (CoT) prompts (Wei et al., 2023) and neutral prompts, where different clusters may capture distinct reasoning strategies embedded in the activation space. After learning reasoning prototypes, for any new input, we construct a unique, tailored steering vector at inference-time by summing the projections of input activations onto prototypes which allows each prototypes to contribute to the construction of the steering vector according to their match with the ongoing reasoning context. The resulting vector provides a context-aware representation of the ongoing reasoning process at the beginning of the decoding. This way, each test example is guided by its custom steering vector, instead of a one-size-fit-all one in the standard difference-of-means technique.

Our approach is grounded in a fundamentally different view of behavioral steering. Traditional methods such as DoM assume that complex behaviors can be shifted by a single, universal direction in activation space, a "behavioral uniformity" view that considers variation in reasoning approaches as noise. In contrast, we argue that reasoning is not monolithic but arises from combinations of multiple distinct cognitive strategies, each occupying a structured subspace of intermediate activations. PDS is designed to leverage this diversity: by projecting input activations onto a basis of reasoning prototypes, it performs input-responsive steering — dynamically selecting the optimal blend of strategies for the specific problem at hand. This perspective reframes steering as problem-tailored enhancement rather than one-size-fits-all intervention, and it underpins every design choice behind PDS.

To evaluate the effectiveness of our method, we apply Prototype-Based Dynamic Steering across three prompting strategies: three prompting strategies regarding CoT: encouraging (CoT), discouraging (Anti-CoT), and making no special requirement (Neutral). In the Neutral setting, our method successfully induces CoT-like behavior and improves accuracy, demonstrating its ability to enhance reasoning by shifting the model toward step-by-step problem-solving. In the Anti-CoT case, where prompts explicitly discourage step-by-step reasoning, our method does not override such discouragement to evoke CoT, but it still leads to a substantial performance gain, suggesting that steering can also improve reasoning quality without altering the high-level behavior. These findings indicate that our method supports reasoning enhancement through two complementary pathways: (i) behavioral shift toward CoT-style reasoning, and (ii) activation-level refinement that improves performance without extra tokens.

In summary we propose a novel method for discovering behavioral prototypes by clustering activation difference vectors, a dynamic steering mechanism that constructs an input-specific steering vector during inference time by projecting the current activations onto the learned prototypes. We show that PDS outperforms DoM on the **GSM8k** (Cobbe et al., 2021a), **AQuA-RAT** (Ling et al., 2017) and **BIG-Bench** (Srivastava et al., 2022) benchmarks.

2 Related Work

A growing body of work (Turner et al., 2023; Subramani et al., 2022; Rimsky et al., 2024; Li et al., 2023; Arditi et al., 2024; Chen et al., 2025; Liu

et al., 2024; Hernandez et al., 2024; Zou and colleagues, 2025; Lee et al., 2025; Li et al., 2025) shows that steering vectors enable adjustable control of large language models without altering model weights. We situate our approach within four complementary research directions.

Foundations of Activation Steering The linear representation hypothesis holds that semantic features correspond to linear directions in transformer activation spaces (Park et al., 2025; Cunningham et al., 2023). Early explorations confirm that model behavior, such as sentiment, can be modulated via directed activation interventions (Tigges et al., 2023). Subramani et al. (2022) initiates latent steering by extracting meaningful activation differences, and Turner et al. (2023) formalizes activation addition for steering model behavior. Rimsky et al. (2024) advances this via Contrastive Activation Addition (CAA), averaging differences between contrasting data pairs for reliable alignment steering. Recent geometric and norm-preserving techniques such as Householder Pseudo-Rotation (Pham and Nguyen, 2024) and Angular Steering (Vu and Nguyen, 2025) address issues of instability in additive steering, offering smoother and more precise behavior control.

Behavioral Steering via Difference-of-Means Building on contrastive activation ideas, several studies compute steering vectors by taking differences in mean activations across paired datasets (Turner et al., 2023; Rimsky et al., 2024; Li et al., 2023). For example, Li et al. (2023) uses sparse linear probes to identify truth-oriented directions, achieving substantial performance gains on TruthfulQA (Lin et al., 2022). Similarly, Zou et al. (2025) extracts activation vectors representing abstract targets like honesty and emotions using this contrastive paradigm. Arditi et al. (2024) identifies and manipulates a single refusal direction, impacting safety-labeled model behavior. The SEAL method (Chen et al., 2025) further refines CoT output steering by dampening transitional and reflective reasoning steps, improving results on GSM8K and Math500 benchmarks (Cobbe et al., 2021a). Moreover, newer contrastive prompting approaches such as Contrastive Prompting (Yao, 2025) and Learning from Contrastive Prompts(Li et al., 2024) improve reasoning by explicitly eliciting correct and incorrect chains. Spectral Editing of Activations (SEA) (Qiu et al., 2024) addresses high variance in difference-of-means methods by constructing robust steering directions via covariance-based subspace projection.

