

BRIDGING THE KNOWLEDGE-PREDICTION GAP IN LLMs ON MULTIPLE-CHOICE QUESTIONS

Yoonah Park*, **Haesung Pyun***, **Yohan Jo†**

Graduate School of Data Science, Seoul National University

{wisdomsword21, haesung.pyun, yohan.jo}@snu.ac.kr

ABSTRACT

Large Language Models (LLMs) often fail on multiple-choice questions (MCQs) despite demonstrating correct knowledge in other contexts, such as free-form generation. To investigate the mechanism underlying this knowledge-prediction gap on MCQs and alleviate it, we conduct a probing analysis and find that residual streams in certain layers contain a subspace spanned by two important bases: a *knowledge basis* that encodes the probability of the ground-truth answer for a given MCQ and a *prediction basis* that encodes the probability of the answer choice predicted by the model. We observe that incorrect predictions arise from a misalignment of the model’s hidden states along these two bases. Hence, we introduce **KAPPA** (Knowledge-Aligned Prediction through Projection-based Adjustment), a parameter-free intervention that transforms the hidden states to align the prediction coordinate with the knowledge coordinate within this subspace. Experiments on binary-choice reformulations of Big-Bench-Hard and ARC-Challenge show that KAPPA substantially improves accuracy and consistently outperforms baselines. While optimal subspaces differ across tasks, subspaces generalize to some extent, as supported by cross-dataset experiments. Moreover, KAPPA extends its effectiveness to free-form questions beyond MCQs. Our work provides a new geometric understanding of the knowledge-prediction gap and offers a practical method for better aligning model behavior with its latent knowledge.¹

1 INTRODUCTION

Multiple-choice questions (MCQs) are a widely used paradigm for evaluating large language models (LLMs) on knowledge- and reasoning-intensive tasks, as they provide clear ground truth and straightforward scoring. However, LLMs often fail to reliably predict correct answers in MCQ settings despite having the required knowledge. For instance, a model may generate the correct answer in a free-form setting but select the wrong option for the same question in the MCQ format (Figure 1a). Recent studies on factuality tasks show that LLMs encode correct answers in their internal representations, but fail to consistently connect them to their output predictions (Marks & Tegmark, 2024; Liu et al., 2023). Even when LLMs select an incorrect answer, the hidden representations frequently encode the correct answer, as detected by simple linear probes (Liu et al., 2023; Marks & Tegmark, 2024; Bao et al., 2025). This knowledge-prediction gap is problematic because the model’s knowledge is not fully exploited in its predictions, and the model exhibits unfaithful behavior where its outputs do not reflect its internal knowledge but are instead driven by superficial cues. While prior work has detected this mismatch, its underlying mechanism remains underexplored. As the first work, we focus on binary-choice questions and examine the hidden states of LLMs to better understand the knowledge-prediction gap and explore ways to mitigate it.

We find that in the residual streams of certain layers, there exists a subspace spanned by two functionally distinct bases: a *knowledge basis* that reliably encodes the probability of the correct answer choice for a given MCQ, and a *prediction basis* that encodes the probability of the answer choice

*Equal contribution

†Corresponding Author

¹We will release our code publicly upon publication.

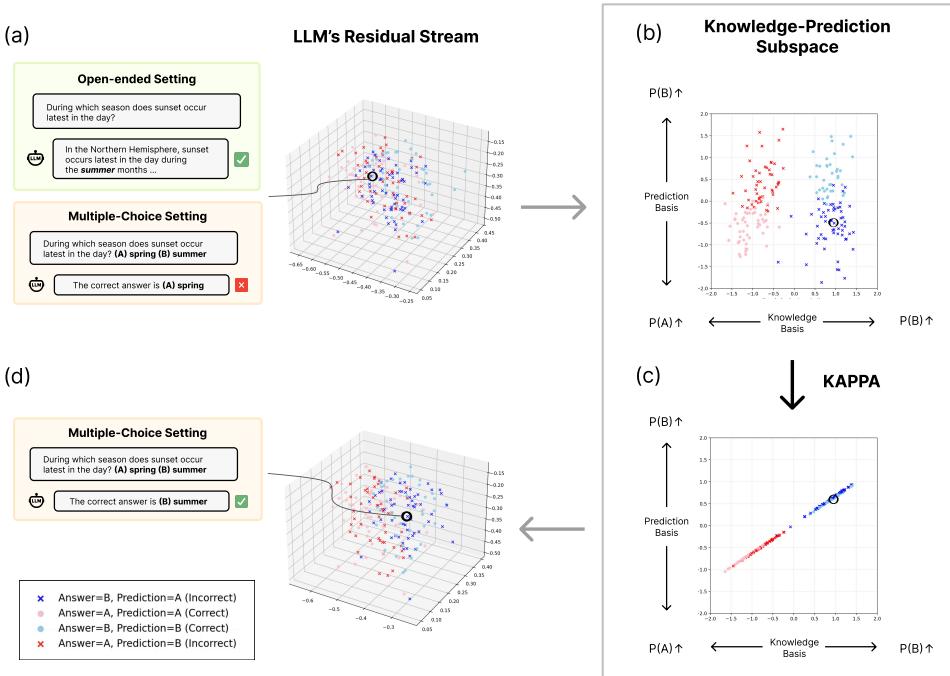


Figure 1: KAPPA resolves the knowledge-prediction gap through geometric realignment. (a) Motivating example: an LLM answers correctly in free-form but fails in MCQ format. 3D visualization shows hidden representations. (b) In the 2D knowledge-prediction subspace, the circled point shows misalignment between knowledge (x-axis) and prediction (y-axis) coordinates. (c) KAPPA geometrically transforms representations to align coordinates. (d) This correction enables faithful expression of internal knowledge, producing the correct answer.

eventually predicted by the model. To construct this subspace, we train two linear probes with disjoint objectives on the same hidden states: one predicts the ground-truth answer, and the other predicts the model’s output (note that questions are binary) (Alain & Bengio, 2017). We then use the trained probe weights as the two bases for the subspace. The coordinates in this subspace correspond to the probabilities of the correct answer and the predicted answer, respectively (Figure 1b). We observe that hidden state activations that lead to correct predictions often align well along both bases, whereas misalignments often result in incorrect predictions.

To drive the model to choose answer choices faithful to its knowledge, we introduce KAPPA (**K**nowledge-**A**ligned **P**rediction through **P**rojection-based **A**djustment), a test-time intervention that adjusts hidden states by calibrating values along the two bases within the knowledge-prediction subspace, adjusting the prediction coordinate to match the knowledge coordinate (Figure 1c). Unlike typical activation steering methods that apply a fixed vector uniformly across all hidden states, KAPPA steers hidden states adaptively on the fly.

Our experiments apply KAPPA to binary-choice tasks and free-form generation, revealing three key insights. First, many failures in knowledge- and reasoning-intensive tasks stem from misalignment between knowledge and prediction rather than inherent lack of knowledge. On the binary reformulation of the challenging BBH dataset (BBH-Binary), KAPPA boosts Llama-2-7B-Chat (Touvron et al., 2023) accuracy from near-random (50.9%) to 70.8%, nearly 20% gain. Second, the knowledge-prediction gap exhibits a geometric structure that generalizes across datasets: probes trained on one dataset (e.g., MMLU-Binary) transfer to others (e.g., BBH-Binary) with gains of up to 4 points, indicating that the subspace reflects fundamental model properties to some extent rather than task-specific artifacts. Third, alignment benefits extend beyond MCQs: KAPPA improves free-form generation accuracy by 3.8%, demonstrating that geometric alignment enhances a model’s ability to faithfully express its internal knowledge.

Contributions. Our work provides a representation-level analysis of the knowledge-prediction gap in LLMs and introduces a test-time intervention as a solution. Our contributions are threefold:

- We present a method for identifying a subspace of LLM hidden states that encodes the probabilities of correct knowledge and model prediction for MCQs, and interpret the knowledge-prediction gap as a misalignment along these bases.
- We introduce a closed-form intervention method that dynamically calibrates activations along the two bases at inference time.
- We show that the subspaces generalize to some extent across datasets and extend beyond MCQs. KAPPA improves free-form generation on knowledge- and reasoning-intensive tasks, making models act more faithfully to their internal knowledge.

2 METHOD

We present KAPPA, a test-time intervention that mitigates the knowledge-prediction gap by applying transformations to residual streams. Our pipeline proceeds in three steps: we collect residual stream activations from binary-choice questions (Figure 2a), then identify the knowledge-prediction subspace via linear probes (Figure 2b), and finally apply our intervention within this subspace (Figure 2c). KAPPA operates entirely within the subspace spanned by the knowledge and prediction bases, offering an interpretable framework for analyzing and manipulating the model’s knowledge-prediction relationship.

2.1 IDENTIFYING KNOWLEDGE AND PREDICTION BASES

Our intervention operates within a subspace of the residual stream that captures the relationship between what the model knows and what it predicts. For each layer l , we define a subspace spanned by the knowledge basis $u_{\text{knowledge}}^{(l)} \in \mathbb{R}^d$, which encodes the probability of the correct answer choice, and the prediction basis $u_{\text{prediction}}^{(l)} \in \mathbb{R}^d$, which encodes the probability of the model’s own choice. We obtain these bases by training linear probes on hidden states.

