Discovering a Shared Logical Subspace: Steering LLM Logical Reasoning via Alignment of Natural-Language and Symbolic Views

Feihao Fang[1] **My T. Thai**[2] **Yuanyuan Lei**[2]

[1]University of Illinois Urbana-Champaign, Champaign, IL
[2]Computer & Information Science and Engineering, University of Florida, Gainesville, FL
feihaof2@illinois.edu, yuanyuan.lei@ufl.edu

Abstract

Large Language Models (LLMs) still struggle with multi-step logical reasoning. Existing approaches either purely refine the reasoning chain in natural language form or attach a symbolic solver as an external module. In this work, we instead ask whether LLMs contain a shared internal *logical subspace* that simultaneously aligns natural-language and symbolic-language views of the reasoning process. Our hypothesis is that this logical subspace captures logical reasoning capabilities in LLMs that are shared across views while remaining independent of surface forms. To verify this, we employ Canonical Correlation Analysis on the paired residual activations from natural-language and symbolic-language reasoning chains, learning a low-dimensional subspace with maximum cross-view correlation. Furthermore, we design a training-free approach that steers LLMs reasoning chain along this logical subspace, thereby leveraging the complementary reasoning signals from both views. Experiments on four logical reasoning benchmarks demonstrate the effectiveness of our approach, improving accuracy by up to 11 percentage points and generalizing well on out-of-domain problems[1].

1 Introduction

Logical reasoning in LLMs refers to their ability to follow rules, connect premises to conclusions, and carry out multi-step inferences (Liu et al., 2023b). Despite recent advances in natural-language understanding, LLMs still fall short on complex, multi-step logical reasoning problems (Xu et al., 2025). Strong logical reasoning capability is important for applications that require multi-step decision making, such as math, scientific analysis, planning, coding etc. (Cheng et al., 2025). Thus, improving LLM logical reasoning is a critical problem.

[1]The code and data link is: https://github.com/lei-nlp-lab/logical_subspace_acl_2026

Natural-Language Proof	**Symbolic-Language Proof**
Context. Every bird can fly. Tweety is a bird. **Claim.** Tweety can fly. **Proof.** Tweety is a bird. All birds can fly. Therefore, Tweety can fly.	**Facts.** $\text{bird}(\text{tweety})$ $\forall x\,(\text{bird}(x) \rightarrow \text{can\_fly}(x))$ **Goal.** $\text{can\_fly}(\text{tweety})$ **Proof.** $\text{bird}(\text{tweety})$. $\forall x\,(\text{bird}(x) \rightarrow \text{can\_fly}(x))$. $\text{can\_fly}(\text{tweety})$.

Figure 1: An example of a logical reasoning chain expressed in natural language and symbolic language.

Prior works can be broadly grouped into two streams: natural-language-dependent methods and neural-symbolic methods. The first line of work uses prompting or training-based techniques to optimize natural-language chain-of-thought traces during decoding or training (Wei et al., 2022; Zou et al., 2025b). Another line of neural-symbolic approaches augment LLMs with external symbolic provers or verifiers (Pan et al., 2023; Lei and Huang, 2024). However, these methods either focus solely on refining reasoning in natural-language form or rely on externally attached symbolic components. In this work, we introduce a new framework that aligns natural-language and symbolic views of the logical reasoning process, thus leveraging complementary reasoning signals from both views.

Our idea is inspired by the observation that each logical reasoning problem can be solved in two complementary representations of the same underlying derivation: (i) a natural-language (NL) proof written as a step-by-step verbal explanation, and (ii) a corresponding symbolic proof expressed as a sequence of rules or logical clauses, as illustrated in Figure 1. The two expressions agree on the logical reasoning process but differ in surface form: the symbolic proof exposes formal structure, while the NL proof reflects how a model verbalizes that structure. This motivates our core research ques-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 38936–38956

tion: *do LLMs contain a shared internal logical subspace that aligns with both views, and can we exploit it to enhance LLM logical reasoning?*

We address this question by hypothesizing that there exists a low-dimensional *multi-view logical subspace* in LLMs, in which activations from paired natural-language and symbolic proofs of the same instance are strongly aligned. To instantiate this idea, we employ Canonical Correlation Analysis algorithm (Raghu et al., 2017) on the paired residual activations from both natural-language and symbolic reasoning chains, learning a shared low-dimensional subspace. By maximizing cross-view correlation, we expect this subspace to capture logical reasoning capabilities in LLMs that are shared across views and independent of surface forms.

Furthermore, to fully exploit the complementary reasoning signals contained in the shared logical subspace, we design a training-free approach that nudges the LLM's reasoning chain at inference time: as the model autoregressively generates each token, we linearly amplify the projection of each token's activation onto the learned subspace, thereby steering the hidden state towards the shared NL-symbolic subspace while leaving model weights unchanged. Our innovation over prior work is that our method moves beyond single-view heuristics and integrates complementary reasoning signals through cross-view alignment, without requiring additional training or external symbolic solvers.

Experiments across four logical reasoning benchmarks and five LLMs demonstrate the effectiveness of our approach, improving accuracy by up to 11%. The learned logical subspace also generalizes well to out-of-distribution reasoning problems. Moreover, our analysis reveals several insights: (i) the logical subspace in LLMs encodes both semantic and logical structure information (ii) alignment between natural-language and symbolic views strengthens in higher layers of LLMs (iii) projection energy within the learned logical subspace correlates positively with reasoning correctness (iv) steering along the logical subspace increases the use of logical connective words such as *since*, *so*, while decreasing reliance on vague reasoning verbs such as *think*, *know*, *assume*.

Our contributions can be summarized as:

- We discover a shared internal *logical subspace* in LLMs that aligns both natural-language and symbolic views of the reasoning process.
- We propose a training-free method that steers LLMs' generation along the logical subspace, thus leveraging cross-view reasoning signals.
- We demonstrate up to 11% improvement on logical reasoning, show strong out-of-domain generalization, and provide analysis revealing the structure and behavior of the subspace.

2 Background and Problem Setup

2.1 Model and Residual Stream

We consider decoder-only language models with L Transformer layers and residual dimension D. Given an input token sequence $x_{1:T}$, let $h_t^{(\ell)} \in \mathbb{R}^D$ denote the residual activation at layer ℓ and position t. Following prior work on activation-level interventions (Zou et al., 2025a; Turner et al., 2024), we treat the residual stream $\{h_t^{(\ell)}\}$ as the main object of our analysis and steering.

2.2 Multi-View Reasoning Data

Each reasoning instance i consists of facts and rules, a query, a truth label, and two proof-style reasoning trajectory of the same derivation: a *natural-language proof* P_i^{NL} written as a step-by-step explanation, and a *symbolic proof* P_i^{Sym} given as a sequence of rule applications or logical clauses.

For each view $v \in \{\mathrm{NL}, \mathrm{Sym}\}$, we concatenate the context, query, and proof in that view into a single input and run the frozen LLM in teacher forcing, collecting residual activations $h_{i,t,v}^{(\ell)} \in \mathbb{R}^D$ at every layer ℓ and position t.

Let $\mathcal{I}_i^v$ be the index set of proof tokens for instance i in view v (excluding context and question tokens). At each layer ℓ, we simply mean-pool residual activations over the proof tokens in each view, yielding $r_{i,\mathrm{NL}}^{(\ell)}$ and $r_{i,\mathrm{Sym}}^{(\ell)}$ for instance i.

Stacking the pooled representations across instances yields, for each layer ℓ, two matrices:

$$X^{(\ell)} \in \mathbb{R}^{N \times D}, \quad Y^{(\ell)} \in \mathbb{R}^{N \times D},$$

whose i-th rows are $r_{i,\mathrm{NL}}^{(\ell)}$ and $r_{i,\mathrm{Sym}}^{(\ell)}$, respectively. We refer to $X^{(\ell)}$ and $Y^{(\ell)}$ as the NL and symbolic views, following standard Canonical Correlation Analysis (CCA) notation (Raghu et al., 2017).

2.3 Subspaces and Projectors

At selected layers, we will learn from the paired NL and symbolic views a low-dimensional subspace of the residual stream for each layer ℓ, represented

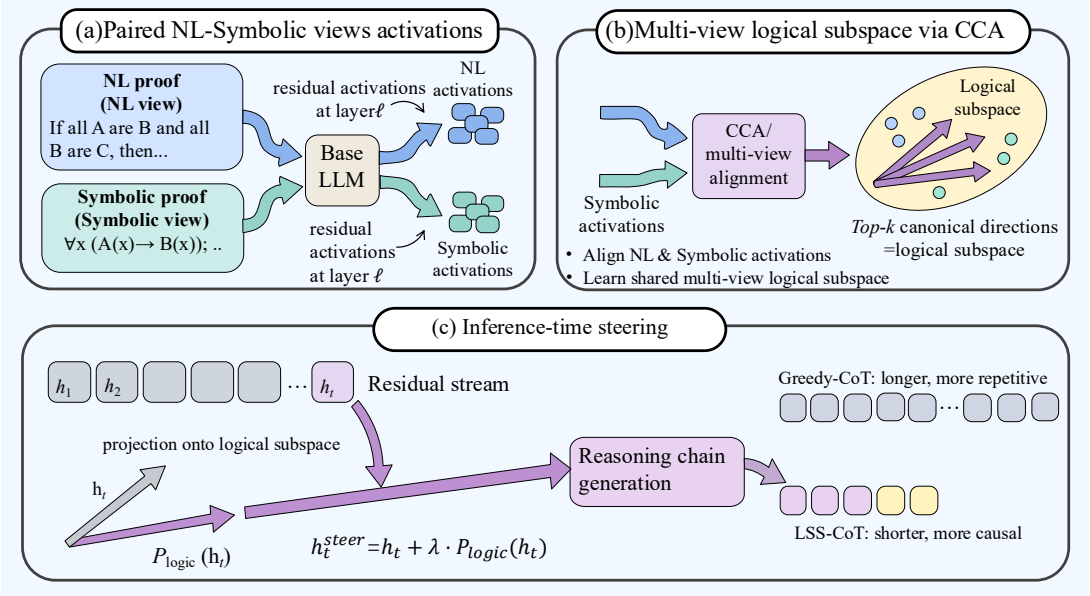


Figure 2: Overview of our logical subspace steering (LSS) method.

by an orthonormal basis $U^{(\ell)} \in \mathbb{R}^{D\times k}$ and its orthogonal projector $P^{(\ell)} = U^{(\ell)}U^{(\ell)\top}$ (details in Section 3.1). We refer to the column span of $U^{(\ell)}$ as the layer-ℓ *multi-view logical subspace*.