Alternative Steering Interventions Beyond residual-based activation addition, steering methods also operate on attention activations and across full transformer layers. Liu et al. (2024) demonstrates that attention-layer interventions can steer toxicity and stylistic traits. Hernandez et al. (2024) explores fine-grained knowledge manipulations to inspect or edit internal representations. Studies like Zou and colleagues (2025) directly optimize steering vectors using single examples with gradient-based methods, leading to high success in suppressing unwarranted refusal behavior across safety benchmarks like HarmBench (Mazeika et al., 2024). Lee et al. (2025) treat each transformer's layer bias term as a trainable vector, leveraging this structure to match or exceed fine-tuned refusal control without weight updates.

Prototype-Based & Sparse Autoencoder Methods Prototype-driven frameworks add interpretability and causal clarity to steering. Sparse autoencoder (SAE)-based methods also extract feature directions and monosemantic latent features from raw CoT traces or residual activations, enhancing reasoning by overcoming polysemanticity and aiding precise behavioral intervention through more interpretable latent features (Li et al., 2025; Cunningham et al., 2023). These approaches help pinpoint which latent features directly cause behavior and can be manipulated for targeted control. Studies further consolidates sparse-autoencoder techniques for interpreting and steering LLMs (Shu et al., 2025). Prototype-based interpretability has shown promise beyond vision applications: NLP extensions of prototypical networks demonstrates that models could learn inherently interpretable embeddings during fine-tuning, capturing representative features for downstream tasks while maintaining competitive performance and transparency at both token and sample levels (Xie et al., 2023). Mechanistic and geometric analyses reveal structured, often orthogonal subspaces in LLM representations, underpinning many steering techniques that treat high-level behaviors as latent directions (Geva et al., 2021; Park et al., 2025).

Positioning of Our Work We build on these insights but depart in two key ways. First, instead of a single difference-of-means vector, we learn a *prototype set*: a small basis that captures diverse

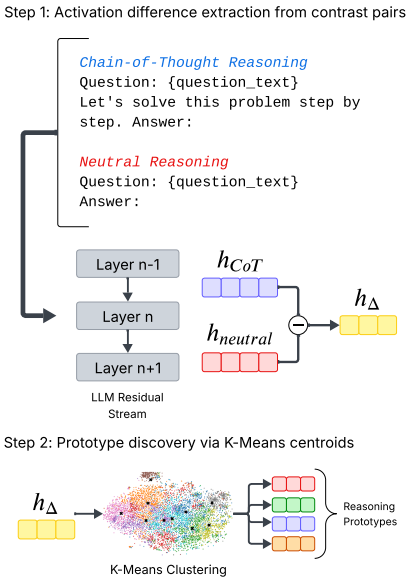

Figure 2: **Overview of prototype discovery in Prototype-Based Dynamic Steering**. Activation differences are extracted for the contrast input pairs. Centroids obtained after K-Means clustering are treated as reasoning prototypes.

facets of the target behavior, making the intervention more interpretable and robust. Second, by projecting inference-time activations onto learned prototypes, the steering vector for the input is constructed as a function of the information related to the problem at hand. Collectively, the literature demonstrates that steering vectors are an emerging technique for interpretable LLM alignment. Our contribution extends this paradigm by introducing prototype-based steering.

3 Methods

Our method consists of three stages: (1) collecting activation difference vectors representing CoT inducing behavior, (2) discovering a set of "reasoning prototypes" by clustering these difference vectors, and (3) using these prototypes to construct an input-specific steering vector for each input dynamically, during inference. We provide theoretical justification for each component and detailed analysis of the geometric properties of the resulting prototype space.

Activation Difference Collection The foundation of our approach rests on the hypothesis that behavioral transformations in neural networks can be captured as directions in activation space (Park et al., 2025; Cunningham et al., 2023). For reasoning, we seek to identify the transformation from neutral problem-solving to explicit step-by-step reasoning while preserving the nuanced variations that different problem types require. For each training example x_i from a particular dataset's training split, we generate two carefully designed prompt variants that isolate the reasoning transformation:

- $p_{i,CoT}$: The CoT prompt, e.g.,"Question: [problem text]... Let's think step by step. Answer: "
- $p_{i,neutral}$: The neutral prompt, e.g., "Question: [problem text]... Answer:"

The CoT prompt explicitly encourages step-by-step reasoning through the instruction phrase, while the neutral prompt represents the model's default problem-solving approach without explicit reasoning guidance. This design ensures that activation differences can be attributed specifically to the reasoning transformation rather than confounding task-specific variations. We process both prompts through the model and extract the final token's hidden state activation h_l at layer l. We empirically determine that layer 16 (out of 32 total layers) provides optimal signal for reasoning behaviors, consistent with prior work suggesting middle layers capture higher-level semantic processing (Liu et al., 2023; Skean et al., 2024; Zheng et al., 2025). From these activations, we compute a set of N activation difference vectors:

$$D = \{d_1, d_2, ..., d_N\}$$

$$d_i = h_l(p_{i,CoT}) - h_l(p_{i,neutral})$$

Each vector d_i represents a specific guidance for reasoning transformation for problem x_i. Crucially, these vectors does not simply capture the **average** shift toward reasoning like DoM, but encode **each specific** problem's idiosyncratic characteristics that may include solution strategies.

