arXiv:2607.22098v1 [cs.AI] 24 Jul 2026

Reasoning Denoiser: Denoising Reasoning Traces for Hallucination Detection in Large Reasoning Models

Junlin Fang[1], **Do Nguyen-Thanh**[1], **Xiaogang Xu**[2], **Zhen Fang**[3] **and Sean Du**[1,*]
[1]College of Computing and Data Science, Nanyang Technological University, [2]Zhejiang University, [3]Australian Artificial Intelligence Institute, University of Technology Sydney, [*]Corresponding author

Abstract

Large reasoning models (LRMs) generate long reasoning traces before producing final answers. While these traces may contain useful signals for hallucination detection, harnessing them is non-trivial because long trajectories often include noisy steps that obscure the cues relevant to truthfulness assessment. In this paper, we identify two prevalent forms of reasoning noises, i.e., irrelevant steps and repetitive steps, and show that both substantially degrade hallucination detection performance. Existing confidence-based scores and naive embedding-based filtering fail to reliably separate noisy from informative steps. To address this challenge, we propose REDE, a novel learning framework for denoising reasoning traces for hallucination detection. Specifically, REDE leverages final-answer attention as an automatic supervision signal to shape the step-level representation space, yielding refined embeddings in which noisy steps can be reliably identified and filtered. REDE can be readily plugged into diverse hallucination detectors by operating on the filtered reasoning trajectory after removing noisy steps. Extensive experiments on multiple reasoning benchmarks show that REDE consistently improves detection performance over competitive baselines.

Date: July 24, 2026
Contact: junlin001@e.ntu.edu.sg; xuefeng.du@ntu.edu.sg

1. Introduction

Hallucination detection for large reasoning models (LRMs) is critical for building reliable AI systems. Recent LRMs, such as OpenAI's o series [27] and DeepSeek-R1 [17], generate long reasoning traces before producing final answers. While these traces can improve reasoning capability, hallucination remains a serious reliability concern [4, 17, 24, 39]. Existing studies have begun to leverage reasoning traces for hallucination detection [40, 54], motivated by the intuition that intermediate reasoning may reveal useful signals about the final answer correctness. However, harnessing reasoning traces is non-trivial. Recent evidence shows that, although long reasoning traces can improve reasoning performance, they can also obscure the cues useful for hallucination detection [10]. This suggests that naively incorporating the full reasoning trace does not consistently benefit detection.

A key challenge is that long reasoning traces often contain *noisy steps* that are weakly informative for hallucination detection. By examining LRM trajectories, we identify two prevalent forms of noisy steps: *irrelevant steps*, which do not contribute meaningful problem-specific information, and *repetitive steps*, whose information is already covered by later steps in the trajectory. As shown in Figure 1(a), these noisy steps can substantially degrade hallucination detection performance, while removing them yields clear improvements. This observation suggests that *identifying and filtering noisy reasoning steps is crucial* for effectively using