Dataset Construction. To learn the basis vectors, we construct training datasets for linear probes. For each input prompt $x_i \in D_{\text{train}}$ containing a binary-choice question, we store both the ground-truth label $y_i \in \{0, 1\}$ and the model’s prediction $\tilde{y}_i \in \{0, 1\}$. At every layer l , we extract the residual stream activation of the last token of x_i , denoted as $h^{(l)}(x_i) \in \mathbb{R}^d$, and construct two datasets. The *knowledge* dataset, $D_{\text{knowledge}}^{(l)} = \{(h^{(l)}(x_i), y_i)\}$, pairs activations with ground-truth labels, while the *prediction* dataset, $D_{\text{prediction}}^{(l)} = \{(h^{(l)}(x_i), \tilde{y}_i)\}$, pairs the same activations with the model’s predicted choices.

Linear Probing. We train separate logistic regression classifiers on each dataset for each layer l . Each probe has parameters consisting of a weight vector $u^{(l)} \in \mathbb{R}^d$ and a bias term $b^{(l)}$. The learned weight vectors provide the two bases for the knowledge-prediction subspace: $u_{\text{knowledge}}^{(l)}$ and $u_{\text{prediction}}^{(l)}$.

With these bases, any hidden state $h^{(l)}(x_i)$ can be projected into the two-dimensional subspace by computing its coordinates along the two directions:

$$\text{Knowledge coordinate: } u_{\text{knowledge}}^{(l)\top} h^{(l)}(x_i), \quad \text{Prediction coordinate: } u_{\text{prediction}}^{(l)\top} h^{(l)}(x_i).$$

These quantities correspond to the logits produced by the linear probes, and thus represent how strongly the activation supports an option as the correct answer (knowledge coordinate) versus how strongly it favors an option as the model’s predicted output (prediction coordinate). As shown in Figure 2b, this construction provides an interpretable two-dimensional geometry that captures the relationship between the model’s internal knowledge and its external predictions.

2.2 KNOWLEDGE-ALIGNED PREDICTION ADJUSTMENT

At inference time, we calibrate the two coordinates of the hidden state to drive the model to make a prediction faithful to its knowledge. Given a question $x_i \in D_{\text{test}}$, we intervene at a chosen layer l

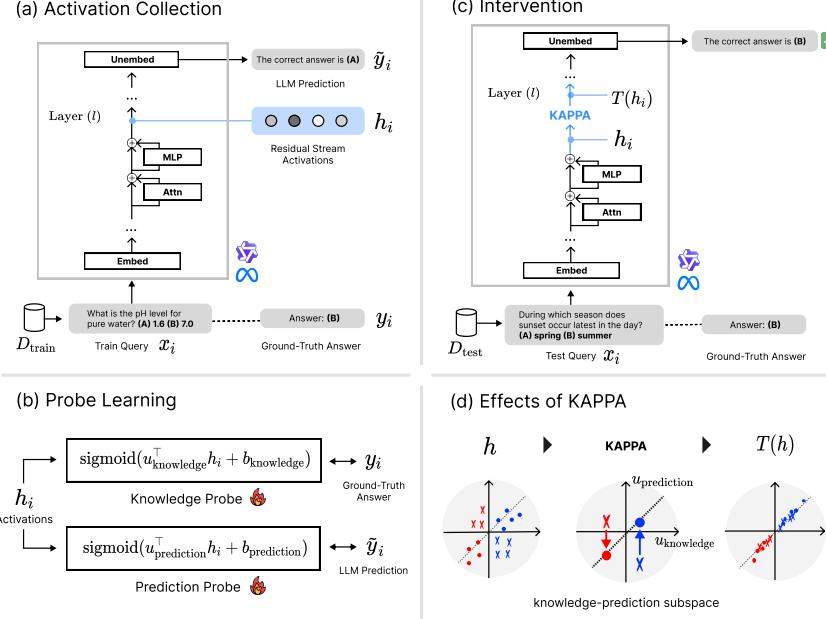


Figure 2: KAPPA pipeline overview. (1) Collect residual stream activations from transformer layers during inference on binary-choice questions. (2) Train knowledge and prediction probes: logistic regression classifiers that predict ground-truth labels and model outputs, respectively, from the same hidden states. (3) At inference, KAPPA applies a geometric transformation to the hidden representation. (4) 2D visualization showing the distribution of representations before (left) and after (right) the intervention in the knowledge-prediction subspace.

by applying a transformation to the residual stream activation $h = h^{(l)}(x_i)$. For clarity, we omit the layer index (l) in the following notation.

Transformation Objective. Our goal is to derive a transformation that aligns the prediction coordinate with the corresponding knowledge coordinate of each hidden state. For a hidden state h , our intervention computes a transformation $\mathcal{T} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ that maps h to a modified state $h' = \mathcal{T}(h)$ under the following design: (1) *Alignment constraint*: the modified prediction coordinate must align with the knowledge coordinate; (2) *Minimal Perturbation*: the perturbation $\|h' - h\|$ is minimized so as to preserve other encoded information.

Constrained Optimization. We formulate the update as the following constrained optimization problem:

$$\min_{h'} \|h' - h\|_2^2 \quad \text{s.t.} \quad \tilde{u}_{\text{prediction}}^\top \tilde{h}' = \tilde{u}_{\text{knowledge}}^\top \tilde{h}, \quad (1)$$

where we use augmented notations

$$\tilde{h}' = \begin{bmatrix} h' \\ 1 \end{bmatrix}, \quad \tilde{h} = \begin{bmatrix} h \\ 1 \end{bmatrix}, \quad \tilde{u}_{\text{knowledge}} = \begin{bmatrix} u_{\text{knowledge}} \\ b_{\text{knowledge}} \end{bmatrix}, \quad \tilde{u}_{\text{prediction}} = \begin{bmatrix} u_{\text{prediction}} \\ b_{\text{prediction}} \end{bmatrix} \in \mathbb{R}^{d+1}.$$

Solving this problem yields the minimal ℓ_2 modification to h that satisfies the alignment constraint.

We derive a closed-form solution, which takes the form of an affine transformation of h (derivation provided in Appendix B):

$$h' = h + \frac{\tilde{u}_{\text{knowledge}}^\top \tilde{h} - \tilde{u}_{\text{prediction}}^\top \tilde{h}}{\|u_{\text{prediction}}\|^2} u_{\text{prediction}}. \quad (2)$$

This transformation modifies the hidden state only in the prediction direction $u_{\text{prediction}}$, leaving all orthogonal components unchanged (Figure 2d). In contrast to prior steering methods that apply fixed modifications regardless of the model’s internal state, our approach dynamically computes the minimal adjustment required based on the knowledge-prediction coordinates.

Generalized Alignment. While the original alignment constraint in Eq. 1 enforces strict equality between the knowledge and prediction coordinates, we extend our method to a more general form to capture more diverse knowledge-prediction relationships. We parameterize the target mapping with hyperparameters $w \in \mathbb{R}$ and $\beta \in \mathbb{R}$:

$$\tilde{u}_{\text{prediction}}^\top \tilde{h}' = w \cdot \tilde{u}_{\text{knowledge}}^\top \tilde{h} + \beta \cdot \text{sign}(\tilde{u}_{\text{knowledge}}^\top \tilde{h}). \quad (3)$$

The original constraint is recovered when $w = 1$ and $\beta = 0$, while larger values of w and β amplify the influence of internal knowledge on model predictions. This generalized formulation accommodates a broader range of relationships, with optimal (w, β) depending on the task. To provide a comprehensive understanding of our method, we visualize the resulting transformations of hidden states in Appendix C and analyze the parameter effects in §3.5.

3 EXPERIMENTS

We evaluate probe reliability (§3.1) and demonstrate KAPPA’s effectiveness against baselines on three benchmarks, including cross-dataset transfer (§3.2). Next, we show KAPPA resolves knowledge-prediction gaps in free-form generation and assess effects on general capabilities (§3.3, §3.4). Finally, we analyze hyperparameter effects (§3.5).

Common Experimental Setup To evaluate our method on reasoning- and knowledge-intensive abilities, we use subsets from three widely adopted MCQ benchmarks: Big-Bench Hard (BBH) (Suzgun et al., 2023), Massive Multi-task Language Understanding (MMLU) (Hendrycks et al., 2021), and AI2 Reasoning Challenge challenge set (ARC-Challenge) (Clark et al., 2018). The full list of subsets for each dataset is provided in Appendix D. As an initial step toward examining the knowledge-prediction gap, we simplify each multiple-choice question into a binary-choice format by pairing the correct answer with a randomly selected incorrect option, yielding BBH-Binary, MMLU-Binary, and ARC-Challenge-Binary. To mitigate the effects of option bias—where the model favors a particular choice (Zheng et al., 2024; Pezeshkpour & Hruschka, 2024)—we evaluate the base model across multiple prompt formats and select the format that yields the most balanced predictions on the validation set. Furthermore, all experiments are repeated with permuted answer orders. We evaluate on Llama-2-7B-Chat (Touvron et al., 2023) and Qwen2.5-7B-Instruct (Qwen et al., 2025), henceforth denoted as Llama-2 and Qwen-2.5, demonstrating the broad applicability of our method across transformer architectures. Additional implementation details are provided in Appendix D.