Prototype Discovery via Clustering Traditional difference-of-means approach computes the average activation difference vector, effectively discarding the rich diversity present in the set of activation differences. Our central insight is that this diversity contains valuable information about distinct reasoning strategies that should be preserved and leveraged rather than averaged away. We apply k-means clustering (MacQueen, 1967; Lloyd, 1982) to discover latent structure in the difference vector space:

$$\{\mu_1, \mu_2, ..., \mu_k\} = \text{k-means}(D, k)$$

The choice of k-means clustering is motivated by several theoretical and practical considerations. K-means groups activation differences into clusters that correspond to distinct problem solving strategies. The centroid of each cluster represents an average activation shift towards the associated reasoning style, making it usable as a targeted steering direction. The number of clusters k is determined via the elbow method, using the within-cluster sum of squares criterion. Each cluster centroid μ_j represents a reasoning prototype - a canonical direction in activation space corresponding to a specific reasoning strategy.

Theoretical Justification for Prototype-Based Steering We provide theoretical motivation from complementary perspectives to answer the question "Why should prototype-based clustering capture meaningful behavioral structure?"

1. **Geometric Perspective:** If reasoning behaviors occupy a structured subspace in activation space, then clustering activation differences should recover the principal directions of this subspace. The success of our method provides empirical evidence that mathematical reasoning indeed occupies such a structured region, validating the linear representation hypothesis for complex cognitive behaviors. (Park et al., 2025; Cunningham et al., 2023).
2. **Cognitive Perspective:** Mathematical problem-solving involves multiple strategies that share core computational principles but differ in execution details. Clustering activations identifies these strategic variations, enabling dynamic selection of appropriate reasoning approaches based on problem-specific characteristics.
3. **Information Theoretic Perspective:** Rather than collapsing the information in activation differences to a mean vector, clustering preserves diversity while reducing dimensionality from N individual examples to k interpretable prototypes, balancing information retention with computational efficiency.

Dynamic Steering via Projection Our steering mechanism is fundamentally dynamic and input-adaptive. For inference on a new problem, we first pass the input prompt through the model (without generation) to obtain activations h_{input} of the final input token at layer l. This vector encodes the model's initial "understanding" of the problem before generation begins. We then construct a custom steering vector v_{steer} by projecting h_{input} onto our learned prototype set:

$$v_{steer}(h_{input}) = \sum_{j=1}^{k} \text{proj}_{\mu_j}(h_{input})$$

This projection operation exhibits 3 crucial properties:

- **Adaptive Weighting:** The magnitude of the constructed steering vector is computed based on the alignment between each prototype and the current input's activation pattern. Problems naturally activating reasoning-related representations produce larger weights for relevant prototypes, inducing an adaptive weighting of the relevant prototype.
- **Subspace Preservation:** The sum of projections yields a vector lying entirely within the subspace spanned by the prototypes, ensuring steering remains within the reasoning manifold rather than venturing into semantically irrelevant directions.
- **Compositional Strategy Selection:** Each projection weight indicates the contribution of a specific reasoning strategy to the final steering vector, enabling interpretable analysis of which cognitive approaches the model considers relevant for each problem.

The steering intervention modifies activations via:

$$h'_{input} = h_{input} + \alpha \cdot v_{steer}(h_{input})$$

where α is a hyperparameter controlling steering strength. We apply this intervention selectively to the first output token only, maximizing impact on reasoning initiation while minimizing interference with subsequent generation dynamics.

Condition	GSM8k			Aqua-Rat		
	Baseline	**DoM**	**PDS**	**Baseline**	**DoM**	**PDS**
CoT	71%	75%	**78%**	49%	50%	**53%**
Neutral	68%	74%	**78%**	46%	50%	**53%**
Anti-CoT	11%	18%	**21%**	29%	30%	**31%**

Table 1: Performance comparison (Accuracy %) across prompting conditions: Baseline (no steering), Difference of Means (DoM), and Prototype-Based Dynamic Steering (PDS) on GSM8k and AQuA-Rat benchmarks.

Geometric Analysis of the Prototype Space Our empirical analysis reveals remarkable geometric properties of the learned prototypes. Computing pairwise angles between prototypes yields a narrow geometric composition where angular separations

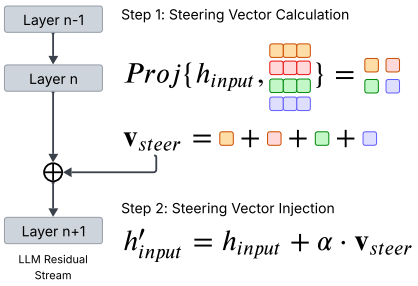

Figure 3: **Overview of steering vector injection in Prototype-Based Dynamic Steering**. During inference, projections of input activations onto prototypes are summed to compute the steering vector, which is then injected into the model's residual stream to enhance reasoning behavior.

are consistently within the narrow range of 8-22 degrees across different datasets and experimental conditions. This tight angular clustering indicates that all prototypes lie approximately on the surface of a high-dimensional cone in activation space.

The Cone's Central Axis: The central axis represents the core cognitive transformation from "direct answering" to "step-by-step reasoning". Mathematically, this axis corresponds to $\sum_{j=1}^{k} \mu_j = k \cdot \text{mean}(D)$, equivalent to the traditional difference-of-means vector scaled by the number of clusters. This reveals that our prototype-based approach contains difference-of-means as a special case while extending significantly beyond it.