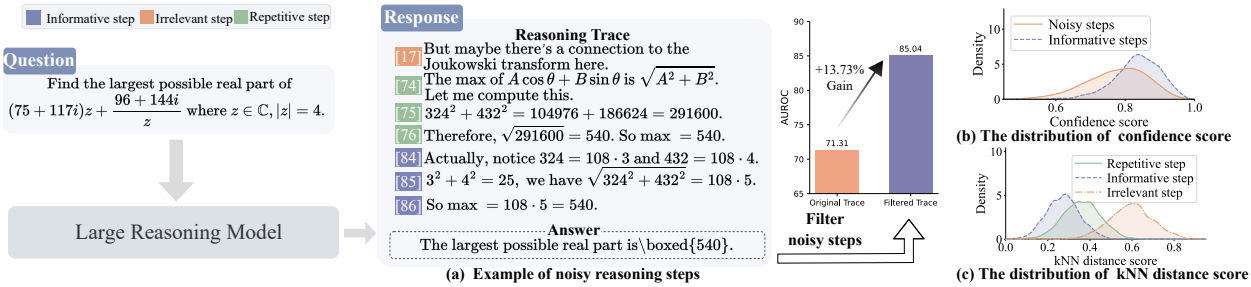

Figure 1 | (a) Long reasoning traces contain *irrelevant* and *repetitive* steps, which hurt hallucination detection; removing them yields clear improvements, as validated with supervised probing [3] using the last-layer embeddings. (b) Confidence-based signals fail to distinguish informative and noisy steps, as their score distributions overlap heavily. (c) Naive kNN-based on raw step embeddings can identify irrelevant steps but fails to separate repetitive from informative ones. Experiments are based on AIME 2024 [26] with Qwen3-8B [64]. We annotate the reasoning steps into informative, irrelevant, and repetitive categories. Annotation details are in Appendix A.3.

reasoning traces in hallucination detection.

A natural approach is to identify noisy steps using confidence-based signals. Recent work has used confidence for improving reasoning-time generation [51, 55, 58, 73]. However, these methods are designed to improve generation quality, rather than to identify which reasoning steps are informative for detecting hallucination in the final answer. Confidence reflects how certain the model is about what it generates, not how useful a step is for downstream hallucination detection. Indeed, Figure 1(b) shows that the confidence distributions of informative and noisy steps overlap heavily, making confidence-based filtering ineffective.

Another possible route is to identify noisy steps in the embedding space. While irrelevant steps may behave as outliers and can sometimes be detected by simple representation-similarity methods such as kNN, this strategy remains inadequate for repetitive steps. Unlike irrelevant steps, repetitive steps are often semantically similar to informative ones and therefore remain close in the raw representation space. As shown in Figure 1(c), naive kNN-based filtering can identify some irrelevant steps but fails to distinguish repetitive from informative ones. These limitations suggest that effective noisy-step filtering requires a signal beyond confidence or raw semantic similarity—one that better reflects each step's contribution to the final answer and to hallucination detection.

To address this challenge, we propose **Reasoning Denoiser (REDE)**, a novel framework for filtering noisy reasoning steps by shaping the step-level representation space. Our key idea is to use the attention score between the final answer token and each reasoning step as a supervision signal, which requires no human annotation. Intuitively, this attention reflects how much a step contributes to forming the final answer: *irrelevant steps* tend to receive low attention because they are weakly connected to the answer, while *repetitive steps* can also receive low attention because their information has already been covered by later steps in the reasoning trajectory. Based on this observation, REDE learns a lightweight projection that pulls informative steps closer together while pushing noisy steps away. The resulting refined step embeddings allow noisy steps to be reliably identified and filtered by simple distance-based methods such as kNN.

REDE can be readily plugged into diverse hallucination detection methods by operating on the filtered reasoning trajectory after removing noisy steps, including probing-based [3, 7], uncertainty-based [50], and verbalized methods [36]. Extensive experiments on four representative reasoning tasks show that REDE consistently improves hallucination detection performance across diverse datasets. Specifically, on a representative benchmark TruthfulQA, REDE improves detection performance by up to 18.69% over using the original unfiltered reasoning trace, and achieves state-of-the-art performance of 87.32%, demonstrating that reasoning-step denoising is an effective and general strategy for hallucination detection in LRMs.

Our key contributions are summarized as follows:

- We identify *noisy reasoning steps* as an overlooked obstacle to hallucination detection. We further show