2.4 Token-level projection energy

Given a layer-ℓ residual activation $r \in \mathbb{R}^D$ and the logical subspace basis $U^{(\ell)} \in \mathbb{R}^{D\times k}$ (Section 3.1), we define the normalized projection energy as:

$$E^{(\ell)}(r) = \frac{\left\|r^\top U^{(\ell)}\right\|_2^2}{\|r\|_2^2}. \tag{1}$$

We also define the contribution of the j-th basis direction $u_j^{(\ell)}$ as:

$$E_j^{(\ell)}(r) = \frac{\left(r^\top u_j^{(\ell)}\right)^2}{\|r\|_2^2}, \quad E^{(\ell)}(r) = \sum_{j=1}^{k} E_j^{(\ell)}(r). \tag{2}$$

Here $r^\top U^{(\ell)}$ are the coordinates of r in the logical subspace, and $E^{(\ell)}(r)$ is the fraction of its ℓ_2 norm explained by that subspace. Further details are given in Appendix K.

3 Method

Our goal is to exploit the paired natural-language and symbolic proofs from Section 2.2 to (i) learn, for selected layers, a low-dimensional "multi-view logical subspace" in the residual stream and (ii) steer the model's hidden states along this subspace during reasoning generation.

3.1 Learning a multi-view logical subspace

For each layer $\ell \in \mathcal{L}_{\text{sub}}$, the construction in Section 2.2 yields pooled activation matrices $X^{(\ell)}$ and $Y^{(\ell)}$ where each row corresponds to one instance, viewed through its natural-language or symbolic proof. We then apply a PCA+CCA pipeline to learn, on the NL side, a low-dimensional subspace that is maximally aligned with the symbolic view.

Step 1: PCA for denoising and compression. We first apply Principal Component Analysis (PCA) separately to $X^{(\ell)}$ and $Y^{(\ell)}$, retaining the smallest number of components that explain a fixed fraction of variance (98%) and obtaining reduced representations

$$\tilde{X}^{(\ell)} \in \mathbb{R}^{N\times d_X}, \qquad \tilde{Y}^{(\ell)} \in \mathbb{R}^{N\times d_Y},$$

with $d_X, d_Y \ll D$. We column-center $\tilde{X}^{(\ell)}$ and $\tilde{Y}^{(\ell)}$ before running CCA.

Step 2: Canonical Correlation Analysis. We then run linear Canonical Correlation Analysis (CCA) (HOTELLING, 1936; Raghu et al., 2017) on $\tilde{X}^{(\ell)}$ and $\tilde{Y}^{(\ell)}$, for a fixed number k of canonical components ($k = 32$ in the main setting; see Appendix J.1 for a robustness study over k). CCA finds directions $\{a_j^{(\ell)}\}_{j=1}^k$ in the NL space and $\{b_j^{(\ell)}\}_{j=1}^k$ in the symbolic space such that the scalar projections $\tilde{X}^{(\ell)}a_j^{(\ell)}$ and $\tilde{Y}^{(\ell)}b_j^{(\ell)}$ have maximal Pearson correlation.

Stacking these directions gives projection matrices $A^{(\ell)} \in \mathbb{R}^{d_X\times k}$ and $B^{(\ell)} \in \mathbb{R}^{d_Y\times k}$.

Step 3: Back-projection and orthonormal basis. Let $V_X^{(\ell)} \in \mathbb{R}^{D \times d_X}$ be the PCA loading matrix for $X^{(\ell)}$. We map the NL-side canonical directions back into the original residual space via

$$W^{(\ell)} = V_X^{(\ell)} A^{(\ell)} \in \mathbb{R}^{D \times k},$$

whose columns span k directions in the layer-ℓ residual stream. To obtain an orthonormal basis, we apply a QR decomposition

$$W^{(\ell)} = Q^{(\ell)} R^{(\ell)},$$

We set $U^{(\ell)} = Q^{(\ell)} \in \mathbb{R}^{D \times k}$, which forms an orthonormal basis for the subspace, and define the corresponding projector $P^{(\ell)} = U^{(\ell)} {U^{(\ell)}}^\top$.

3.2 Inference-time steering along the subspace

Given the learned projectors $\{P^{(\ell)}\}$, we modify the forward pass during Chain-of-Thought (CoT) generation without changing any model parameters.

Steering protocol. We follow a standard CoT prompting setup (Wei et al., 2022; Kojima et al., 2023): given an input, we prepend a CoT-style instruction and let the model generate a natural-language proof and answer, intervening only on the residual stream.

Let $h_t^{(\ell)} \in \mathbb{R}^D$ denote the residual vector at layer ℓ and time step t in the forward pass. For a chosen steering layer $\ell^\star$ and a scalar steering strength λ, we replace $h_t^{(\ell^\star)}$ by

$$\tilde{h}_t^{(\ell^\star)} = h_t^{(\ell^\star)} + \lambda \frac{P^{(\ell^\star)} h_t^{(\ell^\star)}}{\left\| P^{(\ell^\star)} h_t^{(\ell^\star)} \right\|_2} \left\| h_t^{(\ell^\star)} \right\|_2. \quad (3)$$

In other words, we add a perturbation in the direction of the projection onto the multi-view logical subspace with magnitude $\lambda \| h_t^{(\ell^\star)} \|_2$.

In practice, we normalize $P^{(\ell^\star)} h_t^{(\ell^\star)}$ with an ε added to the denominator for numerical stability.

For each model–benchmark pair we use a single steering layer $\ell^\star$ and a single λ, both selected on a held-out development set (see Appendix F for the hyperparameter selection procedure), and apply Eq. (3) only at $\ell^\star$ for tokens in the generated CoT; the encoding of the input context and question uses the original forward pass. This yields a training-free inference-time method that requires only a one-off subspace estimation on gold proofs and a light matrix–vector multiplication per token at the steering layer.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate on three logical reasoning benchmarks: **FOLIO** (Han et al., 2024), a logical entailment dataset with aligned natural-language stories and first-order logic (FOL) formalizations; **PrOntoQA** (Saparov and He, 2023), a synthetic multi-hop ontology QA benchmark; and **ProofWriter** (Tafjord et al., 2021), which provides multi-hop entailment questions paired with natural-language proofs under both open-world (OWA) and closed-world (CWA) semantics. On PrOntoQA and ProofWriter, we learn multi-view subspaces from the paired natural-language and symbolic proofs; on FOLIO, which lacks gold proofs, we instead use the aligned natural-language story and first-order logic (FOL) formalization as the two-view reasoning data. For all benchmarks, we use splits derived from the released data or official generation code, reserving a small development subset for hyperparameter tuning; the exact splits and construction details are given in Appendix A.

Models. We evaluate our method on five open-weight, decoder-only LLMs: Meta-Llama-3.1-8B-Instruct, Llama-3.2-3B-Instruct (Grattafiori et al., 2024), Llama-2-13B-Chat (Touvron et al., 2023), Gemma-2-9B-IT (Team et al., 2024), and Phi-3-Mini-4K-Instruct (Abdin et al., 2024). We use publicly released checkpoints and tokenizers without any additional fine-tuning, and estimate multi-view logical subspaces separately for each model–benchmark pair. Full model and decoding details are provided in Appendix C.

Baselines. For each model–benchmark pair we compare with the following decoding variants:

- **Greedy-CoT**: zero-shot CoT prompting (Wei et al., 2022) with greedy decoding and no activation intervention.
- **3-shot-CoT**: few-shot CoT prompting (Brown et al., 2020; Wei et al., 2022) with three in-context exemplars (Appendix A).
- **SC-3** (self-consistency): CoT prompting with three sampled reasoning paths per instance, following Wang et al. (2023); implementation details are given in Appendix F.