The Cone's Surface: Each prototype captures a strategic variation around the core reasoning transformation. These variations correspond to different problem-solving styles, domain-specific approaches (arithmetic vs. algebraic vs. geometric), and different levels of mathematical abstraction. PDS can capture these variances, unlike difference-of-means which discards it by averaging. PDS can also amplify the representation of input-specific information, since the dynamic projection selects optimal strategy combinations based on problem activation patterns.

4 Evaluation

Our evaluation spans three common reasoning benchmarks. For each dataset, we (1) collect CoT-neutral activation deltas on the training split, (2) cluster them into reasoning prototypes, and (3) evaluate under Neutral, CoT, and Anti-CoT prompt conditions using Accuracy@1.

- **GSM8K:** Grade-school math word problems requiring multi-step arithmetic reasoning. (Cobbe et al., 2021b)
- **AQuA-RAT:** Multiple-choice algebraic word problems that demand symbolic manipulation and higher-order reasoning. (Ling et al., 2017)
- **BIG-Bench Subset:** We selected five tasks that exercise distinct skills: checkmate-in-one, physics-questions, rhyming, temporal sequences, intersect geometry. (Srivastava et al., 2022)

We conduct our experiments using the LLaMA-3-Instruct-8B (Touvron et al., 2023) language model. To account for the increased difficulty of the selected BIG-Bench subset, where tasks such as Checkmate-in-One and Physics Questions present significant challenges for smaller models, we additionally evaluate performance using the larger LLaMA-3-Instruct-70B model. This comparison allows us to assess whether our prototype-based steering method scales effectively with model capacity and to verify that observed improvements are not limited by the expressiveness of the 8B model. Evaluating on both 8B and 70B variants enables a more comprehensive understanding of steering efficacy across model sizes.

Each dataset in our evaluation suite demands diverse and heterogeneous reasoning strategies, even within seemingly domains. While GSM8K and AQuA-RAT are both math-focused, the problems span a variety of reasoning types. For example, GSM8K questions require combining numerical estimation, unit conversion, logical sequencing, or multi-step arithmetic decomposition (e.g., "If a pencil costs $1.25 and a notebook costs twice as much, how much do five notebooks and three pencils cost in total?"). Similarly, AQuA-RAT tasks involve symbolic manipulation, pattern recognition in algebraic expressions, distractor elimination in multiple-choice formats, and even linguistic disambiguation in problem phrasing (e.g., "What is the value of x if three times the difference between x and 4 is equal to 18?"). These subtasks demand distinct cognitive strategies within a single domain. The BIG-Bench subset expands this diversity further by explicitly testing spatial, temporal, linguistic, and logic-based reasoning, such as determining move sequences in chess (Checkmate-in-One), resolving event orders (Temporal Sequences), or rhyming constraints in generative tasks.

- **Neutral:** Provides no reasoning cues. **Question:** "[problem statement]". **Answer:**
- **Chain-of-Thought (CoT):** Encourages the

Dataset	Chain-of-Thought			Neutral			Anti-Chain-of-Thought		
	No-Steering	DoM	PDS	No-Steering	DoM	PDS	No-Steering	DoM	PDS
Checkmate-in-One	0.2%	**0.6%**	**0.6%**	**0.2%**	**0.2%**	0.0%	**0.2%**	0.0%	**0.2%**
Physics Questions	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Rhyming	23.7%	**25.8%**	24.7%	23.7%	**25.8%**	23.7%	24.2%	**24.7%**	**24.7%**
Temporal Sequences	59.5%	50.5%	**60.0%**	57.0%	50.5%	**58.5%**	**60.0%**	50.5%	59.5%
Intersect Geometry	34.6%	**38.8%**	35.4%	34.3%	36.6%	**41.8%**	31.2%	**36.2%**	32.2%

Table 2: Performance comparison (Accuracy %) across prompting conditions: No Steering, Difference of Means (DoM), and Prototype-Based Dynamic Steering (PDS) on BIG-Bench benchmark with Llama-3-Instruct-8B model.

Dataset	Chain-of-Thought			Neutral			Anti-Chain-of-Thought		
	No-Steering	DoM	PDS	No-Steering	DoM	PDS	No-Steering	DoM	PDS
Checkmate-in-One	9.8%	6.8%	**10.4%**	7.0%	7.8%	**7.8%**	14.8%	10.8%	**15.8%**
Physics Questions	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	**6.2%**	**6.2%**
Rhyming	23.7%	14.2%	**25.8%**	**22.6%**	15.8%	**22.6%**	**26.8%**	25.3%	25.8%
Temporal Sequences	**98.5%**	97.0%	**98.5%**	**98.5%**	97.5%	**98.5%**	**98.5%**	97.5%	**98.5%**
Intersect Geometry	**38.4%**	37.4%	**38.4%**	37.6%	**40.0%**	37.2%	**41.4%**	40.2%	40.6%

Table 3: Performance comparison (Accuracy %) across prompting conditions: No Steering, Difference of Means (DoM), and Prototype-Based Dynamic Steering (PDS) on BIG-Bench benchmark with Llama-3-Instruct-70B model.