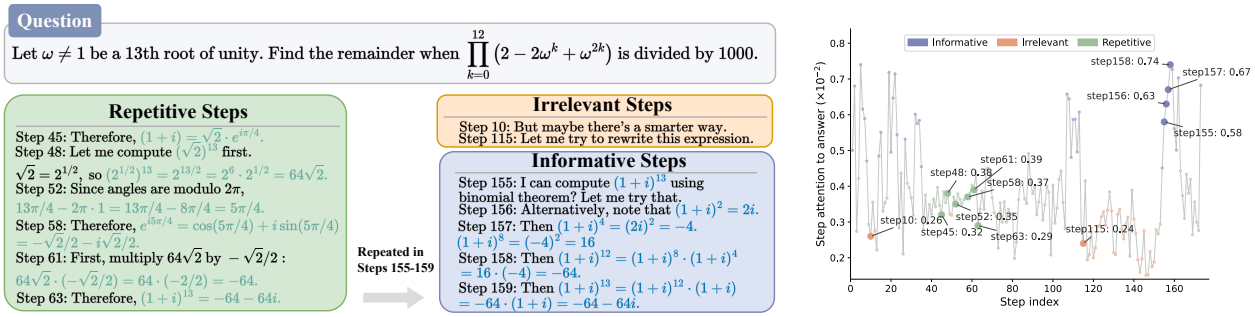

Figure 2 | Attention scores from the final-answer token to each reasoning step on a representative AIME 2024 [26] example with Qwen3-8B [64]. Irrelevant and repetitive steps receive low attention, while informative steps receive high attention. The computation of attention scores is detailed in Section 4.1.

that these noisy steps degrade hallucination detection performance, establishing reasoning-step filtering as an important solution for reliable hallucination detection.

- We propose Reasoning Denoiser (REDE), a novel learning framework that leverages final-answer attention to shape the step-level representation space, enabling reliable filtering of noisy reasoning steps without human annotation.
- Extensive experiments (Section 5.2 and Appendix B) and theoretical analyses (Appendix E) show when and why REDE improves detection performance. We also demonstrate that REDE can be plugged into diverse hallucination detection methods for consistent gains.

2. Problem setup

We study hallucination detection for large reasoning models that generate reasoning traces before final answers, which have become increasingly prominent in today's foundation model landscape.

Reasoning generation. Let an L-layer causal LRM take a prompt $\mathbf{p}$ and generate a response autoregressively. In reasoning-centric settings, we decompose the generated response into a reasoning trace $\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_K)$ and a final answer $\mathbf{a}$, where each $\mathbf{c}_k$ is a contiguous token span corresponding to one reasoning step. Both the number of reasoning steps K and the length of each step can vary across inputs.

Hallucination detection. Given $(\mathbf{p}, \mathbf{C}, \mathbf{a})$, the goal is to determine whether the final answer $\mathbf{a}$ is truthful under the task-specific criterion. Let $y \in \{0, 1\}$ denote the label, where $y = 0$ means the answer is truthful and $y = 1$ means it is hallucinated. We aim to learn a binary detector $G(\mathbf{p}, \mathbf{C}, \mathbf{a}) \in \{0, 1\}$, which predicts the truthfulness of the final answer [40, 54, 59, 70].

Challenge of long reasoning traces. Unlike standard generation, the reasoning trace in an LRM is not uniformly useful for hallucination detection. Long traces often contain noisy steps that are weakly informative about the correctness of the final answer, thereby diluting the signals most relevant to truthfulness assessment [10]. Consequently, naively using the original reasoning trace may hurt detection performance. This motivates our goal of denoising the reasoning trace before hallucination detection. We provide analysis on the noisy reasoning steps next.

3. Understanding noisy reasoning steps for hallucination detection

Before introducing our method, we first seek to better understand the nature of noisy reasoning steps and to identify a suitable signal for detecting them. Our analysis leads to two main observations. First, the final-answer attention in LRMs provides a meaningful signal for distinguishing informative steps from noisy ones. Second, among semantically similar steps, earlier steps are more likely to be redundant, whereas later steps tend to preserve the information most relevant to the final answer. These observations motivate the design of our framework in Section 4.

Final-answer attention reflects step informativeness. We examine whether the LRM's own internal signals can reveal which reasoning steps are useful for hallucination detection. Our key observation is that the attention score from the final-answer token to each reasoning step provides a meaningful relevance signal: steps that contribute more directly to the final answer tend to receive higher attention.

We verify this on our annotated AIME 2024 [26] (*cf.* Appendix A.3) using Qwen3-8B [64], where we compute the attention score assigned by the final-answer token to each reasoning step (formally defined in Eq. 1). Figure 2 shows a representative example. Three patterns emerge clearly: (1) *irrelevant steps* receive low attention; (2) *repetitive steps*, whose content is effectively subsumed by later steps, also receive low attention; and (3) *informative steps* that directly support the final answer receive substantially higher attention. This pattern also holds at the dataset level. Averaged over all examples, informative steps receive a mean attention score of 6.5×10^{-3}, compared to 2.1×10^{-3} for irrelevant steps and 3.3×10^{-3} for repetitive steps. These results suggest that final-answer attention offers a practical and unsupervised signal for separating noisy steps from informative ones. We further validate the effectiveness of this signal in Appendix C.7.