Experiment Setting. Our proposed **LSS-CoT** method applies single-layer steering along the

Model	Setting	Accuracy (%)			
		FOLIO	PrOntoQA (5-hop)	PW-CWA (3-hop)	PW-OWA (3-hop)
Llama-3.1-8B Instruct	Greedy-CoT	51.7	70.6	51.4	50.7
	3-shot-CoT	60.6	72.4	**66.4**	**63.7**
	SC-3	58.1	**79.0**	47.2	53.3
	LSS-CoT	**61.1**	75.4	55.6	55.3
	Δ (LSS–Greedy)	↑ 9.4	↑ 4.8	↑ 4.2	↑ 4.6
LLaMA-3.2-3B Instruct	Greedy-CoT	51.2	50.0	46.6	37.1
	3-shot-CoT	42.4	**61.4**	**56.8**	**51.5**
	SC-3	49.3	49.4	46.8	37.1
	LSS-CoT	**53.7**	53.4	49.8	38.7
	Δ (LSS–Greedy)	↑ 2.5	↑ 3.4	↑ 3.2	↑ 1.6
LLaMA-2-13B Chat	Greedy-CoT	42.9	48.6	54.6	39.1
	3-shot-CoT	**51.2**	51.2	**58.0**	**42.9**
	SC-3	46.8	**55.4**	52.6	42.5
	LSS-CoT	45.8	55.0	56.6	**42.9**
	Δ (LSS–Greedy)	↑ 2.9	↑ 6.4	↑ 2.0	↑ 3.8
Gemma-2-9B It	Greedy-CoT	58.1	87.4	71.4	75.0
	3-shot-CoT	**68.0**	82.8	72.8	71.9
	SC-3	63.1	90.0	72.8	75.6
	LSS-CoT	65.5	**90.2**	**73.8**	**76.6**
	Δ (LSS–Greedy)	↑ 7.4	↑ 2.8	↑ 2.4	↑ 1.6
Phi-3-mini-4k Instruct	Greedy-CoT	62.6	59.6	59.6	51.1
	3-shot-CoT	**68.0**	63.8	**69.2**	**73.9**
	SC-3	67.0	70.2	66.4	59.3
	LSS-CoT	66.0	**70.6**	63.4	56.1
	Δ (LSS–Greedy)	↑ 3.4	↑ 11.0	↑ 3.8	↑ 5.0

Table 1: Main results on four logical reasoning benchmarks. Greedy-CoT, LSS-CoT, SC-3, and 3-shot-CoT are defined in Section 4. For each model, the last row shows LSS–Greedy absolute improvement. Best score per model–benchmark pair is in **bold**, second-best is underlined.

learned multi-view logical subspace during CoT generation (Section 3.2; tuning details in Appendix F). Across all benchmarks we use a single task-specific CoT prompt per task (Appendix D) and, unless otherwise noted, greedy decoding with a budget of 1024 new tokens, treating cases with no answer within this budget as incorrect. Inference overhead is negligible in practice: on our PrOntoQA setup with Llama-3.1-8B-Instruct, throughput is 179 tok/s without steering and 176 tok/s with steering under the same experimental setting.

4.2 Main Results

Table 1 presents our main results across benchmarks and models. Overall, steering along the multi-view logical subspace consistently improves Chain-of-Thought (CoT) accuracy over the Greedy-CoT baseline and often matches or surpasses much more expensive self-consistency decoding.

Gains over Greedy CoT. Across all model–benchmark pairs, **LSS-CoT** improves over **Greedy-CoT** by between 1.6 and 11 absolute accuracy points. The largest gains appear on PrOntoQA for Phi-3-Mini (+11.0) and on FOLIO for Llama-3.1-8B (+9.4), where the base Greedy-CoT performance leaves substantial headroom. Even on stronger configurations such as Gemma-2-9B on PrOntoQA and ProofWriter, steering still yields non-trivial improvements (roughly +2–+3 points), showing that the learned multi-view subspace adds value beyond what pretrained models already achieve with standard CoT prompting. LSS also remains effective on a reasoning-specialized model. On Qwen3-4B evaluated on PrOntoQA, accuracy improves from 87.2 without steering to 93.2 with LSS (+6.0), indicating that the learned multi-view subspace remains beneficial even when the base model already exhibits strong reasoning performance (Appendix F.4).

Comparison to SC-3. Self-consistency with $k = 3$ (**SC-3**) provides a strong test-time scaling baseline, but incurs a 3× increase in inference cost. On several settings, **LSS-CoT** with a single decoding trajectory matches or even outperforms SC-3. For example, on FOLIO with Llama-3.1-8B, our method improves accuracy from 51.7 to 61.1, whereas SC-3 reaches only 58.1. On PrOntoQA, Gemma-2-9B and Phi-3-Mini achieve comparable or slightly higher accuracy under LSS-CoT than under SC-3, despite using only one CoT sampling process instead of three. These results suggest that activation-level steering can recover much of the benefit of test-time ensembling at a fraction of the compute cost.

Method	Accuracy (%)
Greedy-CoT	70.6
3-shot-CoT	72.4
3-shot-CoT + LSS	74.6
SC-3	79.0
SC-3 + LSS	**81.0**

Table 2: Combining LSS with inference schemes (few-shot CoT and self-consistency) on PrOntoQA (5-hop) using Llama-3.1-8B-Instruct.

On smaller models such as Llama-3.2-3B, SC-3 can even degrade performance (e.g., on FOLIO and PrOntoQA) relative to Greedy-CoT. One possible explanation is that its sampled reasoning paths exhibit high variance and introduce additional noise.

In contrast, LSS-CoT consistently delivers modest but reliable improvements on the same model, indicating that manipulating internal activations can serve as a **stabilizer** for weaker architectures: rather than adding more samples, it nudges hidden states toward reasoning-relevant configurations in a controlled way.

Comparison to few-shot CoT. We also compare against **3-shot-CoT**, which augments the prompt with three in-context examples. On FOLIO, LSS-CoT *outperforms or matches* 3-shot-CoT on the LLaMA-3 models while using no in-context examples and thus incurring zero additional prompt length. For example, on Llama-3.1-8B and Llama-3.2-3B, LSS-CoT achieves higher accuracy than 3-shot-CoT despite operating in a purely zero-shot setting, whereas 3-shot-CoT can even degrade performance on the smaller Llama-3.2-3B. This highlights a complementary form of *context efficiency*: rather than relying on longer prompts or manually curated exemplars, our method improves reasoning purely via a lightweight intervention on the residual stream.

Taken together, these results show that steering along the learned multi-view logical subspace yields consistent gains over Greedy-CoT, is competitive with self-consistency at far lower compute cost, and can outperform few-shot CoT without using additional context tokens.

More broadly, they point to a third, complementary route for enhancing reasoning in LLMs: instead of scaling context length or sampling budget, we can directly *align internal representations* via activation-level steering, effectively unlocking latent logical capabilities that are underutilized by standard decoding.

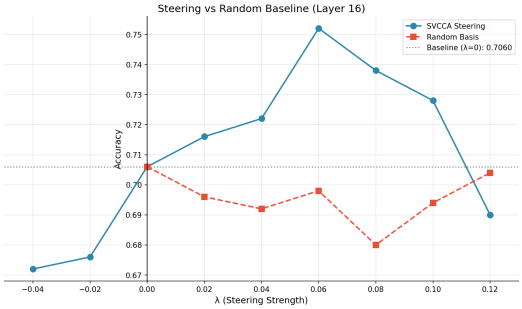


Figure 3: Sensitivity to steering strength. Accuracy on PrOntoQA (Llama-3.1-8B) as a function of steering coefficient λ for our logical subspace direction (blue) vs. random orthogonal directions (red).

4.3 Compatibility with Inference Schemes

Beyond comparing LSS-CoT against the decoding baselines such as few-shot CoT and self consistency, we also test whether the same learned logical subspace can be stacked on top of them. On PrOntoQA (5-hop) with Llama-3.1-8B-Instruct, we reuse the exact same subspace, steering layer, and strength λ selected for LSS-CoT in the main experiment, with no additional retuning. As shown in Table 2, adding LSS on top of 3-shot CoT improves accuracy from 72.4 to 74.6 (+2.2), and adding LSS on top of SC-3 improves accuracy from 79.0 to 81.0 (+2.0), with SC-3+LSS performing best overall. These results show that LSS is not only a lightweight alternative to more expensive inference schemes, but also a complementary intervention that can further improve them.

4.4 Sensitivity to the Steering Direction

We study how performance varies with the steering coefficient λ and direction on PrOntoQA with Llama-3.1-8B. At the chosen layer, we compare steering along our logical subspace direction to steering along random orthogonal directions in the same residual space (Fig. 3; further details are given in Appendix J).

Steering along random directions fails to yield systematic gains and often degrades accuracy across λ. By contrast, steering along the logical subspace direction shows a clear asymmetry: moderate positive λ improves performance, while negative λ causes sharp drops. This pattern supports the view that the extracted direction is *task-positive*, rather than a generic rescaling of activations.

Model	Setting	Accuracy (%)		
		Greedy	LSS	Δ
Llama-3.1-8B	NLI	51.8	**56.2**	↑ 4.4
	MCR	48.8	**51.8**	↑ 3.0
Phi-3-Mini	NLI	52.6	**56.2**	↑ 3.6
	MCR	51.8	**54.4**	↑ 2.6

Table 3: Cross-dataset generalization on LogiQA2.0

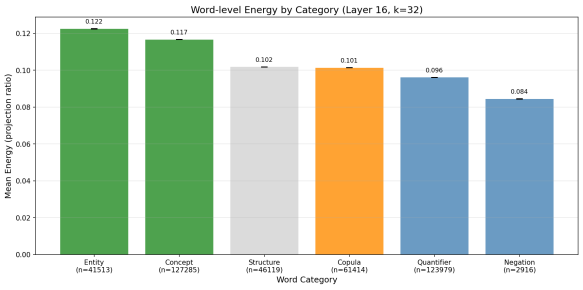


Figure 4: **Global Energy Distribution.** Average projection energy by category (Llama-3.1-8B, Layer 16).

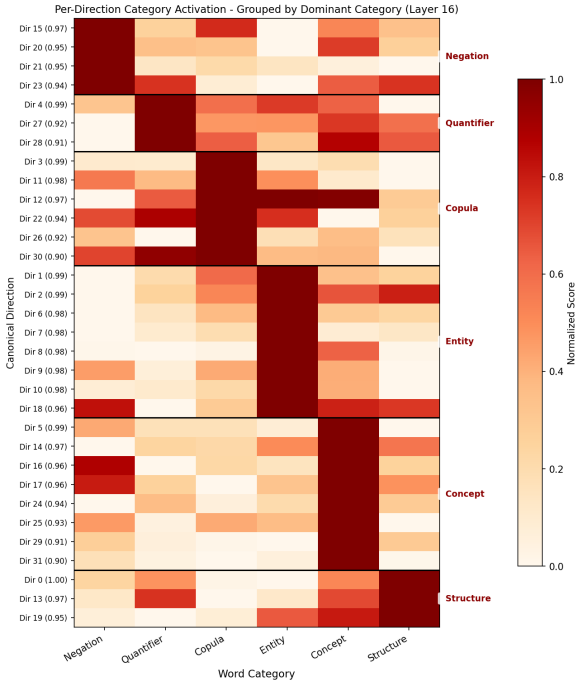


Figure 5: **Heatmap of directional selectivity.** Normalized activations for Llama-3.1-8B (Layer 16), with directions **sorted by dominant category**.