model to reason step by step. **Question:** "[problem statement]". Let's think step by step. **Answer:**

- **Anti Chain-of-Thought (Anti-CoT):** Explicitly instructs the model to avoid reasoning. **Question:** "[problem statement]". Answer immediately, without elaboration. **Answer:**

Our prototype-based method is applied separately to each dataset; steering vectors are extracted from that dataset's training set, clustered into reasoning prototypes, and used for evaluation solely on that dataset. This design avoids any unintended cross-task leakage and ensures that prototype representations are aligned with the unique reasoning requirements of each benchmark. We evaluate three different experiment cases:

- **No Steering:** The model receives no intervention at the activation level.
- **Difference of Means (DoM):** A global steering direction is computed as the mean activation difference between CoT and Neutral responses, then applied to each new sample for steering.
- **Prototype-Based Dynamic Steering (PDS):** Activation differences are clustered into semantically meaningful prototypes, and the most relevant one is dynamically selected at inference time based on projection alignment with the model's hidden state.

This evaluation structure allows us to test whether fine-grained, interpretable, and dynamically applied steering vectors can enhance model performance across varied reasoning demands, both within and across domains.

5 Results & Discussion

Results Across Datasets Tables 1,2 and 3 present our experimental results alongside the difference-of-means (DOM) technique in all three benchmarks. Our prototype-based dynamic steering method demonstrates consistent improvements across all datasets and prompting conditions against the baseline. The results demonstrate remarkable consistent improvements across all three benchmarks. Our dynamic steering method achieves improvements ranging from 0.6% to 9.66% across different conditions and benchmarks. The most significant gains consistently occur in the Neutral and Anti-CoT conditions, where the model lacks explicit reasoning guidance. Notably, even when CoT prompting is already present, our method does not cause performance degradation, suggesting that activation-level steering modify reasoning patterns that prompting already has elicited.

Comparison with Difference of Means Our method outperforms Difference of Means technique across various conditions and datasets in terms of performance. While both methods achieve improvements over the unsteered baseline, the superior performance of our approach validates the importance of capturing behavioral diversity rather than averaging it away. To investigate whether our improvements stem from directional guidance or

Prompt Type	Method	Avg. Tokens	Std. Dev.
Neutral	No Steering	287.31	492.82
	DoM	256.96	358.54
	PDS	**254.58**	**357.60**
CoT	No Steering	275.61	358.16
	DoM	303.09	440.70
	PDS	**258.03**	**146.45**

Table 4: **Average number of output tokens on the AQuA-RAT dataset across prompting types and steering methods**. CoT prompts produce longer, Anti-CoT prompts yield concise completions. PDS improves accuracy in all settings with minimal increase in length.

Dataset	No Steering	PDS	Δ
GSM8k	11.60%	21.12%	+9.52%
Aqua-Rat	29.50%	31.50%	+2.00%
Checkmate-in-One	14.8%	15.8%	+1%
Physics Questions	0%	6.2%	+6.2%
Rhyming	26.8%	25.8%	-1%
Temporal Sequences	98.5%	95.8%	+0%
Intersect Geometry	41.4%	40.6%	-0.8%

Table 5: Performance on Anti-CoT prompting

simple activation scaling, we conduct an experiment using normalized prototypes. Our original prototypes had an average norm of 4.51, representing significant magnitude in the activation space. When the prototypes are normalized to unit length, performance remains unchanged across all conditions and datasets. This result provides evidence that our projection-based steering method operates through directional guidance rather than magnitude scaling.

Our results demonstrate that PDS achieves consistent improvements up to 9.72% across three diverse benchmarks. Consistent improvements across GSM8k, AQuA-Rat, and BIG-Bench suggests that PDS enhances the capturing of various patterns in one steering vector. The largest gains consistently occur in Neutral and Anti-CoT settings, indicating PDS is most effective when explicit reasoning guidance is absent, sufficiently compensating for weak reasoning tendencies.

Effect on Completion Length We analyzed the average number of tokens generated on the AQuA-RAT dataset. While all three methods receive the same input prompt, the steering interventions influence not only accuracy but also the length and structure of the completions.

PDS completions are slightly longer on average, suggesting more structured or elaborated reasoning chains. Notably, the increased length is accompanied by a substantial gain in accuracy, indicating that the additional tokens reflect meaningful intermediate reasoning rather than verbosity.

Reasoning Enhancement without CoT The most impactful finding of our study is the model's behavior under Anti-Chain-of-Thought (Anti-CoT) prompts, which explicitly instruct the model to "answer immediately without elaboration." Conventional activation-steering methods often override such instructions, altering output structure. In contrast, PDS preserves the requested response format while still improving problem-solving capability.

Across benchmarks, PDS consistently boosts accuracy in Anti-CoT settings without inducing step-by-step reasoning. As shown in Table 1, GSM8K accuracy rises from 11% (no steering) to 21% (+10 %), and AQuA-RAT improves from 29% to 31% (+2 %). Similarly, in the BIG-Bench tasks (Table 5), PDS achieves +6.2 pp on Physics Questions and a modest +1% on Checkmate-in-One, while preserving the terse output style mandated by Anti-CoT instructions. This outcome is a core finding of our work and the main proof of PDS's contribution to enhancing latent reasoning capabilities. Unlike prior steering approaches that primarily seek to change model behavior, PDS demonstrates that substantial reasoning gains can be achieved without altering instruction-level outputs. By selectively strengthening reasoning while preserving compliance with instructions, PDS establishes a robust intervention paradigm for solving problems that require diverse reasoning approaches.