Earlier semantically similar steps are often redundant. Our analysis shows low-attention steps are not homogeneous: some are irrelevant, while others are part of repeated reasoning. We next focus on the latter case. When multiple semantically similar steps appear in a reasoning trajectory, an important question is which occurrence preserves the information most relevant to the answer. If later steps consistently retain the essential information better than earlier ones, this suggests earlier occurrences are redundant. To study this, we cluster reasoning steps based on embedding similarity and, from each cluster, retain either the earliest or the latest step; details are in Appendix A.5.

For each retained subset, we concatenate the selected steps with the original question and answer, and then extract the resulting answer embedding from the LRM. We evaluate two criteria: (1) the cosine similarity between the new answer embedding and the original full-trace answer embedding, which measures how well the retained steps preserve the original reasoning signal; and (2) the hallucination detection performance obtained by applying a linear probe to the new answer embedding. Figure 3 shows retaining the latest steps consistently yields higher cosine similarity and stronger detection performance than retaining the earliest steps. This result suggests that *earlier steps are often redundant*, whereas later steps might preserve the information relevant for hallucination detection.

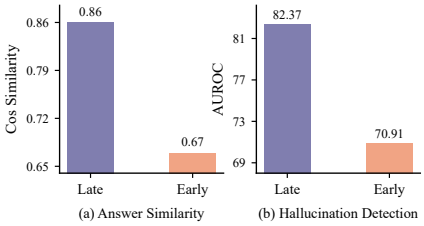

Figure 3 | (a) Cosine similarity between the answer embedding from the retained earliest/latest steps and that from the full reasoning trace. (b) Hallucination detection performance of linear probing based on the retained earliest/latest steps. Results on AIME 2024 [26] with Qwen3-8B [64].

4. Method

Our novel framework REDE addresses the challenge of denoising reasoning traces for hallucination detection. The key idea is to learn a step-level representation space in which informative reasoning steps form a compact region, while noisy steps become isolated and hence easy to filter. Rather than using the raw reasoning trace directly, REDE first identifies an annotation-free supervision signal from the LRM itself, then learns a lightweight projection that reshapes the step embeddings, and finally performs filtering in the learned space before applying a downstream hallucination detector. Our framework is illustrated in Figure 4.

Framework overview. Given a prompt $\mathbf{p}$, a reasoning trace $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, and a final answer $\mathbf{a}$ generated by an LRM, our framework encompasses three components addressing the following questions: (1) *How can we obtain a reliable supervision signal without human annotation?* We estimate how much each reasoning step contributes to the final answer using final-answer attention, yielding an annotation-free step score (Section 4.1). (2) *How can we learn a representation space favorable for noisy-step filtering?*