4.5 Generalization

We test whether the multi-view logical subspace captures abstract reasoning patterns that transfer beyond the synthetic data it was learned on.

Concretely, we keep the PrOntoQA-trained subspace $P^{(\ell)}$ fixed (Section 5.1) and use it to steer models on the LogiQA 2.0 dataset (Liu et al., 2023a), evaluating both its natural language inference (NLI) and multiple-choice reading (MCR) formats. The subspace is learned only from synthetic rule-based PrOntoQA proofs and will not be updated on LogiQA; full setup details, including sample sizes, are given in Appendix B.

Table 3 shows that this PrOntoQA-derived subspace transfers effectively to LogiQA 2.0: steering yields consistent accuracy gains across models and both evaluation formats (e.g., **+3.0** points on Llama-3.1-8B MCR). We observe a similar transfer effect on ReClor dataset (Yu et al., 2020), where reusing the same fixed PrOntoQA-derived subspace improves Llama-3.1-8B from 53.4 to 56.6 (+3.2), without retraining the subspace.

Despite the domain shift from clean synthetic proofs to noisy exam-style passages and multiple-choice logical reading comprehension, these improvements suggest that the multi-view subspace encodes general logical mechanisms, such as rule following and entailment tracking, rather than overfitting to the surface form of PrOntoQA.

We further conduct a non-logic sanity check, by evaluating the effect of steering on a non-logic commonsense reasoning dataset HellaSwag (Zellers et al., 2019). Specifically, we steer Llama-3.1-8B using the same PrOntoQA-derived subspace, layer, and λ as in the main experiment. Accuracy changes only from 0.652 to 0.650, indicating negligible degradation on the non-logic reasoning benchmark.

5 Analysis

We next analyze the learned multi-view logical subspaces and the effect of steering on model behavior. Our goal is to better understand (i) what these logical subspaces encode, (ii) how NL-symbolic alignment evolves within model, (iii) how the logical subspace relate to reasoning success, and (iv) how steering reshapes the structure of CoT generation.

5.1 What does the multi-view subspace encode?

To understand what the learned multi-view logical subspaces capture, we use the token-level projection energy $E^{(\ell)}(r)$ defined in Eq. 1, which measures how much of a token's representation lies inside the logical subspace, and its per-direction decomposition (Eq. 2). We apply this analysis to a synthetic rule-based corpus generated with the official PrOntoQA code; details in Appendix F.2.

At a global level, we ask which token categories

carry most of the subspace energy. We aggregate $E^{(\ell)}(r)$ over tokens of each category within a proof and then average across instances. Across both Llama-3.1-8B and Phi-3-Mini, entity and concept tokens have the highest mean energy, followed by structure and copula tokens, while negations and quantifiers carry lower but still clearly non-zero energy. In other words, the shared subspace allocates most of its mass to "who" the proof is about and "what" properties or relations hold, while still systematically responding to logical operators rather than treating them as noise. Similar patterns appear across several middle and upper layers, suggesting that this focus on core predicate content is a stable property of the subspace (see Fig. 4 and Appendix K).

At a finer-grained level, we ask whether individual canonical directions $u_j^{(\ell)}$ correspond to specific logical roles. We compute their average normalized contribution to each word category and visualize the resulting matrix. Directions cluster into interpretable groups: some are dominated by entities and concepts, others by quantifiers or negation, and others by structural markers such as connectives and conclusion phrases (Fig. 5). This "role-wise" organization is consistent between layers and models, indicating that the multi-view logical subspace is not an arbitrary slice of the residual stream but a structured space whose basis directions specialize in different aspects of the proof. Additional visualizations are provided in Appendix F.3.

5.2 How does the alignment between language and logic evolve across layers?

We next ask how strongly the natural-language and symbolic views are aligned across layers. Following prior work on representation similarity (Raghu et al., 2017), we compute, for each layer ℓ, the mean canonical correlation $\bar{\rho}^{(\ell)}$ between the pooled NL and symbolic activations on PrOntoQA (details in Appendix E). Intuitively, $\bar{\rho}^{(\ell)}$ ranges from 0 (no shared structure) to 1 (perfect alignment).

Figure 6 shows the resulting trajectory for Llama-3.1-8B. Across all evaluated models we observe a robust **"high–low–high"** pattern: alignment is high in the early layers, drops in the middle layers, and rises again in upper layers (full curves in Appendix G). This suggests that the logical subspace is *dynamic*: alignment is weakened in the middle layers, where the model appears to encode information into more task-specific representations, and then strengthened again near the top.

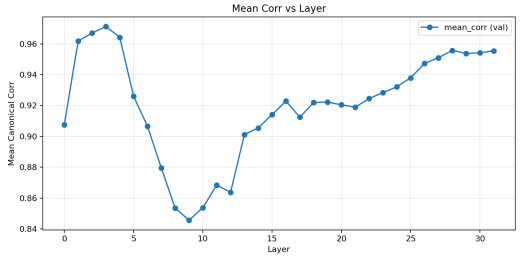


Figure 6: Layer-wise mean canonical correlation between language and logic (Llama-3.1-8B).

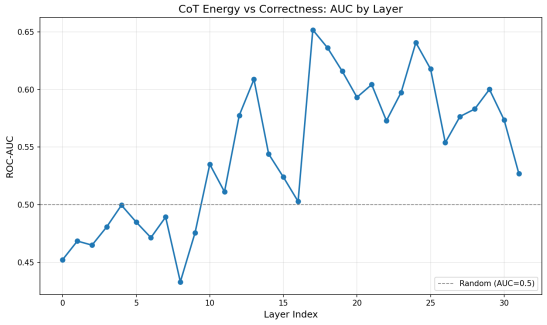


Figure 7: Layer-wise ROC-AUC of projection-energy-based correctness prediction on PrOntoQA for Llama-3.1-8B.

5.3 Can the logical subspace tell when a reasoning chain is correct?

We now ask whether activations that lie more strongly in the logical subspace are also associated with correct CoTs. For Llama-3.1-8B on PrOntoQA, we reuse the layer-wise projectors $P^{(\ell)}$ from the steering setup and score Greedy-CoT traces, for each layer, by their mean projection energy over CoT tokens (see Appendix E for details). We treat this mean energy as a scalar score for each chain and, for each layer, compute the ROC-AUC of this score for distinguishing correct from incorrect answers (Fig. 7). An AUC of 0.5 corresponds to random guessing and 1.0 to perfect discrimination.

Across most layers, projection energy yields AUCs clearly above 0.5, indicating that correct reasoning paths tend to align more strongly with the learned logical subspace. The signal is strongest in middle-to-late layers (peaking around AUC $\approx$ 0.65), consistent with the alignment dynamics in Section 5.2: after the mid-layer "decoupling" phase, deeper layers again bring language and logic into closer alignment, so projection energy provides a non-trivial signal of CoT correctness.

Method	Reasoning chain
Greedy-CoT	*(After reaching the key fact, continues with irrelevant checks.)* "we cannot conclude anything about Polly" "we cannot conclude anything" *(repeats for many more steps)* ⇒ **wrong label**.
LSS-CoT	*(Goal-directed, early stop.)* "Since Polly is a shumpus and shumpuses are not snowy, therefore Polly is not snowy." ⇒ **correct label**.

Table 4: PrOntoQA case study (abridged).

5.4 How does steering change the model's reasoning behavior?

We analyze how steering changes CoT behavior on PrOntoQA for Llama-3.1-8B, using the same setup as in Section 4 and computing statistics over the CoT span only (details in Appendix E).

Steering noticeably reshapes both length and style. CoTs that are already correct under Greedy-CoT become slightly longer (on average **+2.11** steps), whereas those incorrect ones become much shorter (on average -9.83 steps), suggesting that steering prunes unproductive loops. Lexical counts of key reasoning markers (Appendix I) show a similar shift: "reasoning verbs" such as *think*, *know*, and *assume* decrease by **16.7%**, while logical connectives increase by **4.8%**, with *since* and *so* showing the largest gains (**+7.0%** and **+18.3%**). Overall, LSS-CoT uses fewer conversational fillers and more explicit causal links (e.g., "since A, so B"), yielding shorter and more deductive chains.

These aggregate trends are mirrored in individual examples. Table 4 shows an abridged PrOntoQA instance where Greedy-CoT enters an overlong self-checking loop and predicts an incorrect label, whereas LSS-CoT follows a short causal chain and stops early with the correct answer.

6 Related Work

Chain-of-Thought (CoT) prompting improves multi-step reasoning in large language models by eliciting intermediate reasoning steps rather than direct answers. Subsequent work develops stronger test-time inference schemes, such as self-consistency and other compute-scaled decoding strategies, to further improve reasoning robustness (Wei et al., 2022; Kojima et al., 2023; Wang et al., 2023; Long, 2023; Zhou et al., 2023).

A related line of work seeks to combine natural-language and symbolic reasoning. One class of methods does so at inference time by translating natural-language inputs or intermediate reasoning into symbolic forms and invoking external provers, solvers, or verifiers (Olausson et al., 2023; Pan et al., 2023; Pei et al., 2025).

Another class uses training or finetuning with natural-language and symbolic supervision so that the model internalizes stronger alignment between the two views during parameter learning (Zhou et al., 2025; Tan et al., 2025). These approaches improve reasoning either through external tool use or through additional supervision.