Baselines. We compare our method against the following baselines: (i) **Base**: the original LLM without any intervention. (ii) **Steering**: activation steering using logistic regression weights (Li et al., 2023). (iii) **LoRA FT** (Hu et al., 2022): LoRA fine-tuning on the train dataset. (iv) **ICL (BM25)** (Brown et al., 2020): in-context learning with BM25-retrieved demonstrations. Details are presented in Appendix E.2

3.1 LINEAR PROBE EVALUATION

To validate that hidden knowledge can be captured with linear probes, we report the test performance of our trained probes in Figure 3. Our analysis yields two key insights: First, linear probes reliably capture knowledge and prediction signals from hidden states in the middle-to-late layers, consistently achieving over 70% accuracy and in some cases exceeding 95%. Knowledge probe accuracy is low in early layers but rises in mid-to-late layers, with variation across models yet consistent layer patterns across datasets, aligning with prior work showing that LLMs encode richer semantic and knowledge representations in later layers (Skean et al., 2025; Meng et al., 2022; Dai et al., 2022). Second, knowledge probes often surpass the model’s own prediction accuracy. This occurs in at least one layer for every setting shown in Figure 3. In particular, on BBH-Binary with the Llama-2, the knowledge probe achieves over 70% accuracy, while the base model itself performs near chance level (50.9%). These findings indicate that models internally encode correct knowledge even when their predictions are incorrect, highlighting the significance of the knowledge-prediction gap.

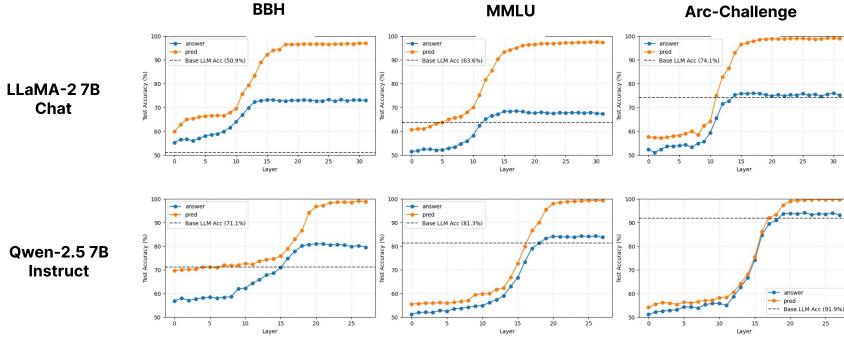


Figure 3: Accuracy of knowledge probes (blue) and prediction probes (orange) across transformer layers for LLaMA-2 and Qwen-2.5 on BBH-Binary, MMLU-Binary, and ARC-Challenge-Binary datasets. Dashed lines indicate base model accuracy for each dataset.

Method	Llama-2-7B-Chat			Qwen2.5-7B-Instruct		
	BBH	MMLU	ARC-Challenge	BBH	MMLU	ARC-Challenge
Base	50.9	63.6	74.1	71.1	80.8	91.9
Knowledge Probe	72.8	68.4	76.0	81.0	84.1	94.0
LoRA FT	79.2	75.6	82.0	86.1	84.6	94.6
ICL (BM25)	51.4	65.1	75.8	71.0	84.7	95.2
Steering	50.1	63.3	74.2	71.0	81.6	92.0
KAPPA (1-layer)	69.8	64.9	76.3	72.3	83.0	93.1
KAPPA (6-layer)	70.8	66.2	75.8	74.3	83.3	93.5

Table 1: Result of Multiple-Choice Intervention on binary-choice reformulations of BBH, MMLU, and ARC-Challenge datasets using LLaMA-2 and Qwen-2.5. Numbers represent test accuracy percentages. KAPPA variants indicate interventions applied to 1 layer or 6 layers, respectively.

3.2 MULTIPLE-CHOICE INTERVENTION

Experiment Setup. Using the knowledge and prediction probes trained on each layer for each dataset, we apply our method, KAPPA, at inference time. For analysis on generalization, we conduct a cross-dataset transfer experiment, where the knowledge-prediction subspace is identified using one dataset and then directly applied to another. This setup tests whether the subspaces are shared and reusable across distinct datasets. Further details on intervention layers and experimental settings are provided in Appendix E.1.

Results. Table 1 demonstrates KAPPA’s effectiveness across models and tasks, revealing several key insights about the knowledge-prediction gap phenomenon. We report average performance across subsets within each dataset and individual results are provided in Appendix H.

Approaching performance of the knowledge probe. Our knowledge probes always show superior performance compared to the base models (row 2 in Table 1), revealing two key insights. First, these probes serve approximate upper bounds for each model’s achievable performance without any external information. Second, the performance discrepancy between even high-performing models like Qwen-2.5 and their knowledge probes confirms that knowledge-prediction gaps persist across model, indicating substantial untapped potential within existing parameters. Given these insights, KAPPA’s effectiveness becomes clear: it consistently approaches the performance of knowledge probes, demonstrating successful unlocking of existing internal knowledge. For instance, KAPPA achieves consistent improvements over Qwen-2.5’s baseline (1% on BBH-Binary, 2% on MMLU-Binary and ARC-Challenge-Binary), revealing that even well-performing models contain internal knowledge that can be better expressed through knowledge-prediction alignment.

Substantial improvements on reasoning-intensive tasks. BBH-Binary serves as a crucial benchmark as it requires sophisticated reasoning that remains challenging even when simplified to binary choices—evidenced by LLaMA-2’s near-random baseline performance (50.9%). KAPPA achieves a remarkable 18.9% point improvement, transforming this into competent problem-solving (69.8%). Even the stronger Qwen-2.5 shows consistent 1-3% point gains. These results reveal a critical in-

Method	Train Data	Test Data			Qwen2.5-7B-Instruct		
		BBH	MMLU	ARC-Challenge	BBH	MMLU	ARC-Challenge
Base	-	50.9	63.6	74.1	71.1	81.3	91.9
ICL	BBH	51.4	66.0	73.5	71.0	83.5	94.3
	MMLU	52.6	65.1	74.4	69.5	84.7	95.0
	ARC-Challenge	51.5	66.9	75.8	69.2	83.4	95.2
LoRA FT	BBH	79.2	67.8	75.9	86.1	82.8	94.6
	MMLU	57.8	75.6	79.8	82.8	84.6	95.3
	ARC-Challenge	54.0	71.0	82.0	76.0	82.6	94.6
Steering	BBH	50.1	63.6	74.6	71.0	81.6	91.9
	MMLU	50.3	63.3	74.6	71.2	81.6	91.9
	ARC-Challenge	49.6	63.2	74.2	71.1	81.4	92.0
KAPPA (1-layer)	BBH	69.8	63.2	73.3	72.3	81.0	91.9
	MMLU	54.0	64.9	73.2	71.6	83.0	93.5
	ARC-Challenge	54.9	63.2	76.3	72.0	82.4	93.1
KAPPA (6-layers)	BBH	70.8	61.6	71.6	74.3	80.6	91.8
	MMLU	54.0	66.2	75.5	71.1	83.3	93.6
	ARC-Challenge	52.5	62.9	75.8	68.6	81.3	93.5

Table 2: Cross-dataset generalization results for KAPPA and baseline methods. Each method is trained using one dataset (rows) and applied to different test datasets (columns). Diagonal entries show in-domain performance; off-diagonal entries demonstrate cross-dataset transfer capability.

sight: models internally possess substantial knowledge but struggle to express it during choice selection. Our method successfully unlocks this latent capability through geometric realignment of representations, achieving substantial performance improvements without Chain-of-Thought prompting (Wei et al., 2022) or external knowledge injection.

Consistent gains across knowledge-intensive domains. On knowledge-heavy tasks (MMLU-Binary, ARC-Challenge-Binary), KAPPA shows steady 1-3% point improvements across both model families. While these gains appear modest compared to ICL baselines with Qwen-2.5, this reflects our fundamentally different objective. The knowledge probe performance (row 2 in Table 1) shows that internal knowledge ceilings are lower than ICL performance. Rather than competing with external information injection, KAPPA focuses on unlocking pre-existing capabilities, consistently approaching the knowledge probe ceiling to maximize utilization of encoded knowledge.

Generalization across diverse domains and tasks. KAPPA demonstrates cross-dataset generalization (Table 2), revealing that the knowledge-prediction subspace captures universal aspects of the internal knowledge rather than task-specific artifacts. Probes trained on MMLU-Binary and ARC-Challenge-Binary transfer effectively to BBH-Binary, improving Llama-2’s accuracy by up to 4% points and outperforming Steering by over 2% points. This subspace benefits even for a stronger model, with MMLU-trained probes improving Qwen-2.5’s ARC-Challenge performance by nearly 2% points. KAPPA maintains performance without significant degradation across datasets, demonstrating the robustness of the underlying knowledge-prediction geometry.

KAPPA unlocks models’ latent potential across formats. We hypothesize that when models answer correctly in free-form but fail in MCQs, they possess the knowledge but cannot express it effectively for the MCQs. As shown in Figure 4a, Llama-2 exhibits this inconsistency in 11.4% of cases (201 out of 1,759). KAPPA reduces this gap to 5.1%, outperforming strong baselines including LoRA FT, demonstrating that it effectively unlocks the model’s capability by calibrating model predictions with latent knowledge.