6 Conclusion

Prototype-Based Dynamic Steering (PDS) is an inference-time method that enhances reasoning in large language models without altering weights or prompts. By clustering activation differences between CoT and neutral prompts, PDS uncovers interpretable prototypes representing diverse reasoning strategies. Input activations are projected onto these prototypes to generate context-sensitive steering vectors, outperforming the standard difference-of-means approach across benchmarks and prompting styles. Crucially, PDS boosts performance even when explicit step-by-step instructions are omitted, revealing latent reasoning structure in activation space. Future work may extend prototype discovery techniques and apply dynamic steering cross-modally.

7 Limitations

While Prototype-Based Dynamic Steering (PDS) demonstrates strong reasoning improvements and introduces a novel activation-space perspective, it has several limitations. First, our current prototype discovery process relies on a predefined dataset of reasoning traces represented as activation differences and may not generalize optimally to domains with different reasoning styles. Second, the effectiveness is not uniform across all tasks according to the empirical outcomes. In particular, improvements tend to be more modest in settings where the model's baseline reasoning capability is already near saturation, leaving less latent structure to exploit. Third, tasks that rely more heavily on domain-specific knowledge rather than on compositional or step-by-step reasoning offer fewer opportunities for PDS to induce substantial gains. These observations suggest that the strength of PDS is closely tied to the presence of underutilized reasoning capacity in the base model, which is an insight that points to future directions in designing steering strategies tailored to different reasoning regimes. Finally, as with other activation-level interventions, steering techniques can be misused to create jailbreak cases or amplify undesired behaviors. While our work focuses on reasoning enhancement, the fact that PDS provides an alternative mechanism for constructing steering vectors highlights the importance of developing complementary safety measures to mitigate potential jailbreak risks.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *arXiv preprint*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Ilya Sutskever, and Dario Amodei et al. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Runjin Chen, Zhenyu Zhang, Junyuan Hong, Souvik Kundu, and Zhangyang Wang. 2025. Seal: Steerable reasoning calibration of large language models for free. *arXiv preprint*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman et al. 2021a. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021b. Training verifiers to solve math word problems. *arXiv: Machine Learning*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *Preprint*, arXiv:2309.08600.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. 2024. Inspecting and editing knowledge representations in language models. *Preprint*, arXiv:2304.00740.

Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik Opitz, and Tobias Hecking. 2024. Style vectors for steering generative large language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 782–802, St. Julian's, Malta. Association for Computational Linguistics.

Tom A. Lamb, Adam Davies, Alasdair Paren, Philip H. S. Torr, and Francesco Pinto. 2025. Focus on this, not that! steering llms with adaptive feature specification. *Preprint*, arXiv:2410.22944.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. 2025. Programming refusal with conditional activation steering. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*. ArXiv:2504.17130.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems 36*. ArXiv:2306.03341.

Mingqi Li, Karan Aggarwal, Yong Xie, Aitzaz Ahmad, and Stephen Lau. 2024. Learning from contrastive prompts: Automated optimization and adaptation. *Preprint*, arXiv:2409.15199.

Zihao Li, Xu Wang, Yuzhe Yang, Ziyu Yao, Haoyi Xiong, and Mengnan Du. 2025. Feature extraction and steering for enhanced chain-of-thought reasoning in language models. *arXiv preprint*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *ACL*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2024. In-context vectors: Making in context learning more effective and controllable through latent space steering. *Preprint*, arXiv:2311.06668.

Stuart P. Lloyd. 1982. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

J. B. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *Preprint*, arXiv:2402.04249.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, and Pamela Mishkin et al. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. 2025. The geometry of categorical and hierarchical concepts in large language models. *Preprint*, arXiv:2406.01506.

Van-Cuong Pham and Thien Huu Nguyen. 2024. Householder pseudo-rotation: A novel approach to activation editing in LLMs with direction-magnitude perspective. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13737–13751, Miami, Florida, USA. Association for Computational Linguistics.

Joris Postmus and Steven Abreu. 2025. Steering large language models using conceptors: Improving addition-based activation engineering. *Preprint*, arXiv:2410.16314.

Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo M. Ponti, and Shay B. Cohen. 2024. Spectral editing of activations for large language model alignment. *Preprint*, arXiv:2405.09719.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of ACL 2024*, pages 15504–15522.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. 2025. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. *Preprint*, arXiv:2503.05613.

Oscar Skean, Md Rifat Arefin, Yann LeCun, and Ravid Shwartz-Ziv. 2024. Does representation matter? exploring intermediate layers in large language models. *Preprint*, arXiv:2412.09563.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. 2022. Extracting latent steering vectors from pretrained language models. *Preprint*, arXiv:2205.05124.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. *Preprint*, arXiv:2310.15154.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David S. Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint*.