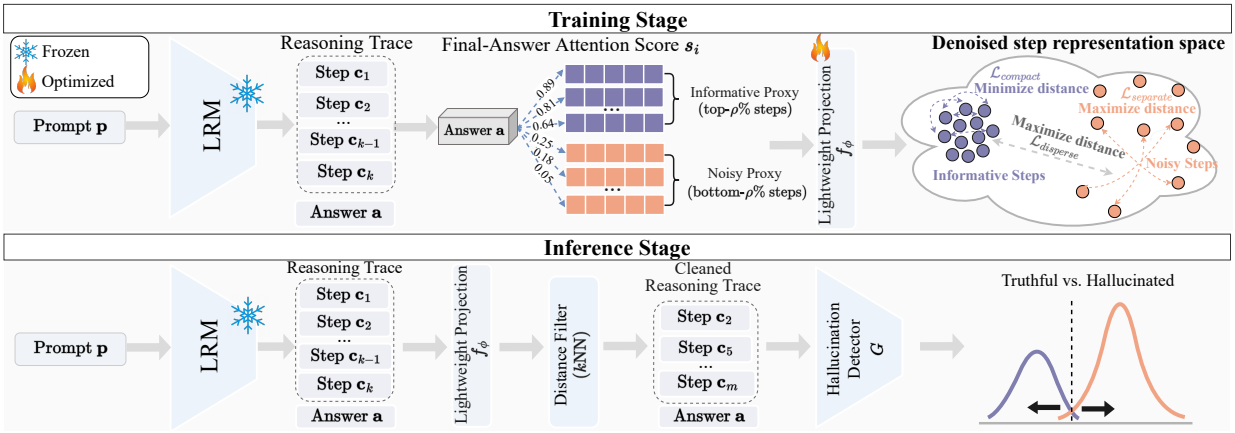

Figure 4 | **Overview of the REDE framework for hallucination detection in LRMs.** During training, REDE scores each reasoning step via final-answer attention and learns a lightweight projection f_ϕ that maps step embeddings into a denoised representation space where informative and noisy steps are well separated. During inference, step embeddings are projected by f_ϕ and noisy steps are filtered via distance-based scoring for downstream hallucination detection.

Using the attention scores as supervision, we train a projection network that maps step embeddings into a new space where informative and noisy steps are more separable (Section 4.2). (3) *How can we perform efficient filtering at test time?* We filter noisy steps in the shaped space and feed the resulting trace to a downstream hallucination detector (Section 4.3). The training and inference procedures are summarized in Algorithms 2 and 3.

4.1. Scoring reasoning steps with final-answer attention

REDE estimates the usefulness of each reasoning step for producing the final answer. As shown in Section 3, the internal attention of LRMs provides a natural signal for this purpose.

Let a_T denote the last token of answer $\mathbf{a}$, $\mathbf{x}_{1:N} = (x_1, \ldots, x_N)$ denote the full causal context visible to a_T, and $I_i \subseteq \{1, \ldots, N\}$ denote the set of token indices corresponding to the i-th reasoning step $\mathbf{c}_i$. We define the score of step $\mathbf{c}_i$ as the total attention mass assigned by a_T to the tokens in that step:

$$s_i = \sum_{t \in I_i} \frac{\exp\big(\mathbf{q}(a_T)^\top \mathbf{k}(x_t)/\sqrt{\omega}\big)}{\sum_{u=1}^{N} \exp(\mathbf{q}(a_T)^\top \mathbf{k}(x_u)/\sqrt{\omega})}, \tag{1}$$

where $\mathbf{q}(\cdot)$ and $\mathbf{k}(\cdot)$ are the query and key projections, and ω is the query/key projection dimension.

The score s_i measures how strongly the final answer attends to step $\mathbf{c}_i$, which admits a simple interpretation. If a step is *irrelevant* to the answer, its key representations are weakly aligned with the query of the answer token, resulting in a low score. If a step is *repetitive*, its content is often subsumed by later steps, and the attention tends to concentrate on the later occurrence, again leading to a low score. By contrast, informative steps that directly support the answer receive higher scores.

From attention scores to representation shaping. While directly thresholding the attention score s_i at test time and discarding low-attention steps is feasible, this approach has two important limitations. First, it requires recomputing attention scores for every test sample, which incurs non-trivial inference overhead. Second, although attention provides a meaningful signal when aggregated across many examples, per-instance attention scores can vary with reasoning trace length and structure, and prior work has shown that vanilla attention may not reliably transfer to downstream tasks without further learning [28, 61]. We verify this empirically in Section 5.2, where direct attention-based filtering consistently underperforms our learning-based approach.

Instead of using attention as a direct filter, we use it as an annotation-free supervision signal to shape the

step-level representation space. By learning a lightweight projection over many examples, REDE converts the per-instance attention signal into a more stable embedding structure that reflects step informativeness. We introduce this process next.

4.2. Learning a denoised step representation space

We now describe how REDE extracts step embeddings and learns a shaped representation space for noisy-step filtering.