These patterns support our core hypothesis: LLMs often encode correct answers internally yet fail to express this knowledge in their predictions. By calibrating hidden states across the knowledge and prediction bases, KAPPA mitigates this limitation and enables effective access to internal knowledge across diverse tasks and architectures. Moreover, KAPPA achieves substantial improvements at inference time without additional training, supporting that model behaviors can be adapted even after deployment through lightweight interventions.

3.3 FREE-FORM GENERATION

Experimental Setup. We test whether KAPPA’s benefits can be extended to improve free-form generation directly. Specifically, we apply KAPPA during inference on the free-form versions of the same dataset used for training, to validate whether aligning the model’s internal state can enhance the quality and factual accuracy of its generated text. Details are provided in Appendix J.

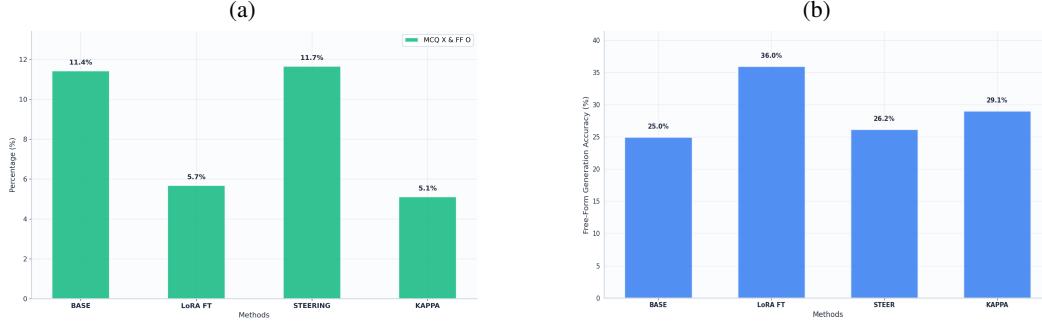


Figure 4: (a) Response alignment patterns for each method between MCQ and free-form formats. (b) Performance of each method in free-form and MCQ formats.

Results. We find that KAPPA’s benefits extend beyond MCQs to improve free-form generation. Figure 4b shows that applying KAPPA increases the accuracy of free-form answers from 25.0% to 29.1%—a significant 4.1% point improvement. While LoRA FT achieves a higher accuracy of 36.0%, KAPPA’s gains are achieved without any parameter updates. This result confirms that KAPPA’s success is not merely an artifact of manipulating binary choices but stems from an enhancement of the model’s ability to express its knowledge, indicating its potential to improve performance across a broader range of generative tasks.

3.4 IMPACT ON GENERAL CAPABILITY

Experimental Setup. We evaluate how KAPPA affects the general capabilities using `vicuna-eval` (Chiang et al., 2023), comparing against Steering and LoRA FT. For KAPPA and Steering, we apply probes trained on the BBH-Binary to the `Llama-2` model. Response quality is assessed with `gpt-5` (OpenAI, 2025), which compares outputs from KAPPA and the baselines, repeating each comparison with swapped response orders. Details are provided in Appendix K.

Result. We show that KAPPA enhances performance on knowledge- and reasoning-intensive tasks without harming the model’s ability to generate coherent responses. KAPPA performs on par with the baselines, while surpassing Steering (Table 3). KAPPA consistently achieves win rates above 50% in the *knowledge*, *counterfactual*, and *fermi*, which require models to retrieve facts and perform structured reasoning—whether through quantitative estimation, factual recall, or counterfactual inference. This highlights its strength in faithfully leveraging internal knowledge during generation.

Category	vs Steering (%)	vs LoRA FT (%)
Overall	51.2	46.9
fermi	65.0	65.0
counterfactual	60.0	75.0
knowledge	55.0	55.0
coding	57.1	28.6
math	50.0	33.3
generic	60.0	45.0
common-sense	40.0	25.0
roleplay	40.0	40.0
writing	35.0	40.0

Table 3: Win rates of KAPPA against Steering and LoRA FT across categories in `vicuna-eval`. Bold numbers indicate categories where KAPPA achieves at least 50% wins.

3.5 HYPERPARAMETER ANALYSIS

To examine the causal effects of the alignment parameters w and β in Equation 3, we conduct a hyperparameter sweep. Probes are trained and also evaluated on BBH-Binary. Our analysis shows

that varying w and β produces clear causal effects: larger values consistently improve accuracy and narrow the gap to the knowledge probe accuracy (Figure 5). Increasing either parameter amplifies the prediction coordinate, thereby steering outputs toward the knowledge probe’s preferred option. Notably, even with $w = 0$, a sufficiently large β alone can yield substantial gains. These findings demonstrate that the alignment parameters causally steer the model to better match its internal knowledge.

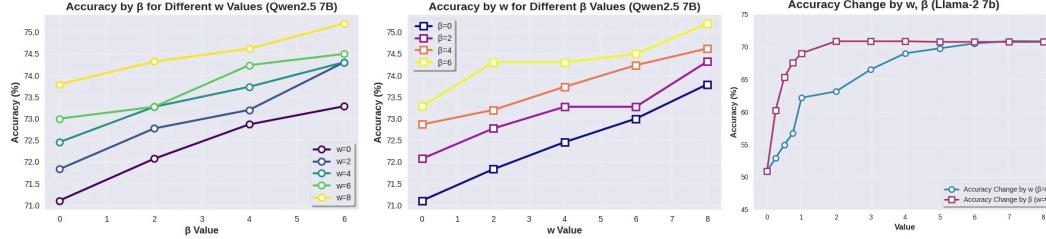


Figure 5: Effect of alignment parameters w and β on model accuracy. (**Left, Middle**) Results on Qwen-2 7B Instruct across five w values (0, 2, 4, 6, 8) and four β values (0, 2, 4, 6). (**Right**) Results on Llama-2 7B Chat, contrasting the cases where only w is varied ($\beta = 0, w \neq 0$) versus only β is varied ($w = 0, \beta \neq 0$). Across both models, increasing either parameter consistently improves accuracy, demonstrating the causal effect of amplifying prediction coordinates.

4 RELATED WORK

The Knowledge-Prediction Gap. Recent studies have highlighted the mismatch between LLMs’ internal knowledge and their external predictions. Prior work shows that linear probes can extract correct answers from hidden representations even when the model outputs are wrong (Marks & Tegmark, 2024; Liu et al., 2023; Azaria & Mitchell, 2023), and that these representations can also predict failures and even error types (Wang et al., 2024; Su et al., 2024; Orgad et al., 2025). However, existing approaches largely emphasize detection rather than characterizing the geometric structure of misalignment or closing the gap. Building on this, KAPPA both detects the model’s internal knowledge and aligns predictions with it.

Intervening in Model Behavior via Activation Steering. Several approaches have sought to directly modify model behavior through representation-level interventions (Zou et al., 2025; Ardit et al., 2024; Rimsky et al., 2024; Hendel et al., 2023; Ilharco et al., 2023). Li et al. (2023) introduced inference-time intervention by identifying linear directions that correlate with the desired concept and applying targeted modifications to internal representations without parameter updates. This approach has been extended to diverse objectives, including bias mitigation (Lu & Rimsky, 2024), refusal programming (Lee et al., 2025), and controlled generation (Dathathri et al., 2020; Turner et al., 2025). However, existing approaches primarily target behavioral shift and apply uniform interventions across all instances, without fully considering the latent knowledge. Our work proposes an adaptive intervention that dynamically computes corrections based on the knowledge and prediction coordinates.

5 CONCLUSION

This work addresses the *knowledge-prediction gap* in LLMs—the phenomenon where models internally encode correct knowledge yet fail to express it in their predictions. We show that this gap can be captured by a subspace of residual streams, spanned by a *knowledge basis* encoding ground-truth probabilities and a *prediction basis* encoding output probabilities. Our proposed method, KAPPA, dynamically calibrates activations along the two bases at inference time derived from closed-form optimization. Our analysis of binary-choice benchmarks shows that KAPPA yields substantial improvements on BBH-Binary and consistent performance gains across diverse evaluation settings. Furthermore, we find that the identified knowledge-prediction subspace generalizes across datasets, model architectures, and even free-form settings, suggesting it reflects a fundamental property of transformer representations. These findings provide both a new theoretical perspective

on the model’s failure mode and a practical method for eliciting more faithful expressions of internal knowledge. Future work will extend KAPPA from binary-choice to general multiple-choice settings, advancing its potential as a broadly applicable tool for aligning internal knowledge with predictions.

REFERENCES

- Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2017. URL <https://openreview.net/forum?id=ryF7rTqgl>.
- Andy Ardit, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 136037–136083. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f545448535dfde4f9786555403ab7c49-Paper-Conference.pdf.
- Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it’s lying. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.68. URL <https://aclanthology.org/2023.findings-emnlp.68/>.
- Yuntai Bao, Xuhong Zhang, Tianyu Du, Xinkui Zhao, Zhengwen Feng, Hao Peng, and Jianwei Yin. Probing the geometry of truth: Consistency and generalization of truth directions in LLMs across logical transformations and question answering tasks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 682–700, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.38. URL <https://aclanthology.org/2025.findings-acl.38/>.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS ’20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.
- Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. Answer matching outperforms multiple choice for language model evaluation, 2025. URL <https://arxiv.org/abs/2507.02856>.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. <https://lmsys.org/blog/2023-03-30-vicuna/>, March 2023. Accessed: 2025-09-25.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.
- Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.581. URL <https://aclanthology.org/2022.acl-long.581>.
- Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=H1edEyBKDS>.