Hieu M. Vu and Tan Minh Nguyen. 2025. Angular steering: Behavior control via rotation in activation space. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*.

Tianlong Wang, Xianfeng Jiao, Yinghao Zhu, Zhongzhi Chen, Yifan He, Xu Chu, Junyi Gao, Yasha Wang, and Liantao Ma. 2025. Adaptive activation steering: A tuning-free llm truthfulness improvement method for diverse hallucinations categories. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 2562–2578. ACM.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Sean Xie, Soroush Vosoughi, and Saeed Hassanpour. 2023. Proto-lm: A prototypical network-based framework for built-in interpretability in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3964–3979, Singapore. Association for Computational Linguistics.

Shu Yang, Shenzhe Zhu, Liang Liu, Lijie Hu, Mengdi Li, and Di Wang. 2025. Exploring the personality traits of llms through latent features steering. *Preprint*, arXiv:2410.10863.

Liang Yao. 2025. Large language models are contrastive reasoners. *Preprint*, arXiv:2403.08211.

Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2025. Attention heads of large language models. *Patterns*, 6(2). Published February 14, 2025.

Andy Zou and colleagues. 2025. Investigating generalization of one-shot llm steering vectors. *arXiv preprint*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, and Xuwang Yin et al. 2025. Representation engineering: A top-down approach to ai transparency. *Preprint*, arXiv:2310.01405.

A Appendix

Dataset Details

We evaluate our method across three complementary reasoning benchmarks, selected for their diversity in problem format, reasoning complexity, and domain coverage. For each dataset, we use a subset of the training split to extract steering prototypes and evaluate performance on the official test set.

- **GSM8K:** GSM8K consists of grade-school math word problems requiring multi-step arithmetic reasoning. We use all of the 7470 samples from the training set to compute steering vectors and evaluate on the full test set of 1,319 examples.
- **AQuA-RAT:** AQuA-RAT is a dataset of multiple-choice algebraic word problems that often involve symbolic manipulation and intermediate computation steps. We sample 10,000 questions from the training set to derive steering representations and test on a 254-example evaluation set.
- **BIG-Bench Subset** From the BIG-Bench benchmark, we select five tasks that reflect distinct reasoning types: *Checkmate-in-One*, *Physics Questions*, *Rhyming*, *Temporal Sequences*, and *Intersect Geometry*. Table REF presents the number of samples taken from each set (training, test) for each task.

Dataset	Training Set	Test Set
Checkmate-in-One	1000	699
Physics Questions	38	16
Rhyming	763	190
Temporal Sequences	800	200
Intersect Geometry	1000	1000

Table 6: Number of samples used for prototype generation and evaluation for BIG-Bench dataset.

Prototype Cluster Descriptions

To interpret the latent structure captured by Prototype-Based Dynamic Steering (PDS), we collect 50 representative prompts from each cluster and query multiple large language models (GPT-4.5, Claude Sonnet 4, Gemini 2.5 Pro) to assign semantic labels. Each model independently analyzed 50 randomly sampled questions from each prototype's cluster to assign a cluster name and explain each cluster without any context and information about our methodology. Cluster-level names for each model and dataset are provided in Tables 7, 8 and 9.

Philosophy of Prototype-Based Steering Our approach embodies a fundamentally different philosophical stance toward behavioral steering compared to traditional methods.

Traditional steering methods, particularly Difference of Means, operate under the assumption that complex behaviors can be captured by single, universal directions in the activation space. This "behavioral uniformity" hypothesis treats variations in reasoning approaches as noise to be averaged away. Our prototype-based philosophy challenges this assumption by recognizing that **complex cognitive behaviors** such as mathematical reasoning are formed by unique combinations of multiple distinct "reasoning strategies". Rather than viewing this diversity as problematic variation, we treat it as valuable information that should be preserved and leveraged.

Our philosophy extends beyond static intervention to embrace **input-responsive steering**. The projection mechanism dynamically selects the optimal combination of reasoning strategies based on the current problem's activation pattern. This represents a shift from "one-size-fits-all" steering to "problem-tailored" enhancement. This approach

reflects a deeper understanding of how cognitive behaviors manifest in neural networks: rather than being represented by single directions, complex behaviors occupy structured vector subspaces that require nuanced intervention strategies.

Inference and Steering Hyperparameters

All experiments use standard autoregressive decoding with temperature 0.1 and top-p sampling at 0.9 during prototype collection. Evaluation is performed using greedy decoding with a maximum length of 512 tokens. We fix the intervention layer to 16 for all steering experiments, based on preliminary probing of reasoning signal strength across layers.

The steering coefficient α is held constant at 1.0 across datasets and prompt types. This consistent configuration allows us to isolate the effect of steering vector construction (PDS vs. DoM) without confounding from hyperparameter tuning.

Computational Setup

All experiments were conducted on a high-performance computing node equipped with 8 NVIDIA H100 GPUs (each with 80GB VRAM). We used PyTorch 2.1 and Hugging Face's Transformers library for model loading and inference. Steering interventions were implemented through forward hooks on hidden states during inference. Each evaluation run was parallelized across GPUs using batch-level distribution. Prototype clustering and projection computations were performed on the same hardware, with minimal additional overhead relative to inference. No fine-tuning or model weight modifications were performed. All results reflect inference-time interventions on the pretrained LLaMA-3-Instruct-8B model.