Activation steering provides a complementary way to improve model behavior by intervening directly in internal representations at inference time, often using directions learned from contrastive activations, representation engineering, or sparse features (Turner et al., 2024; Rimsky et al., 2024; Li et al., 2025; Galichin et al., 2025).

Our work offers a new perspective on combining natural-language and symbolic reasoning. Instead of relying on external symbolic tools at inference time or using additional supervision to internalize the alignment during training, we ask whether a shared logical subspace aligning the two views is already internalized in a LLM. We then operationalize this perspective through a training-free activation intervention: using paired natural-language and symbolic proofs, we estimate a shared multi-view logical subspace and steer CoT generation along it without changing model weights or relying on external symbolic solvers.

7 Conclusion

We introduce logical subspace steering, which uses paired natural-language and symbolic proofs to estimate a multi-view logical subspace in a frozen LLM and steer Chain-of-Thought reasoning at inference time. Across multiple logical reasoning benchmarks, this LSS-CoT outperforms greedy CoT and recovers much of the benefit of more expensive self-consistency and few-shot prompting without changing model weights or relying on external provers. Analysis shows that steering yields shorter, more causal reasoning chains and interpretable logical structure, giving a representation-level handle on the model's logical reasoning.

Limitations

Our work focuses on logical reasoning, and the findings may not directly generalize to broader settings such as open-domain dialogue, mathematical

problem solving, long-horizon agentic tasks, multilingual applications, or extremely large proprietary or specialized theorem-proving models. Evaluating whether similar multi-view logical subspaces arise, and remain amenable to steering, in these broader domains, architectures, and training regimes is an important direction for future work. In addition, we primarily evaluate reasoning quality through final-answer accuracy. Developing more fine-grained evaluations of intermediate reasoning steps, including their correctness and faithfulness, is a valuable avenue for future research.

Ethics Statement

This work studies representation-level methods for improving logical reasoning in open-weight language models by steering internal activations along a learned multi-view logical subspace. Our experiments are conducted on publicly available benchmarks (FOLIO, PrOntoQA, ProofWriter, and LogiQA 2.0) and open-weight models within the 3B–13B parameter range, and do not involve private user data or human subjects. We release code and data with the expectation that they will be used responsibly for research and social good.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

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, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Fengxiang Cheng, Haoxuan Li, Fenrong Liu, Robert van Rooij, Kun Zhang, and Zhouchen Lin. 2025. Empowering llms with logical reasoning: A comprehensive survey. *arXiv preprint arXiv:2502.15652*.

Andrey Galichin, Alexey Dontsov, Polina Druzhinina, Anton Razzhigaev, Oleg Y. Rogov, Elena Tutubalina, and Ivan Oseledets. 2025. I have covered all the bases here: Interpreting reasoning features in large language models via sparse autoencoders. *Preprint*, arXiv:2503.18878.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alex Wardle-Solano, Hannah Szabo, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. Folio: Natural language reasoning with first-order logic. *Preprint*, arXiv:2209.00840.

HAROLD HOTELLING. 1936. Relations between two sets of variates*. *Biometrika*, 28(3-4):321–377.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Yuanyuan Lei and Ruihong Huang. 2024. Boosting logical fallacy reasoning in LLMs via logical structure tree. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13157–13173, Miami, Florida, USA. Association for Computational Linguistics.

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. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10904–10924, Suzhou, China. Association for Computational Linguistics.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023a. Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023b. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*.

Jieyi Long. 2023. Large language model guided tree-of-thought. *Preprint*, arXiv:2305.08291.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176, Singapore. Association for Computational Linguistics.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

Yu Pei, Yongping Du, and Xingnan Jin. 2025. FoVer: First-order logic verification for natural language reasoning. *Transactions of the Association for Computational Linguistics*, 13:1340–1359.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Preprint*, arXiv:1706.05806.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *Preprint*, arXiv:2210.01240.

Geoff Sutcliffe and Christian Suttner. 1998. The tptp problem library. *J. Autom. Reason.*, 21(2):177–203.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.

Xingwei Tan, Marco Valentino, Mahmud Elahi Akhter, Maria Liakata, and Nikolaos Aletras. 2025. Enhancing logical reasoning in language models via symbolically-guided Monte Carlo process supervision. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 31886–31900, Suzhou, China. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

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 Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *Preprint*, arXiv:2308.10248.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2025. Are large language models really good logical reasoners? a comprehensive evaluation and beyond. *IEEE Transactions on Knowledge and Data Engineering*, 37(4):1620–1634.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations (ICLR)*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. *Preprint*, arXiv:2205.10625.

Yujun Zhou, Jiayi Ye, Zipeng Ling, Yufei Han, Yue Huang, Haomin Zhuang, Zhenwen Liang, Kehan Guo, Taicheng Guo, Xiangqi Wang, and Xiangliang Zhang. 2025. Dissecting logical reasoning in LLMs: A fine-grained evaluation and supervision study. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17075–17098, Suzhou, China. Association for Computational Linguistics.

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, and 2 others. 2025a. Representation engineering: A top-down approach to ai transparency. *Preprint*, arXiv:2310.01405.

Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. 2025b. Reasonflux-prm: Trajectory-aware prms for long

chain-of-thought reasoning in llms. *Preprint*, arXiv:2506.18896.

A Dataset Construction and Splits

A.1 FOLIO

FOLIO (Han et al., 2024) provides 1,001 training instances and 203 development instances, but no public test set. Each instance contains natural-language premises, a corresponding first-order logic (FOL) formalization, a hypothesis, and a three-way label (TRUE, FALSE, UNCERTAIN).

Since FOLIO does not include gold proofs, we treat the premises plus hypothesis as the core reasoning context, and use the ground-truth label to form complete inputs for both views when constructing multi-view activations. Concretely, in the natural-language view we concatenate the premises, hypothesis, and a final sentence of the form *"The hypothesis is [label]."*; in the symbolic view we concatenate the FOL formalization, the hypothesis in FOL, and the same label statement in natural language. These two inputs describe the same entailment judgement in different forms and are fed to the frozen model in a teacher-forcing pass to collect residual activations (Section 2.2). Note that this label information is used only for subspace estimation and is never available at test time when generating CoTs or evaluating accuracy.

For our experiments, we use the released development set as the test set. From the 1,001 training instances, we randomly sample 50 examples as a held-out development set for hyperparameter tuning and use the remaining 951 examples to estimate the multi-view logical subspaces. All random splits are controlled by a fixed seed for reproducibility.

FOLIO example instance

For illustration, we show a toy FOLIO-style instance and how we form the two views used for activation extraction.

Raw fields (toy example).

- Premises (NL):
  All birds are animals. Tweety is a bird.
- Premises (FOL):
  $\forall x\,(\mathrm{Bird}(x) \rightarrow \mathrm{Animal}(x))$
  $\mathrm{Bird}(\mathrm{tweety})$
- Conclusion (NL):
  Tweety is an animal.
- Conclusion (FOL):
  $\mathrm{Animal}(\mathrm{tweety})$
- Label: TRUE

NL-view input. *Premises:*
All birds are animals. Tweety is a bird.

Hypothesis:
Tweety is an animal.

The Hypothesis is True.

Symbolic-view input. *Premises (FOL):*
$\forall x\,(\mathrm{Bird}(x) \rightarrow \mathrm{Animal}(x))$
$\mathrm{Bird}(\mathrm{tweety})$

Hypothesis (FOL):
$\mathrm{Animal}(\mathrm{tweety})$

The Hypothesis is True.

A.2 PrOntoQA

For PrOntoQA (Saparov and He, 2023), we follow the FICTIONAL-5HOP setting and use the official synthetic data generation code released by the authors. We generate 2,000 five-hop entailment instances with the default parameters.

Each generated instance includes the same natural-language premises (describing a fictional ontology) and query, paired with (i) a natural-language chain-of-thought reasoning trace and (ii) a sequence of first-order logic proof steps produced by the generator. We treat these as the natural-language and symbolic proof views, respectively.

We randomly split the 2,000 instances into 1,000 training, 500 development, and 500 test examples. The 1,000 training instances are used to estimate the multi-view logical subspaces, the 500 development instances are used for hyperparameter tuning (e.g., steering layer and λ), and the 500 test instances are used solely for evaluation.

Example. Below we show a toy PrOntoQA-style instance with its two views: a natural-language proof-style explanation and a corresponding symbolic logic proof trace.

Natural-language (NL) proof view

```
Premises:
All flurps are red.
All red things are warm.
Mip is a flurp.

True or false: Mip is warm.

Mip is a flurp.
```

```
All flurps are red.
So Mip is red.
All red things are warm.
So Mip is warm.

The query is True.
```

Symbolic proof view

```
Premises:
All flurps are red.
All red things are warm.
Mip is a flurp.

True or false: Mip is warm.

flurp(Mip)
![X1]:((flurp(X1) => red(X1)))
red(Mip)
![X1]:((red(X1) => warm(X1)))
warm(Mip)

The query is True.
```

A.3 ProofWriter (OWA and CWA)

ProofWriter (Tafjord et al., 2021) provides multi-hop entailment questions under both open-world (OWA) and closed-world (CWA) semantics, together with gold natural-language proofs.[2]

In this work we use ProofWriter in two ways. First, for learning multi-view subspaces we focus on the 5-hop regime. For the OWA setting, we sample 2,000 5-hop training instances from the released data and a fixed random seed. Each such instance includes (i) a natural-language proof explaining the entailment decision. We translate each gold natural-language proof into a TPTP-style (Sutcliffe and Suttner, 1998) symbolic proof using a GPT-5 few-shot prompt and treat the resulting pair as the natural-language and symbolic views for subspace estimation.