- Roe H. Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors, 2023. URL <https://arxiv.org/abs/2310.15916>.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=d7KBjmI3GmQ>.
- 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. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=nZeVKeeFYf9>.
- Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic, 2023. URL <https://arxiv.org/abs/2212.04089>.
- Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=Oi47wc10sm>.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=aLLuYpn83y>.
- Kevin Liu, Stephen Casper, Dylan Hadfield-Menell, and Jacob Andreas. Cognitive dissonance: Why do language model outputs disagree with internal representations of truthfulness? In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4791–4797, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.291. URL <https://aclanthology.org/2023.emnlp-main.291/>.
- Dawn Lu and Nina Rimsky. Investigating bias representations in llama 2 chat via activation steering, 2024. URL <https://arxiv.org/abs/2402.00402>.
- Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets, 2024. URL <https://arxiv.org/abs/2310.06824>.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS ’22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.
- OpenAI. Introducing gpt-5. <https://openai.com/index/introducing-gpt-5/>, August 2025. Accessed: 2025-09-25.
- Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=KRnsX5Em3W>.
- Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2006–2017, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.130. URL <https://aclanthology.org/2024.findings-naacl.130/>.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL <https://arxiv.org/abs/2412.15115>.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL <https://aclanthology.org/2024.acl-long.828/>.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Schwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=WGXb7UdvTX>.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. Unsupervised real-time hallucination detection based on the internal states of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14379–14391, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.854. URL <https://aclanthology.org/2024.findings-acl.854/>.

Mirac Suzgun, Nathan Scales, Nathanael Schärfli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL <https://aclanthology.org/2023.findings-acl.824/>.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esibou, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL <https://arxiv.org/abs/2307.09288>.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2025. URL <https://openreview.net/forum?id=2XBPdP1cFK>.

Yanling Wang, Haoyang Li, Hao Zou, Jing Zhang, Xinlei He, Qi Li, and Ke Xu. Hidden question representations tell non-factualty within and across large language models. *CoRR*, abs/2406.05328, 2024. URL <https://doi.org/10.48550/arXiv.2406.05328>.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS ’22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=shr9PXz7T0>.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2025. URL <https://arxiv.org/abs/2310.01405>.

A LLM ACKNOWLEDGMENT

We use LLM to polish the writing and refactor our codebase.

B OPTIMIZED TRANSFORMATION OF KAPPA

Derivation. We rewrite our optimization problem Eq. 1 as

$$\min_{h'} \|h' - h\|_2^2 \quad \text{s.t.} \quad u_{\text{prediction}}^\top h' + b_{\text{prediction}} = u_{\text{knowledge}}^\top h + b_{\text{knowledge}}. \quad (4)$$

Equivalently, the constraint can be written as

$$u_{\text{prediction}}^\top h' = p_{\text{target}}, \quad p_{\text{target}} := u_{\text{knowledge}}^\top h + b_{\text{knowledge}} - b_{\text{prediction}}.$$

We derive the solution via Lagrange multipliers. Define the Lagrangian

$$\mathcal{L}(h', \lambda) = \|h' - h\|_2^2 + \lambda(u_{\text{prediction}}^\top h' - p_{\text{target}}),$$

with multiplier $\lambda \in \mathbb{R}$. Setting the gradient with respect to h' to zero gives

$$2(h' - h) + \lambda u_{\text{prediction}} = 0 \implies h' = h - \frac{\lambda}{2} u_{\text{prediction}}.$$

Substituting this into the constraint,

$$u_{\text{prediction}}^\top h - \frac{\lambda}{2} u_{\text{prediction}}^\top u_{\text{prediction}} = p_{\text{target}},$$

yields

$$\lambda = 2 \frac{u_{\text{prediction}}^\top h - p_{\text{target}}}{u_{\text{prediction}}^\top u_{\text{prediction}}}.$$

Hence, the closed-form solution is

$$h' = h + \frac{p_{\text{target}} - u_{\text{prediction}}^\top h}{u_{\text{prediction}}^\top u_{\text{prediction}}} u_{\text{prediction}}.$$

This coincides with the derived solution in Eq. 2, after resolving p_{target} .

Interpretations. The solution yields the closed-form update, which is an affine transformation of h :

$$h' = \left(I - \frac{u_{\text{prediction}} u_{\text{prediction}}^\top}{\|u_{\text{prediction}}\|^2} \right) h + \frac{p_{\text{target}}}{\|u_{\text{prediction}}\|^2} u_{\text{prediction}}.$$

Note that the original hidden state h can be decomposed as

$$h = \left(I - \frac{u_{\text{prediction}} u_{\text{prediction}}^\top}{\|u_{\text{prediction}}\|^2} \right) h + \frac{u_{\text{prediction}}^\top h}{\|u_{\text{prediction}}\|^2} u_{\text{prediction}}.$$

Compared to this decomposition, our update h' simply replaces the prediction coordinate $\frac{u_{\text{prediction}}^\top h}{\|u_{\text{prediction}}\|^2}$ with $\frac{p_{\text{target}}}{\|u_{\text{prediction}}\|^2}$, while leaving all orthogonal components unchanged. Geometrically, h' is the orthogonal projection of h onto the hyperplane $\{z \mid u_{\text{prediction}}^\top z = p_{\text{target}}\}$.

C TRANSFORMED DISTRIBUTION AFTER KAPPA

We visualize how the hidden states are transformed after applying KAPPA by projecting them onto the two-dimensional knowledge-prediction subspace. We consider two representative variants of KAPPA. KAPPA-Linear corresponds to the case with $w > 0$ and $\beta = 0$, where the prediction coordinate is scaled in proportion to the knowledge coordinate. KAPPA-Scalar corresponds to the case with $w = 0$ and $\beta > 0$, where the prediction coordinate is set to a fixed value determined by the sign of the knowledge coordinate. Figure 6 shows the distributions for the original LLM (left), KAPPA-Linear (middle), and KAPPA-Scalar (right).

In the original distribution, correct cases typically appear in the first and third quadrants, where the knowledge and prediction coordinates share the same sign. Incorrect cases, in contrast, appear in the second and fourth quadrants, where the coordinates are misaligned. This observation motivates KAPPA: by shifting hidden states in the direction of the prediction basis, points in the error regions (quadrants II and IV) can be moved toward the correct regions (quadrants I and III), thereby aligning the model’s predictions with its internal knowledge and reducing error cases. After applying KAPPA, we indeed observe that the transformed hidden states become more concentrated in the first and third quadrants, demonstrating that knowledge and behavior are effectively aligned through the intervention.

In the KAPPA-Linear case ($w > 0, \beta = 0$), the values on the knowledge and prediction coordinates are aligned along a line, indicating that prediction scores scale proportionally with the knowledge signal. By contrast, in the KAPPA-Scalar case ($w = 0, \beta > 0$), the prediction coordinate is aligned to a fixed value determined by the sign of the knowledge coordinate. This illustrates that KAPPA can induce qualitatively different forms of alignment between knowledge and prediction, depending on the choice of (w, β) .

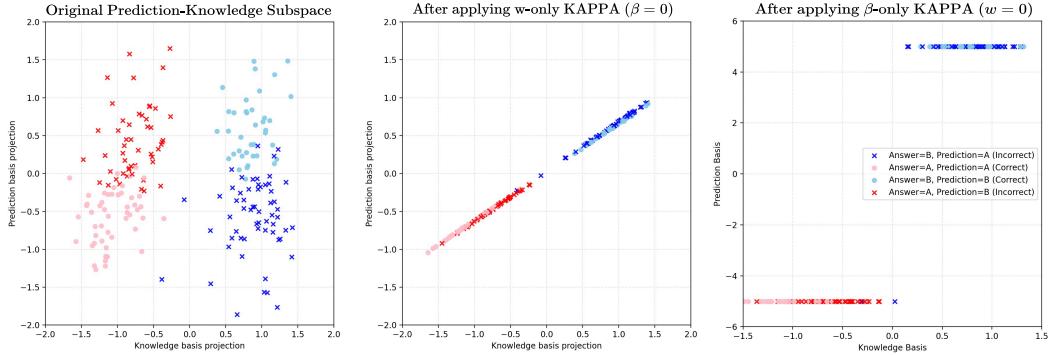


Figure 6: The distribution of hidden states projected to the 2D knowledge-prediction subspace. From the left, LLM, KAPPA-Linear, KAPPA-scalar.

D EXPERIMENTAL SETUP DETAILS

D.1 DATA SUBSETS

BBH-Binary Subsets We list below all the subsets included in the BBH-Binary benchmark.

Boolean Expressions	Date Understanding	Disambiguation QA
Geometric Shapes	Hyperbaton	Logical Deduction (5 Objects)
Logical Deduction (7 Objects)	Logical Deduction (3 Objects)	Movie Recommendation
Navigate	Penguins in a Table	Reasoning about Colored Objects
Ruin Names	Salient Translation Error Detection	Sports Understanding
Temporal Sequences	Tracking Shuffled Objects (5)	Tracking Shuffled Objects (7)
Tracking Shuffled Objects (3)	Web of Lies	

MMLU-Binary Subsets We list below all the subsets included in the MMLU-Binary benchmark.