Prompt-Length and Token Statistics

To evaluate the economic efficiency of different steering strategies, we collect output token statistics across datasets and prompting conditions. For each combination of dataset, prompt type, and method (No Steering, DoM, PDS), we report the average number of tokens generated and the standard deviation. Across all datasets, Prototype-Based Dynamic Steering (PDS) consistently produces shorter or comparable outputs to DoM, especially under CoT and Neutral prompting, while maintaining or improving accuracy. Under Anti-CoT prompts, all methods yield similarly minimal output, as expected by design. Detailed statistics are presented in Table 12.

Licenses and Usage

All models and datasets used in this study are publicly available and used in accordance with their respective licenses. Experiments involving LLaMA-3 models were conducted under their original open-source licenses (Llama 3 Community License Agreement). Benchmark datasets including GSM8K, AQuA-RAT, and BIG-Bench are licensed for research use (MIT or and Apache 2.0, respectively), and we adhere to all associated terms.

Cluster	GPT-4.5	Gemini 2.5 Pro	Claude Sonnet-4
0	Multi-step Arithmetic Problems	Multi-step Arithmetic Problems	Multi-Step Calculations
1	Comparative Quantity Problems	Relational & Dependent Variables	Complex Multi-Variable Problems
2	Basic Quantitative Reasoning	Straightforward Two-Step Calculations	Basic Arithmetic Applications
3	Complex Multi-condition Problems	Complex Proportional Reasoning	Business and Real-World Scenarios
4	Financial Arithmetic Problems	Rate, Cost, and Profit	Cost Analysis
5	Ratio and Comparative Age Problems	Logical Deduction Puzzles	Fraction and Percentage Problems
6	Arithmetic with Units & Conversions	Comparative Relational Arithmetic	Simple Relational Problems
7	Sequential Operations & Patterns	Complex Narrative Problems	Complex Logical Reasoning
8	Percentage and Fraction Problems	Direct Rate and Unit Problems	Direct Calculation Problems

Table 7: Cluster interpretability validation for GSM8k dataset via blind evaluation.

Cluster	GPT-4.5	Gemini 2.5 Pro	Claude Sonnet-4
0	Rhyming Choices	Phonetic Sound Recognition	Rhyming Word Recognition
1	Shape Intersections	Intersection Analysis	Geometric Intersection Counting
2	Chess Mate-One	Chess Checkmate Puzzles	Chess Mate-in-One Puzzles
3	Time Interval Deduction	Logical Word Problems	Schedule Logic Reasoning

Table 8: Cluster interpretability validation for BIG-Bench dataset via blind evaluation.

Cluster	GPT-4.5	Gemini 2.5 Pro	Claude Sonnet-4
0	Diverse Multi-Step Word Problems	Advanced Word Problems	Complex Multi-Step Problems
1	Basic Numerical Computation	Number Properties and Operations	Basic Mathematical Operations
2	Financial & Work-Rate Scenarios	Financial and Proportional Reasoning	Business and Investment Problems
3	Fundamental Arithmetic Concepts	Foundational Math Concepts	Elementary Calculations
4	Geometry & Probability Applications	Complex Quantitative Scenarios	Applied Mathematical Reasoning
5	Intensive Numeric Challenges	Basic Arithmetic and Number Sense	Computational Mathematic
6	Complex Algebraic & Rate Problems	Rate and Change Problems	Intermediate Problem Solving
7	SApplied Real-World Quant Problems	Applied Business and Finance	Advanced Word Problems

Table 9: Cluster interpretability validation for AQuA-RAT dataset via blind evaluation.

Parameter	Value
Model	LLaMA-3-Instruct-8B
Temperature	0.7
Top-p (nucleus sampling)	0.9
Max tokens	4096
Sampling strategy	Temperature Sampling

Table 10: Inference hyper-parameters used across all evaluations.

Dataset	Prompt Type	Intervention Layer	Steering Strength (α)
GSM8K	Neutral	16	1
	CoT	16	1
	Anti-CoT	16	10
AQuA-RAT	Neutral	16	7
	CoT	16	1
	Anti-CoT	16	10
BIG-Bench	Neutral	15	1
	CoT	15	1
	Anti-CoT	15	1

Table 11: Steering intervention hyperparameters used for each dataset and prompt condition.

Dataset	Prompt Type	No Steering		DoM		PDS	
		Avg.	Std.	Avg.	Std.	Avg.	Std.
AQuA-RAT	Neutral	287.31	492.82	256.96	358.54	**254.58**	**357.60**
	CoT	275.61	358.16	303.09	440.70	**258.03**	**146.45**
	Anti-CoT	10.05	1.70	10.05	1.69	10.05	1.70
GSM8k	Neutral	155.12	61.56	158.74	55.69	**126.32**	**132.32**
	CoT	182	121.86	180.27	58.92	**90.29**	**100.51**
	Anti-CoT	7.03	3.54	6.93	2.91	**6.39**	**4.25**

Table 12: Token length statistics (average and standard deviation) across datasets and prompt types for each steering method.