For subspace learning, we split the 2,000 translated 5-hop instances into 1,500 training and 500 development examples. Training examples are used to estimate the multi-view logical subspaces, and the remaining 500 5-hop examples are reserved for qualitative and diagnostic analysis only (see Section 5); they are never used for hyperparameter selection or test-time evaluation.

For steering hyperparameters, we operate in the same 3-hop regime as used at test time. From the official 3-hop validation splits, we subsample 500 OWA and 501 CWA instances with approximately balanced labels and use them solely to select the steering layer and λ.

For final evaluation, we focus on the 3-hop subset of ProofWriter, which offers a stable and interpretable regime for assessing CoT-based reasoning. From the official 3-hop test splits, we subsample another 500 OWA and 501 CWA instances (again balanced in labels), and these held-out 3-hop instances are used exclusively for reporting final accuracy in our main experiments. OWA and CWA are reported as separate evaluation settings in all result tables.

Example (ProofWriter: NL vs. symbolic view). Below we show a toy ProofWriter-style instance with its two views: a natural-language proof-style explanation and a corresponding first-order logic proof trace.

Natural-language (NL) proof view

```
Facts: Alice is young. All young people
are students. All students are diligent.

Query: Alice is diligent.

Proof: Alice is young. All young people
are students.  Therefore Alice is a
student.  Alice is a student.  All
students are diligent. Therefore Alice
is diligent.

The query is True.
```

Symbolic proof view

```
Facts:  young(Alice).  ![X]:((young(X)
=> student(X))). ![X]:((student(X) =>
diligent(X))).

Query: diligent(Alice).

Proof:  young(Alice).  ![X]:((young(X)
=>    student(X))).    student(Alice).
student(Alice).  ![X]:((student(X) =>
diligent(X))). diligent(Alice).

The query is True.
```

A.4 Few-Shot Examples

We use 3-shot in-context learning with one example for each label (True, False, Uncertain). Below we show one representative example.

[2] Under OWA, facts not stated in the context may be unknown; under CWA, unstated facts are treated as false.

Few-Shot Example

Premises:
All eels are fish.
No fish are plants.
Everything displayed in the collection is either a plant or an animal.
All multicellular animals are not bacteria.
All animals displayed in the collection are multicellular.
A sea eel is displayed in the collection.
The sea eel is an eel or an animal or not a plant.
Hypothesis:
The sea eel is bacteria.
Reasoning:
Let's analyze step by step:
1. From premise 6: The sea eel is displayed in the collection.
2. From premise 3: Since it's displayed, the sea eel is either a plant or an animal.
3. From premise 7: The sea eel is an eel or an animal or not a plant.
4. From premise 1: All eels are fish.
5. From premise 2: No fish are plants. So if the sea eel is an eel, it's a fish, thus not a plant.
6. From premise 3: Since it's displayed and not a plant, the sea eel must be an animal.
7. From premise 5: All animals displayed in the collection are multicellular.
8. So the sea eel is a multicellular animal.
9. From premise 4: All multicellular animals are not bacteria.
10. Therefore, the sea eel is NOT bacteria.
Truth value: False

B Generalization Setup

This appendix provides additional details for the cross-dataset generalization experiments from PrOntoQA to LogiQA 2.0 reported in Section 4.5. We use an analogous transfer protocol for ReClor, reusing the same fixed PrOntoQA-derived subspace without retraining.

B.1 Source of the Steering Subspace

For all LogiQA 2.0 results, the multi-view logical subspaces are estimated *exclusively* from synthetic PrOntoQA data. Specifically, we use the official PrOntoQA code to generate 5,000 fictional 5-hop entailment instances in the FICTIONAL-COMPOSED-RULE setting and run the PCA+CCA pipeline from Section 3.1 on the pooled residual activations of paired natural-language and symbolic proofs. For each model and each layer ℓ, this yields an orthonormal NL-side basis $U^{(\ell)}$ and the corresponding projector $P^{(\ell)} = U^{(\ell)}U^{(\ell)\top}$.

No LogiQA 2.0 examples are used when fitting PCA, running CCA, or selecting the subspace dimension k. As in our in-domain experiments, we restrict steering to a single layer $\ell^\star$ per model, chosen based on the mean canonical correlation criterion described in Appendix F.1. The choice of $\ell^\star$ is fixed before any LogiQA 2.0 evaluation.

B.2 LogiQA 2.0 Tasks and Splits

LogiQA 2.0 is a logical reading comprehension benchmark derived from Chinese civil service examinations and provides two evaluation settings that we consider:

- **MCR (multiple-choice reading)**: given a passage, a question, and four options (A–D), the model must select the correct answer;
- **NLI (natural language inference)**: given a set of premises and a conclusion, the model must decide whether the conclusion is logically supported.

For each model and each setting (MCR/NLI), we tune the steering strength λ on the official LogiQA 2.0 development split and evaluate on a subset of the official test split. Concretely, we randomly sample 500 test examples for MCR and another 500 test examples for NLI from the corresponding LogiQA 2.0 test sets and report accuracy on these held-out subsets. LogiQA 2.0 examples are therefore used only to select λ and for evaluation; they never participate in subspace estimation.

B.3 Prompting and Decoding on LogiQA 2.0

We use the same general Chain-of-Thought prompting protocol as in our main experiments (Section 4), adapted to the two LogiQA 2.0 formats. All experiments use greedy decoding (`do_sample = False`) with a maximum of 1024 new tokens; generations that fail to produce a valid final-line tag within this budget are counted as incorrect.

For the MCR setting we use the following prompt template:

LogiQA 2.0 MCR prompt

System: You are a careful logician. Use classical deductive reasoning.

User:
Passage:
{text}
Question: {question}
Options:
A. {option_a}
B. {option_b}
C. {option_c}
D. {option_d}
Instructions:

- First, reason step by step.
- Then, on the last line, output exactly: `Answer: <A|B|C|D>`

For the NLI setting we use the following template:

LogiQA 2.0 NLI prompt

System: You are a careful logician. Use classical deductive reasoning.

User:
Premises:
{premises}
Conclusion:
{conclusion}
Instructions:

- First, reason step by step about whether the conclusion follows from the premises.
- Then, on the last line, output exactly: `Answer: <Entailed|Not Entailed>`

These templates mirror the CoT prompts used for our other benchmarks (Appendix D) but are adapted to the passage–question format and label space of LogiQA 2.0.

B.4 Steering Protocol and Hyperparameters

At inference time on LogiQA 2.0, we reuse the projector $P^{(\ell^\star)}$ learned from PrOntoQA and apply the same inference-time steering rule as in Eq. (3). For each generated CoT token at time step t, we replace the residual vector $h_t^{(\ell^\star)}$ by

$$\tilde{h}_t^{(\ell^\star)} = h_t^{(\ell^\star)} + \lambda \frac{P^{(\ell^\star)} h_t^{(\ell^\star)}}{\left\| P^{(\ell^\star)} h_t^{(\ell^\star)} \right\|_2 + \varepsilon} \left\| h_t^{(\ell^\star)} \right\|_2,$$

with a small ε added for numerical stability. We intervene only on tokens in the generated CoT; the encoding of the passage, question, premises, and conclusion is left unchanged.

For each model and each LogiQA 2.0 setting (MCR/NLI), we perform a small grid search over

$$\lambda \in \{0.02, 0.04, 0.06, 0.08, 0.10, 0.12\}$$

on the development split and select the value that maximizes CoT accuracy. The selected λ is then fixed for all test examples. No PCA or CCA is ever recomputed on LogiQA 2.0; the subspace is entirely inherited from PrOntoQA.

C Models

Model	Params	Reference
Llama 3.1 8B Instruct	8B	Grattafiori et al. (2024)
Llama 3.2 3B Instruct	3B	Grattafiori et al. (2024)
Llama 2 13B Chat	13B	Touvron et al. (2023)
Gemma 2 9B IT	9B	Team et al. (2024)
Phi-3-mini-4k-instruct	3.8B	Abdin et al. (2024)

Table 5: Open-weight decoder-only models used in our experiments.

D Prompt Templates

In this appendix we list the exact prompt templates used in our experiments. For models that support a `system` role (e.g., LLaMA and Phi-3 families), we send the *System* and *User* messages below as two separate chat turns. For Gemma, which does not expose a system role, we concatenate the system content and the user content into a single user message.

D.1 FOLIO Evaluation Prompt

System message

You are a helpful reasoning assistant.

User message

You are a careful logician. Use classical deductive reasoning.
Premises:
{premises}
Hypothesis:
{hypothesis}
Instructions:

- First, reason step by step.
- Then, on the last line, output exactly: Truth value:<True|False|Uncertain>

D.2 PrOntoQA Evaluation Prompt

System Message

You are a helpful reasoning assistant.

User Message

You are a careful logician. Use classical deductive reasoning.
Premises:
{premises}
True or false: {query}
Instructions:

- First, reason step by step.
- Then, on the last line, output exactly: Truth value: <True|False>

D.3 ProofWriter (CWA) Evaluation Prompt

System Message

You are a helpful reasoning assistant.

User Message

You are a careful logician. Use classical deductive reasoning.
Facts:
{facts}
Rules:
{rules}
Query: {query}
Instructions:

- First, reason step by step. (MAXIMUM 25 steps)[a]
- Then, on the last line, output exactly: Truth value: <True|False>
- IMPORTANT: If you cannot complete reasoning in 25 steps, you MUST output your best judgment anyway on the last line.

[a]We set a 25-step limit to prevent runaway generation; this is well above the maximum proof depth in ProofWriter.

D.4 ProofWriter (OWA) Evaluation Prompt

System Message

You are a helpful reasoning assistant.