Abstract Algebra	Anatomy	Astronomy
Business Ethics	Clinical Knowledge	College Biology
College Chemistry	College Computer Science	College Mathematics
College Physics	Computer Security	Conceptual Physics
Econometrics	Electrical Engineering	Elementary Mathematics
Global Facts	High School Biology	High School Chemistry
High School Computer Science	High School Geography	High School Government & Politics
High School Macroeconomics	High School Mathematics	High School Physics
High School Psychology	High School Statistics	High School US History
Human Aging	Human Sexuality	Jurisprudence
Management	Marketing	Medical Genetics
Miscellaneous	Moral Disputes	Moral Scenarios
Nutrition	Philosophy	Prehistory
Professional Accounting	Professional Medicine	Professional Psychology
Public Relations	Security Studies	Sociology
US Foreign Policy	Virology	World Religions

D.2 PROMPT FORMATS

To mitigate potential option bias, we varied the prompt format across 32 combinations, derived from four alternative instructions, two answer formats, and four option symbols. We present each component in Table 4, Table 5, and Table 6.

Instruction	Text
1	Provide your answer with {option_text}, following the output format below. Output Format: {format_text}
2	Given the following question and candidate answers, choose the correct option. Output Format: {format_text}
3	Choose the correct option between {option_text}. Use the format shown below when you answer. Output Format: {format_text}
4	Choose the best answer. Follow the output format below. Output Format: {format_text}

Table 4: Task instructions used in our experiments.

Answer Format Type	Format Text
1	The correct answer is (
2	Answer: (

Table 5: Answer format types used in experiments.

Option Symbol Type	Examples
Common Alphabet Upper	(A), (B)
Roman Numerals	(i), (ii)
Common Numbers	(1), (2)
Random Alphabet Upper	(Q), (K)

Table 6: Option symbol types used in experiments.

D.3 PROMPT EXAMPLE

We present one example of the prompt format used in our experiments, shown below.

You are given a question on college computer science. You are also given answer options, out of which only one is correct. You need to answer the question by selecting the correct option. You should only answer with the choice letter, not the whole answer. Provide your answer with (A) or (B), following the output format below.

Output Format: The correct answer is (A) or The correct answer is (B)

Many cryptographic protocols base their security on assumptions about the computational difficulty of integer factorization. Integer factorization serves this purpose because we believe that

Choices:

- (A) : testing primality is computationally intractable
- (B) : integer multiplication is a function whose inverse, factorization, remains difficult for a large class of inputs

E INTERVENTION EXPERIMENT DETAILS

E.1 KAPPA CONFIGURATIONS

For our primary evaluation, we select intervention layers based solely on the validation probe accuracy, without consulting the validation performance of the intervened model. This ensures that layer choices are determined independently of downstream results. We consider both single-layer and multi-layer interventions. In the single-layer setting, we intervene at layer 17 for LLaMA-2 and layer 20 for Qwen2.5. For the six-layer setting, we intervene at layers 12–17 for LLaMA-2 and 15–20 for Qwen2.5. This design allows us to test whether interventions at broader depth ranges yield more robust improvements. Regarding hyperparameters, we consider w and β . For simplicity, we fix $\beta = 0$ (KAPPA-Linear) in all experiments. We set $w = 5.0$ for LLaMA-2 and $w = 8.0$ for Qwen2.5.

E.2 BASELINE CONFIGURATIONS

Base. The original large language model (LLM) without any intervention or modification. This baseline reflects the model’s raw performance on the evaluation datasets.

Steering. Activation steering using logistic regression weights, following Li et al. (2023). We train a logistic regression classifier on hidden activations to separate correct vs. incorrect cases, and use the normalized weight vector as a steering direction applied at inference time. The intervention layer is fixed to be identical to the configuration used for KAPPA. The steering strength multiplier is selected from $\{1, 2, 4, 8\}$, tuned on the validation set to achieve the best performance. For generalization experiments, where the probe is trained on one dataset and directly applied to another, we apply the same multiplier value that achieved the best validation performance on the training dataset, without additional tuning on the out-of-distribution dataset.

LoRA Fine-tuning (LoRA FT). Parameter-efficient fine-tuning with LoRA adapters applied to the attention projection layers. We fine-tune the model on the same training split as used for our probe training, selecting hyperparameters (rank, learning rate, number of steps) by validation performance. This baseline represents a strong adaptation method requiring training and weight updates. For ARC-Challenge-Binary, we use $r = 16$, $\alpha = 32$, and set LORA dropout to 0.1. We use an effective batch size of 8 in training time, run for 1 epoch with $5e^{-5}$ learning rate. For BBH, we use $r = 8$, $\alpha = 16$, and set LORA dropout to 0.05. We use an effective batch size of 4 in training time, run for 1 epoch with $2e^{-4}$ learning rate. For BBH, we use $r = 64$, $\alpha = 64$, and set LORA dropout to 0.1. We use an effective batch size of 16 in training time, run for 1 epoch with $3e^{-5}$ learning rate.

In-Context Learning (ICL). We provide the model with demonstrations retrieved from the training dataset using BM25 retrieval. At inference time, for each test instance, we retrieve $k = 3$ examples and prepend them to the prompt in a few-shot format. This baseline reflects a retrieval-augmented prompting strategy without parameter updates.

F ADDITIONAL PROBING RESULTS

In this section, we provide supplementary probing results to complement the main findings reported in § 3.1. These results further validate that the identified knowledge and prediction directions are consistent across models and datasets. Figure 7 presents results on MMLU-Binary using Llama-2, showing that the probe trained to separate correct answers from distractors achieves stable performance across layers. Figure 8 shows results on BBH-Binary for both Llama-2 (upper) and Qwen-2.5 (lower). We observe broadly consistent patterns across the two models, indicating that

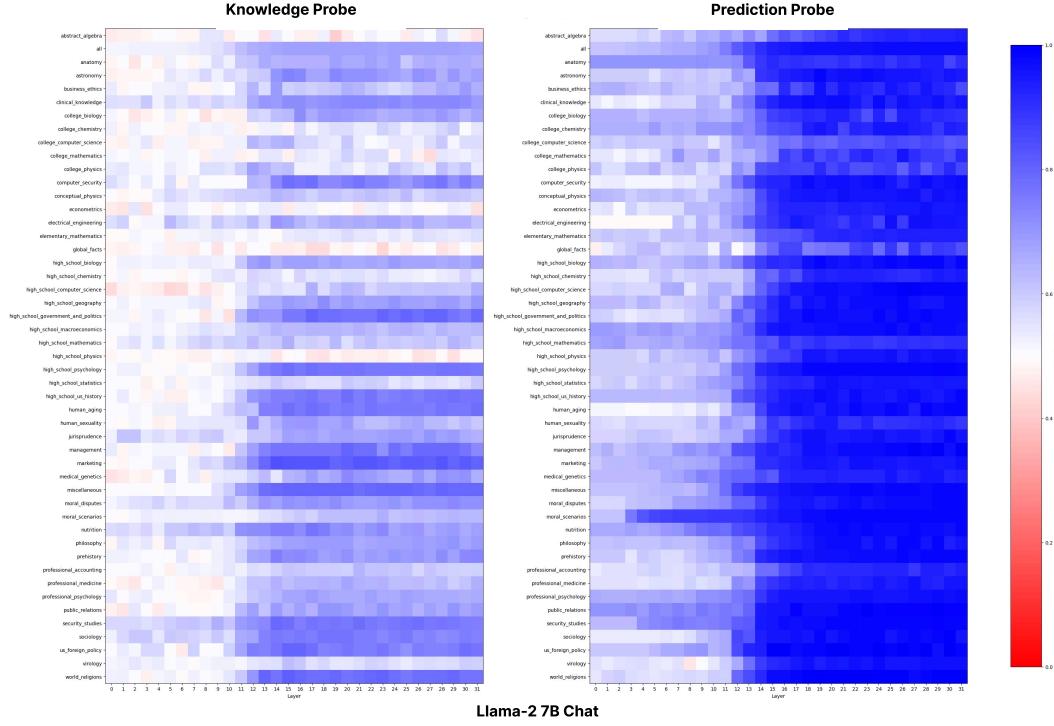


Figure 7: Probing results on MMLU-Binary using Llama-2.

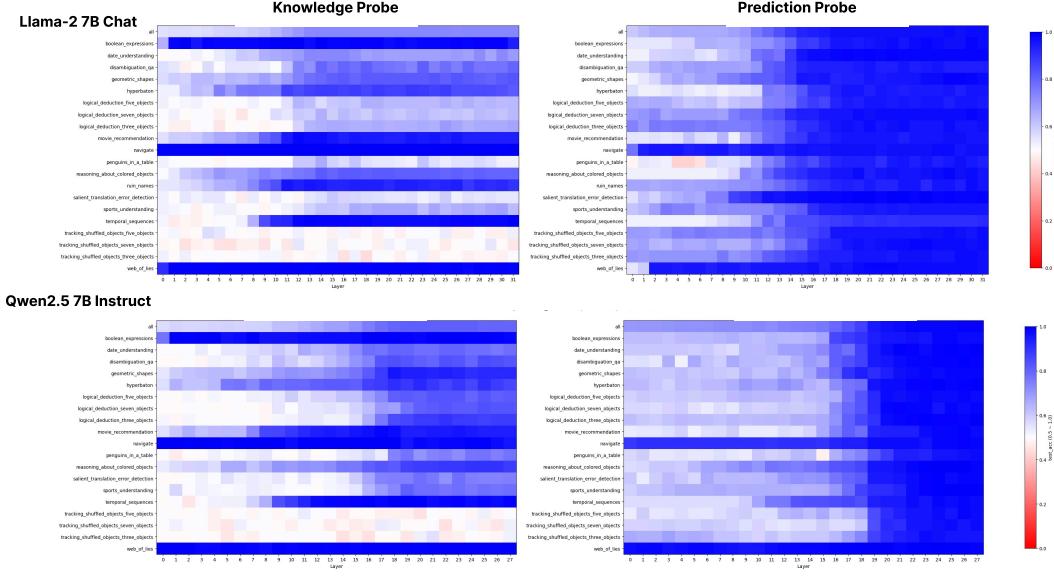


Figure 8: Probing results on BBH-Binary using Llama-2 (upper) and Qwen-2.5 (lower).

the knowledge-prediction subspace is not specific to a single model family but extends across architectures. Unlike the shared probes used in the main text, here we train a separate probe for each subset individually. While some subsets show clear improvements in probe accuracy, others exhibit limited gains. We attribute these limited improvements to cases where the model itself lacks sufficient internal knowledge. Overall, these additional probing results provide further evidence that the knowledge and prediction signals identified by our method are robust, reproducible, and generalizable across datasets and model backbones.

G ADDITIONAL INTERVENTION RESULTS

We further report additional results in Figure 9 using the Llama-2 model on the BBH-Binary dataset. In this setting, we train a separate probe for each subset and validate across eight layer configurations (12, 13, 14, 15, 16, 17, 15–17, and 12–17), selecting the best-performing layer per subset. Applying KAPPA-Scalar ($w = 0, \beta = 5$), we achieve 75.3% accuracy, which even surpass the accuracy of the best knowledge probe trained jointly on all subsets (72.8%). This indicates that while a shared subspace exists across different reasoning tasks, the optimal subspace varies by task, suggesting that there remains room for further optimization when applying our method to downstream tasks. We further observe that knowledge probe accuracy varies across subsets, and the intervened accuracy tends to increase accordingly, following the same trend. This indicates that our method effectively aligns model behavior with the knowledge captured by the probe.

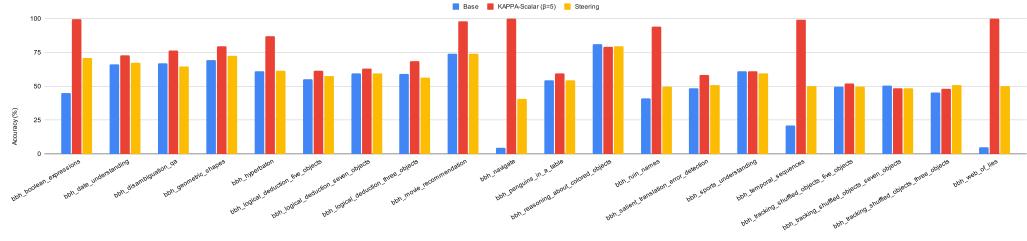


Figure 9: Subset-wise results on the BBH-Binary dataset with Llama-2. For each subset, we train a separate knowledge probe and apply KAPPA-Scalar ($w = 0, \beta = 5$) at the best-performing layer. We observe that higher knowledge probe accuracy is accompanied by higher intervened accuracy, indicating that our method effectively aligns the model’s predictions with the knowledge captured by the probe.

H FULL INTERVENTION RESULTS

Detailed results for all datasets are provided in Table 7 (Llama-2) and Table 8 (Qwen-2.5).

Table 7: Llama-2 full intervention results. Accuracy (%) across datasets with different intervention layers.

Dataset - Subset	Base Acc	1-Layer (17)	6-Layers (12–17)
ARC - Challenge	74.1	76.3	75.8
BBH - Boolean Expressions	45.0	92.5	96.5
BBH - Date Understanding	66.0	66.5	65.5
BBH - Disambiguation QA	67.0	73.0	71.5
BBH - Geometric Shapes	69.5	57.0	68.5
BBH - Hyperbaton	61.0	78.5	77.0
BBH - Logical Deduction (5 Objects)	55.0	61.0	60.0
BBH - Logical Deduction (7 Objects)	59.5	60.0	62.5
BBH - Logical Deduction (3 Objects)	59.0	64.0	63.5
BBH - Movie Recommendation	74.0	82.0	87.0
BBH - Navigate	4.5	99.0	98.0
BBH - Penguins in a Table	54.2	58.5	60.2
BBH - Reasoning About Colored Objects	81.0	79.0	78.0
BBH - Ruin Names	41.0	76.5	82.0
BBH - Salient Translation Error Detection	48.5	55.5	55.5
BBH - Sports Understanding	61.0	53.5	57.0
BBH - Temporal Sequences	21.0	95.5	91.5
BBH - Tracking Shuffled Objects (5 Objects)	49.5	51.0	49.0
BBH - Tracking Shuffled Objects (7 Objects)	50.5	49.0	50.0
BBH - Tracking Shuffled Objects (3 Objects)	45.5	50.5	50.5
BBH - Web of Lies	5.0	93.0	91.5

Continued on next page

Table 7: (continued)

Dataset - Subset	Base Acc	1-Layer (17)	6-Layers (12–17)
BBH - Average	50.9	69.8	70.8
MMLU - Abstract Algebra	63.8	52.5	47.5
MMLU - Anatomy	63.0	63.9	64.8
MMLU - Astronomy	68.0	68.0	68.9
MMLU - Business Ethics	66.3	60.0	65.0
MMLU - Clinical Knowledge	72.2	75.5	74.1
MMLU - College Biology	66.4	69.0	65.5
MMLU - College Chemistry	60.0	55.0	50.0
MMLU - College Computer Science	57.5	63.8	65.0
MMLU - College Mathematics	50.0	53.8	52.5
MMLU - College Physics	42.7	46.3	51.2
MMLU - Computer Security	72.5	71.3	75.0
MMLU - Conceptual Physics	60.6	61.7	61.7
MMLU - Econometrics	54.3	53.3	54.4
MMLU - Electrical Engineering	52.6	59.5	61.2
MMLU - Elementary Mathematics	53.0	54.0	54.6
MMLU - Global Facts	55.0	55.0	55.0
MMLU - High School Biology	68.5	67.7	69.0
MMLU - High School Chemistry	54.3	56.1	56.7
MMLU - High School Computer Science	60.0	60.0	61.3
MMLU - High School Geography	68.8	68.8	70.6
MMLU - High School Government and Politics	77.6	77.6	80.1
MMLU - High School Macroeconomics	60.6	62.5	67.0
MMLU - High School Mathematics	48.6	52.8	55.6
MMLU - High School Physics	50.0	49.2	49.2
MMLU - High School Psychology	75.2	76.2	77.3
MMLU - High School Statistics	55.2	58.6	58.1
MMLU - High School US History	73.2	74.4	74.4
MMLU - Human Aging	78.3	75.0	75.6
MMLU - Human Sexuality	60.4	68.9	71.7
MMLU - Jurisprudence	64.8	67.1	67.1
MMLU - Management	69.0	75.0	77.4
MMLU - Marketing	75.0	78.2	84.0
MMLU - Medical Genetics	60.0	65.0	65.0
MMLU - Miscellaneous	76.4	77.9	78.0
MMLU - Moral Disputes	70.5	74.5	73.0
MMLU - Moral Scenarios	49.2	49.4	50.6
MMLU - Nutrition	62.2	64.2	69.9
MMLU - Philosophy	65.2	67.2	68.8
MMLU - Prehistory	66.9	70.0	72.7
MMLU - Professional Accounting	59.7	58.4	59.7
MMLU - Professional Medicine	66.1	67.4	67.4
MMLU - Professional Psychology	62.9	67.1	66.9
MMLU - Public Relations	68.2	70.5	72.7
MMLU - Security Studies	62.8	65.8	68.9
MMLU - Sociology	75.3	75.3	75.9
MMLU - US Foreign Policy	72.5	75.0	77.5
MMLU - Virology	60.4	59.7	64.9
MMLU - World Religions	79.0	78.3	82.6
MMLU - Average	63.6	64.9	66.2

Table 8: Qwen2 . 5 full intervention results. Accuracy (%) across datasets with different intervention layers.