User Message

You are a careful logician. Use classical deductive reasoning.
Facts:
{facts}
Rules:
{rules}
Query: {query}
Instructions:

- First, reason step by step (MAXIMUM 25 steps).
- Then, on the last line, output exactly: Truth value: <True|False|Uncertain>
- IMPORTANT: If you cannot complete reasoning in 25 steps, you MUST output your best judgment anyway on the last line.

E Additional Analysis Setup

Layer-wise NL–symbolic alignment. For Section 5.2, we estimate layer-wise PCA+CCA mappings between pooled NL and symbolic proof activations on the PrOntoQA FICTIONAL-5HOP training split, and compute mean canonical correlations $\bar{\rho}^{(\ell)}$ on 500 held-out validation instances using proof tokens only.

Projection energy and correctness. For Section 5.3, we use Llama-3.1-8B and reuse the same layer-wise projectors $\{P^{(\ell)}\}$ learned on the PrOntoQA training split. On 500 PrOntoQA test instances we generate zero-shot Greedy-CoT traces (no steering) and take the mean projection energy of CoT tokens at each layer as an unsupervised confidence score for ROC-AUC evaluation.

Behavioral comparison of Greedy vs. Steered. For Section 5.4, we again use the same 500 PrOntoQA test instances and generate two CoTs per instance: Greedy-CoT and LSS-CoT with layer-16 steering and $\lambda = 0.08$. Step counts are defined as the number of lines in the explanation before the final `Truth value:` line, and all length and lexical statistics are computed over these explanation spans only. The lexicons used in the style analysis are listed in Appendix I.

F Additional Experimental Details

F.1 Selecting Steering Layers

In our main experiments we restrict steering to a single layer $\ell^\star$ for each model–benchmark pair. Rather than sweeping over all decoder layers, we select a small set of candidate layers based on the strength of the learned multi-view alignment.

Concretely, let L denote the number of decoder layers of a given model. Using the training split for a benchmark, we first estimate the multi-view logical subspace at every layer $\ell \in \{1, \ldots, L\}$ (Section 3.1) and compute a scalar score $\bar{\rho}^{(\ell)}$ defined as the mean canonical correlation of the top-k canonical components. We then form a candidate set $\mathcal{C}$ by taking the 8 layers with the highest $\bar{\rho}^{(\ell)}$ in the middle-to-upper part of the network.

For each model–benchmark pair, we perform a small grid search over $\ell \in \mathcal{C}$ and

$$\lambda \in \{0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14\}$$

on a held-out development set, and select the combination $(\ell^\star, \lambda)$ that maximises CoT accuracy. Once selected, $(\ell^\star, \lambda)$ is fixed for all test instances for that model and benchmark. Empirically, the chosen steering layers for different models and tasks always fall in the middle-to-upper range of the stack, where NL–symbolic alignment is strongest.

F.2 Corpus and tagging for energy analysis

For the token-level energy analyses in Section 5, we construct a synthetic rule-based corpus using the official PrOntoQA generation code.

Corpus construction. We generate 5,000 fictional 5-hop instances using the `Fictional-5hop` configuration with composed rule option.

Token categorization. We assign each proof token to one of six coarse categories: negation, quantifier, copula, entity, concept, and structure. This is done using a small hand-crafted lexicon and simple pattern-based heuristics; tokens that do not match any category are ignored in the energy aggregations.

Subspace for analysis. For a fixed model and layer, we estimate the multi-view logical subspace $U^{(\ell)}$ on the full 5,000-instance corpus and then reuse this subspace when computing all token-level and per-category energies in Section 5.

Lexicon for token categories. For reproducibility, we list the exact lexical sets used to tag tokens in the synthetic PrOntoQA-style corpus into the six coarse categories (entities, concepts, properties, quantifiers, negation, and structure).

Entity names. We tag the following surface forms as entities: `fae`, `rex`, `sally`, `max`, `alex`, `sam`, `polly`, `stella`, `wren`.

Concept names. We tag the following nonce nouns (including both singular and plural forms) as concept tokens: `wumpus`, `yumpus`, `zumpus`, `dumpus`, `rompus`, `numpus`, `tumpus`, `vumpus`, `impus`, `jompus`, `gorpus`, `shumpus`, `lempus`, `sterpus`, `grimpus`, `lorpus`, `brimpus`, `timpus`, `yimpus`, `rempus`, `fompus`, `worpus`, `terpus`, `gerpus`, `kerpus`, `scrompus`, `zhorpus`, `bompus`, `jelpus`, `felpus`, `chorpus`, `hilpus`, `storpus`, `yerpus`, `boompus`, `gwompus`, `rorpus`, `quimpus`, `wumpuses`, `yumpuses`, `zumpuses`, `dumpuses`, `rompuses`, `numpuses`, `tumpuses`, `vumpuses`, `impuses`, `jompuses`, `gorpuses`, `shumpuses`, `lempuses`, `sterpuses`, `grimpuses`, `lorpuses`, `brimpuses`.

Properties. Adjectives and property words are taken from: `blue`, `red`, `brown`, `orange`, `small`, `large`, `metallic`, `wooden`, `luminous`, `liquid`, `transparent`, `opaque`, `nervous`, `happy`, `feisty`, `shy`, `bright`, `dull`, `sweet`, `sour`, `spicy`, `bitter`, `floral`, `fruity`, `earthy`, `hot`, `cold`, `temperate`, `kind`, `mean`, `angry`, `amenable`, `aggressive`, `melodic`, `muffled`, `discordant`, `loud`, `slow`, `moderate`, `fast`, `windy`, `sunny`, `overcast`, `rainy`, `snowy`.

Quantifiers. We tag the following as quantifier tokens: `each`, `every`, `all`, `a`, `an`, `no`, `everything`, `that`.

Negation. We treat `not` as the sole explicit negation marker in this corpus.

Structural tokens. Finally, we group meta-level and connective words into a coarse "structure" category: `true`, `false`, `the`, `query`, `or`, `and`, `premises`, `assume`, `this`, `contradicts`, `with`, `fact`.

F.3 Additional Energy Plots Across Models and Layers

Token-level projection energy. Figure 4 in the main text shows category-wise projection energy for a representative middle layer. Here we provide additional plots for other layers and models.

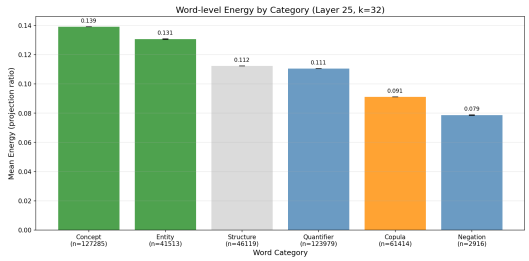


Figure 8: Token-level projection energy $E^{(\ell)}(r)$ aggregated by token category for Llama-3.1-8B at layer 25.

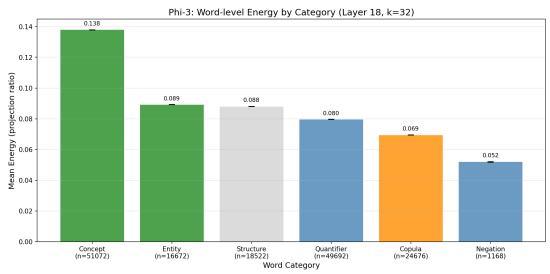


Figure 9: Token-level projection energy $E^{(\ell)}(r)$ aggregated by token category for Phi-3-Mini at layers 18.

Per-direction contribution heatmaps. We also visualize the per-direction contributions for additional layers.

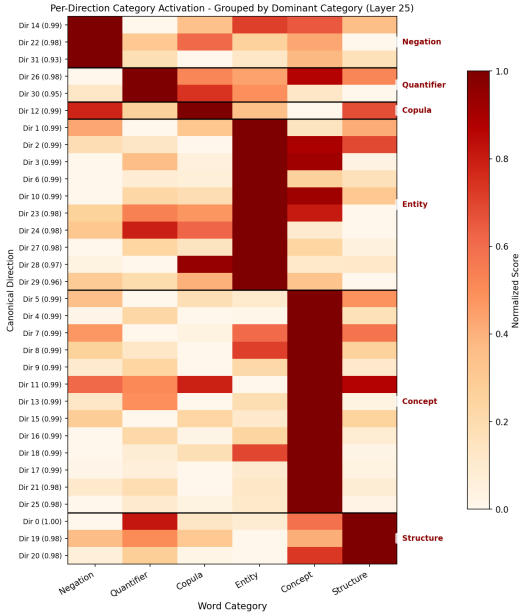


Figure 10: Per-direction contribution heatmap for Llama-3.1-8B at layer 25. Rows correspond to canonical directions and columns to token categories (row-normalized).

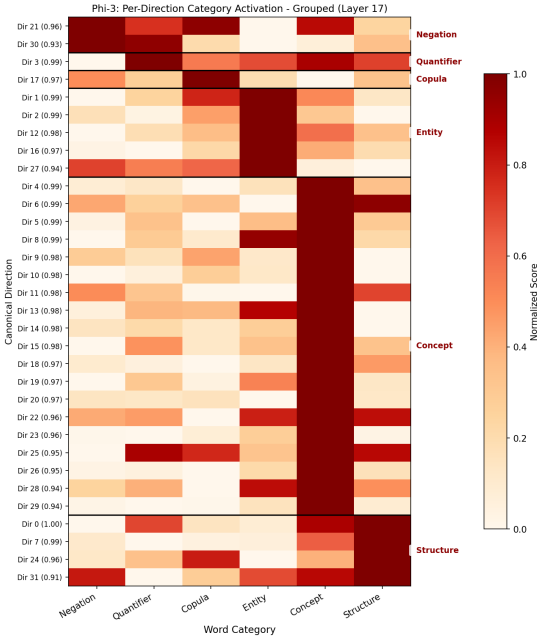


Figure 11: Per-direction contribution heatmaps for Phi-3-Mini at layers 17 (left) and 18 (right).