Dataset - Subset	Base Acc	1-Layer (?)	6-Layers (?)
ARC - Challenge	91.9	93.1	93.5
BBH - Boolean Expressions	51.0	51.5	51.5

Continued on next page

Table 8: (continued)

Dataset - Subset	Base Acc	1-Layer (?)	6-Layers (?)
BBH - Date Understanding	76.5	76.5	76.5
BBH - Disambiguation QA	74.5	74.0	74.5
BBH - Geometric Shapes	77.5	79.5	81.5
BBH - Hyperbaton	64.5	68.0	75.0
BBH - Logical Deduction (5 Objects)	79.0	81.5	83.0
BBH - Logical Deduction (7 Objects)	76.5	80.0	81.5
BBH - Logical Deduction (3 Objects)	83.5	83.0	84.0
BBH - Movie Recommendation	57.5	59.5	71.0
BBH - Navigate	90.5	91.0	92.5
BBH - Penguins in a Table	74.6	72.9	73.7
BBH - Reasoning About Colored Objects	86.0	87.0	89.0
BBH - Ruin Names	61.5	63.5	69.0
BBH - Salient Translation Error Detection	71.5	72.5	72.0
BBH - Sports Understanding	73.5	73.5	73.5
BBH - Temporal Sequences	80.0	85.0	93.0
BBH - Tracking Shuffled Objects (5 Objects)	49.0	51.0	50.0
BBH - Tracking Shuffled Objects (7 Objects)	51.0	51.0	49.5
BBH - Tracking Shuffled Objects (3 Objects)	45.5	45.5	44.5
BBH - Web of Lies	98.5	100.0	100.0
BBH - Average	71.1	72.3	74.3
MMLU - Abstract Algebra	65.0	66.3	60.0
MMLU - Anatomy	76.9	77.8	78.7
MMLU - Astronomy	90.2	93.4	93.4
MMLU - Business Ethics	81.3	78.8	77.5
MMLU - Clinical Knowledge	92.5	92.9	92.0
MMLU - College Biology	96.6	94.8	94.8
MMLU - College Chemistry	67.5	63.8	67.5
MMLU - College Computer Science	82.5	80.0	83.8
MMLU - College Mathematics	62.5	73.8	72.5
MMLU - College Physics	69.5	75.6	73.2
MMLU - Computer Security	81.3	82.5	83.8
MMLU - Conceptual Physics	81.9	85.1	84.0
MMLU - Econometrics	78.3	83.7	84.8
MMLU - Electrical Engineering	77.6	79.3	80.2
MMLU - Elementary Mathematics	80.9	80.9	82.2
MMLU - Global Facts	62.5	62.5	60.0
MMLU - High School Biology	89.5	90.3	90.7
MMLU - High School Chemistry	78.7	75.6	79.9
MMLU - High School Computer Science	78.8	80.0	80.0
MMLU - High School Geography	86.9	89.4	90.6
MMLU - High School Gov. & Politics	95.5	95.5	93.6
MMLU - High School Macroeconomics	88.8	89.0	88.8
MMLU - High School Mathematics	66.2	68.5	69.0
MMLU - High School Physics	78.7	78.7	77.9
MMLU - High School Psychology	89.5	91.3	92.7
MMLU - High School Statistics	82.2	86.2	86.8
MMLU - High School US History	90.2	91.5	92.1
MMLU - Human Aging	83.3	83.3	85.0
MMLU - Human Sexuality	85.9	89.6	89.6
MMLU - Jurisprudence	81.8	84.1	85.2
MMLU - Management	86.9	92.9	94.1
MMLU - Marketing	93.1	93.1	93.6
MMLU - Medical Genetics	90.0	88.8	90.0
MMLU - Miscellaneous	88.2	89.7	90.1
MMLU - Moral Disputes	80.9	85.3	84.2
MMLU - Moral Scenarios	53.6	55.2	54.2
MMLU - Nutrition	82.9	85.8	86.6
MMLU - Philosophy	80.0	83.2	81.2
MMLU - Prehistory	84.6	86.5	86.9
MMLU - Professional Accounting	73.5	72.6	73.9

Continued on next page

Table 8: (continued)

Dataset - Subset	Base Acc	1-Layer (?)	6-Layers (?)
MMLU - Professional Medicine	83.9	83.5	85.3
MMLU - Professional Psychology	80.8	83.5	83.9
MMLU - Public Relations	83.0	88.6	89.8
MMLU - Security Studies	76.0	84.7	84.2
MMLU - Sociology	90.1	89.5	88.3
MMLU - US Foreign Policy	88.8	91.3	95.0
MMLU - Virology	68.7	69.4	70.2
MMLU - World Religions	92.8	94.9	97.1
MMLU - Average	81.3	83.0	83.3

I STEERING BASELINE EXPERIMENTAL DETAILS

We summarize the steering baseline configuration in Table 9. For both `Llama-2` and `Qwen2.5`, we fix the intervention layer in advance, normalize each probe weight vector, and choose the best-performing multiplier from {1, 2, 4, 8} using validation accuracy on the training dataset. When applying to other datasets for generalization, we consistently use the same layer and multiplier configuration without additional tuning.

Table 9: Configuration of the Steering baseline. We fix the intervention layer per model, normalize the probe vector, and select the best multiplier from {1, 2, 4, 8} based on validation performance. The same configuration is applied when generalizing to other datasets.

Model	Training Dataset	Layer	Multiplier
<code>Llama-2</code>	BBH	17	8.0
	MMLU	17	4.0
	ARC	17	1.0
<code>Qwen2.5</code>	BBH	20	1.0
	MMLU	20	4.0
	ARC	20	4.0

J FREE-FORM GENERATION DETAILS

J.1 EXPERIMENTAL DETAILS

Dataset Construction. We construct a free-form version of the BBH dataset by systematically removing multiple-choice options from the original questions. To ensure the questions remain suitable for open-ended evaluation, we filter out items that inherently require option-based responses, such as those beginning with “Which statement best explains...” or similar phrasings that presuppose a selection from given alternatives. This filtering process yields 1,759 free-form questions that maintain the core reasoning challenges of the original BBH benchmark while enabling open-ended response evaluation.

Evaluation Methodology. For assessment, we employ an LLM-as-a-Judge approach, following established practices in recent work (Chandak et al., 2025). The judge model evaluates whether the ground-truth answer is semantically contained within or equivalent to the model’s generated response, allowing for flexible matching that accounts for varied expression while maintaining accuracy standards.

J.2 EVALUATION PROMPT

Your task is to judge whether the given response to a question matches a given ground truth answer or not. You are provided with a question, a ground truth response, and the response you need to judge.

For a response to "match", it must have at least as much information as the ground-truth.

The response can have more information than the ground-truth. It can be more specific (for example, "Labrador" is more specific than "dog"), or have additional possible correct answers. But it must cover everything mentioned in the ground-truth. It is okay if it covers it in different words, i.e. paraphrased.

For numeric answers, the relative error, defined as $|\text{response} - \text{ground truth}| / \text{mean}(\text{response}, \text{ground truth})$, must be less than 1\% for the response to be judged as a correct match. Here, if the ground truth is a specific numeric quantity but the response is a range, then they don't match (even if the range contains the ground truth).

Possible judgments:

"0": The response does not match the ground-truth answer.

"1": The response matches the ground-truth.

Question: [question]

Ground truth: [target]

Response: [response]

Your job is to ONLY check whether the given response matches the ground truth answer or not in the context of the question. You DO NOT NEED to assess the correctness of the response. This is part of an automated evaluation process, therefore you MUST OUTPUT your final answer as "0" or "1" in <answer> </answer> tags.

K GENERAL CAPABILITY EVALUATION DETAILS

K.1 EVALUATION PROMPT

We include the full prompts used for evaluation across different categories (*general*, *coding*, and *math*). For each category, we present three components: (i) the **system prompt**, (ii) the **format prompt**, which specifies the evaluation structure, and (iii) the **task instruction**, describing the specific evaluation criteria. In constructing the final evaluation input, the format prompt is concatenated with the task instruction to form the **user message**. Within the format prompt, the original question is inserted together with two model responses: one from KAPPA and one from the baseline. To mitigate order bias, both possible permutations of the two responses are provided to the evaluator.

System Prompt:

You are a helpful and precise assistant for checking the quality of the answer.

Format:

[Question]
{question}

[The Start of Assistant 1's Answer]
{answer_1}
[The End of Assistant 1's Answer]

[The Start of Assistant 2's Answer]
{answer_2}
[The End of Assistant 2's Answer]

Generic Task Instructions:

We would like to request your feedback on the performance of two AI assistants
in response to the user question displayed above.

Please rate the helpfulness, relevance, accuracy, level of details of their responses.

Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance.

Please first output a single line containing only two values indicating the scores for Assistant 1 and 2, respectively. The two scores are separated by a space.

In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.

Coding Task Instructions:

Your task is to evaluate the coding abilities of the above two assistants. They have been asked to implement a program to solve a given problem. Please review their code submissions, paying close attention to their problem-solving approach, code structure, readability, and the inclusion of helpful comments.

Please ensure that the assistants' submissions:

- Correctly implement the given problem statement.
- Contain accurate and efficient code.
- Include clear and concise comments that explain the code's logic and functionality.
- Adhere to proper coding standards and best practices.

Once you have carefully reviewed both submissions, provide detailed feedback on their strengths and weaknesses, along with any suggestions for improvement. You should first output a single line containing two scores on the scale of 1-10 (1: no code/no sense; 10: perfect) for Assistant 1 and 2, respectively. Then give extra comments starting from the next line.

Mathematical Task Instructions:

We would like to request your feedback on the mathematical proficiency of two AI assistants regarding the given user question displayed above. First, please solve the problem independently, without referring to the answers provided by Assistant 1 and Assistant 2.

Afterward, please examine the problem-solving process of Assistant 1 and Assistant 2 step-by-step to ensure their correctness, identifying any incorrect steps if present. Your evaluation should take into account not only the answer but also the problem-solving steps.

Finally, please output a Python tuple containing two numerical scores for Assistant 1 and Assistant 2, ranging from 1 to 10, respectively. If applicable, explain the reasons for any variations in their scores and determine which assistant performed better.