F.4 Additional results on a reasoning model

To test whether LSS remains effective on a reasoning-specialized model, we evaluate LSS-

Setting (Qwen3-4B, PrOntoQA)	Accuracy (%)
No steer	87.2
LSS ($\lambda = 0.02$, layer=20)	89.8
LSS ($\lambda = 0.04$, layer=20)	89.8
LSS ($\lambda = 0.06$, layer=20)	91.2
LSS ($\lambda = 0.08$, layer=20)	92.0
LSS ($\lambda = 0.10$, layer=20)	**93.2**
LSS ($\lambda = 0.12$, layer=20)	93.0

Table 6: Additional results on Qwen3-4B on PrOntoQA. LSS remains effective on a reasoning-specialized model, with the best setting improving accuracy from 87.2 to 93.2.

CoT on Qwen3-4B on PrOntoQA. Without steering, Qwen3-4B achieves 87.2% accuracy. With LSS, the best setting reaches 93.2% (+6.0 points), showing that steering still yields substantial gains even when the base model already performs strongly.

G Additional Layer-wise Analysis

In Section 5.2, we presented the alignment dynamics for Llama-3.1-8B. Figure 12 presents the corresponding analysis for the remaining models. Despite differences in architecture and depth, all models exhibit the characteristic U-shaped trajectory: high initial alignment, a drop in the middle reasoning layers, and a recovery in the final layers.

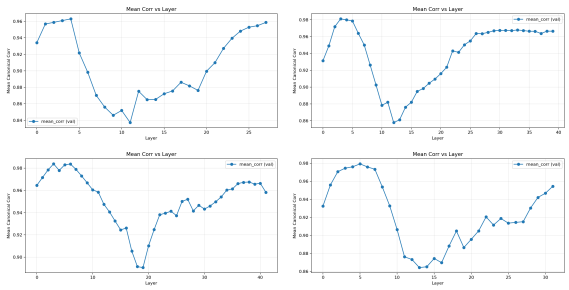


Figure 12: Layer-wise canonical correlation for all evaluated models. The "high-low-high" trend is universal.

H Details on Correctness Prediction

In Section 5.3, we demonstrated that the projection energy in the logical subspace acts as a proxy for reasoning correctness. Figure 13 provides the detailed Receiver Operating Characteristic (ROC) curve for the best-performing layer (Layer 17) of Llama-3.1-8B.

The Area Under the Curve (AUC) is **0.6441**. While not a perfect classifier, this result is significant given that the metric is entirely **unsupervised**—derived solely from the geometric properties of the activation space without any training on labeled correctness data.

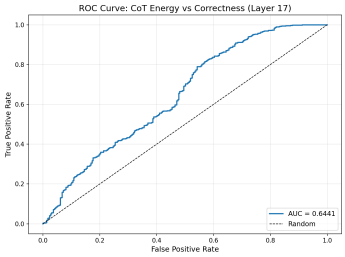


Figure 13: **ROC Curve for Correctness Prediction (Layer 17).** Discriminative performance of the unsupervised projection energy metric (solid line) compared to random chance (dashed line).

I Lexicons for Style Analysis

For the linguistic style analysis in Section 5.4, we group tokens into two small hand-crafted lexicons and report frequency changes at the group level.

Reasoning verbs. We treat the following tokens as "reasoning verbs": `{know, given, conclude, think, assume}`. These mainly occur in metacognitive phrases such as *"we know that ..."*, *"given that ..."*, or *"we can conclude that ..."* and typically mark subjective or self-referential reasoning.

Logical connectives. We treat the following tokens as "logical connectives": `{since, if, then, so, therefore, because}`. These terms usually appear in explicit deductive patterns such as *"since A, ..."*, *"if A then B"*, or *"..., so B"* and directly signal causal structure.

All counts computed as raw token frequencies over the CoT spans, and percentage changes are reported relative to the Greedy-CoT baseline.

J Steering Robustness and Directionality

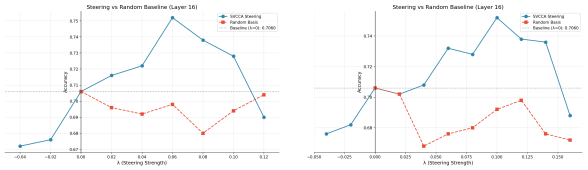
To verify the robustness and semantic specificity of our steering method, we conducted a detailed sensitivity analysis on **Llama-3-8B** using the **ProntoQA** dataset. We examined the impact of steering strength (λ) across different layers (e.g., Layers 16 and 25), explicitly comparing **our steering method** against a **randomly generated orthogonal basis**.

The results (representative plot for Layer 16 shown in Figure 14) highlight two key findings:

- **Directional Semantics:** The sharp performance drop for negative λ values confirms

Category / token	Baseline	Steered	Δ	Δ%
Logical connectives				
All connectives	4263	4466	+203	+4.8%
since	1800	1926	+126	+7.0%
if	1128	1179	+51	+4.5%
then	641	664	+23	+3.6%
so	126	149	+23	+18.3%
therefore	443	447	+4	+0.9%
because	123	101	-22	−17.9%
Reasoning verbs				
All reasoning verbs	7379	6145	-1234	−16.7%
know	1419	944	-475	−33.5%
given	5176	4483	-693	−13.4%
conclude	770	700	-70	−9.1%

Table 7: Llama-3.1-8B: lexical changes in CoTs after symbolic-subspace steering. Logical connectives slightly increase overall, while reasoning verbs decrease.



(a) Layer 16 (b) Layer 25

Figure 14: **Analysis of steering directionality and strength on Llama-3-8B.** We compare Layer 16 (a) and Layer 25 (b). The blue line (**Our Steering**) shows clear semantic directionality compared to the random baseline (orange).

that the extracted vector aligns with a specific, task-relevant semantic direction.

- **Superiority over Random Basis:** Our method consistently outperforms the **random orthogonal basis**, which exhibits high variance and no meaningful directionality, further validating that the performance gains are driven by the learned subspace structure.

J.1 Robustness to the subspace dimension k

We also study robustness to the number of canonical components used to define the shared logical subspace. In our main experiments we set $k = 32$. To test whether performance is sensitive to this choice, we train additional shared subspaces with $k = 16$ and $k = 64$ on Llama-3.1-8B on PrOntoQA, while reusing the same steering layer and λ as in the main experiment (i.e., no retuning).

Table 8 shows that performance is not fragile around the choice of k: a smaller subspace ($k = 16$) remains competitive, while a much larger subspace ($k = 64$) performs worse, likely because it introduces noisier or less stable directions. These results support the use of a moderate low-dimensional shared subspace.

k	**Accuracy (%)**
16	73.0
32 (main)	**75.4**
64	71.0

Table 8: Robustness to the subspace dimension k on PrOntoQA with Llama-3.1-8B. Additional subspaces with $k = 16$ and $k = 64$ are trained, while the steering layer and λ are reused from the main experiment without retuning.

K Projection Energy Definitions

Token-level subspace energy. For a layer-ℓ residual vector $r \in \mathbb{R}^D$ and the multi-view logical subspace basis $U^{(\ell)} = [u_1^{(\ell)}, \dots, u_k^{(\ell)}] \in \mathbb{R}^{D\times k}$, we define the normalized projection energy

$$E^{(\ell)}(r) = \frac{\left\|r^\top U^{(\ell)}\right\|_2^2}{\|r\|_2^2} = \sum_{j=1}^{k} \frac{\left(r^\top u_j^{(\ell)}\right)^2}{\|r\|_2^2}. \quad (4)$$

This is the quantity used in Eq. (1) in the main text.

Global per-direction scores. For a collection of tokens $\{r_i\}_{i=1}^M$ (e.g., all tokens in a dataset or all tokens of a given type), we define the global per-direction score

$$s^{(\ell)}_{\text{global}}(i,j) = \frac{\left(r_i^\top u_j^{(\ell)}\right)^2}{\|r_i\|_2^2}, \quad (5)$$

so that $E^{(\ell)}(r_i) = \sum_{j=1}^k s^{(\ell)}_{\text{global}}(i,j)$. Averaging $s^{(\ell)}_{\text{global}}(i,j)$ over tokens of a given class gives an empirical estimate of the fraction of total variance of that class explained by direction $u_j^{(\ell)}$ (see below).

Subspace-normalized per-direction scores. We also consider a version normalized within the subspace:

$$s^{(\ell)}_{\text{subspace}}(i,j) = \frac{\left(r_i^\top u_j^{(\ell)}\right)^2}{\sum_{m=1}^k \left(r_i^\top u_m^{(\ell)}\right)^2}, \quad (6)$$

which decomposes the token's subspace energy across the k directions and satisfies $\sum_{j=1}^k s^{(\ell)}_{\text{subspace}}(i,j) = 1$ whenever $E^{(\ell)}(r_i) > 0$.

Explained-variance perspective. At the population level, let R be a random residual vector at layer ℓ with covariance $\Sigma^{(\ell)}$. The variance of R along direction $u_j^{(\ell)}$ is $\mathbb{E}[(R^\top u_j^{(\ell)})^2]$, and the total variance is $\mathbb{E}[\|R\|_2^2] = \mathrm{trace}(\Sigma^{(\ell)})$. Thus, averaging $s^{(\ell)}_{\mathrm{global}}(i,j)$ over a set of tokens of a given type yields an empirical estimate of

$$\frac{\mathbb{E}[(R^\top u_j^{(\ell)})^2]}{\mathbb{E}[\|R\|_2^2]},$$

i.e., the fraction of total variance of that token type explained by direction $u_j^{(\ell)}$. Similarly, averaging $s^{(\ell)}_{\mathrm{subspace}}(i,j)$ over high-energy tokens provides an estimate of how variance *within* the logical subspace is distributed across the k directions.