Step embedding extraction. A reasoning step often contains both informative and uninformative tokens. To construct a step representation that emphasizes content-bearing tokens, we follow [44] and adopt a perplexity-weighted aggregation of token embeddings. Specifically, for the i-th reasoning step $\mathbf{c}_i$ spanning token indices I_i, we define

$$\mathbf{e}_i = \frac{\sum_{t \in I_i} \mathrm{PPL}(x_t)\, \mathbf{h}(x_t)}{\sum_{t \in I_i} \mathrm{PPL}(x_t)}, \qquad \mathrm{PPL}(x_t) = \frac{1}{p_\theta(x_t \mid x_{<t})}, \tag{2}$$

where $\mathbf{h}(x_t) \in \mathbb{R}^d$ denotes the hidden representation of token x_t and $p_\theta(x_t \mid x_{<t})$ is the next-token probability assigned by the LRM. This weighting assigns larger importance to less predictable tokens, which typically carry richer semantic information (different strategies are compared in Appendix B.7).

Selecting informative and noisy proxies. For each training trace, we rank the steps by the attention scores $\{s_i\}_{i=1}^K$ (Eq. 1). We denote by $\mathcal{T}$ the top-$\rho\%$ steps and by $\mathcal{B}$ the bottom-$\rho\%$ steps. The set $\mathcal{T}$ serves as a proxy set of informative steps, whereas $\mathcal{B}$ serves as a proxy set of noisy steps (the effect of ρ is studied in Appendix B.1).

Projection learning. We train a lightweight projection network $f_\phi : \mathbb{R}^d \to \mathbb{R}^{d'}$ to map each step embedding $\mathbf{e}_i$ to a projected embedding $\mathbf{z}_i = f_\phi(\mathbf{e}_i)$. Throughout, the underlying LRM is frozen and only the projection parameters ϕ are optimized.

The goal of the projection is to induce an embedding distribution favorable for distance-based filtering: informative steps should form a compact cluster, noisy steps should remain scattered rather than collapse together, and the two groups should be well separated. To realize this, we define the cosine similarity between projected steps i and j as $\mathrm{sim}_{ij} = \mathbf{z}_i^\top \mathbf{z}_j / (\|\mathbf{z}_i\|_2 \|\mathbf{z}_j\|_2)$, and optimize the following objective:

$$\mathcal{L} = \underbrace{\mathbb{E}_{i \neq j,\, i,j \in \mathcal{T}} \,(1 - \mathrm{sim}_{ij})}_{\mathcal{L}_{\text{compact}}} + \lambda_{\text{disperse}} \cdot \underbrace{\mathbb{E}_{i \neq j,\, i,j \in \mathcal{B}} \,\mathrm{sim}_{ij}}_{\mathcal{L}_{\text{disperse}}} + \lambda_{\text{separate}} \cdot \underbrace{\mathbb{E}_{i \in \mathcal{T},\, j \in \mathcal{B}} \,\mathrm{sim}_{ij}}_{\mathcal{L}_{\text{separate}}}. \tag{3}$$

Each term plays a distinct role. $\mathcal{L}_{\text{compact}}$ pulls informative steps together, forming a compact region. $\mathcal{L}_{\text{disperse}}$ discourages noisy steps from clustering with one another, so that they remain individually identifiable as outliers. $\mathcal{L}_{\text{separate}}$ pushes informative and noisy steps apart, improving their global separability. Overall, these three terms shape a representation space in which noisy steps have large distances to the informative region and can therefore be filtered effectively using simple non-parametric rules (sensitivity to loss weights is analyzed in Appendix B.4).

Discussion. Our design is intentionally lightweight: the projection network operates on frozen step representations and requires only annotation-free supervision from the LRM itself. Moreover, the learned embedding distribution is detector-agnostic—once the noisy steps are removed, the filtered trace can be passed to any downstream hallucination detector. We further provide a formal mathematical analysis showing that *hallucination detection based on the filtered trace admits a bounded detection error*, where the bound depends directly on the quality of noisy-step filtering induced by the learned projection. The full theorem and proof are deferred to Appendix E.

Table 1 | **Comparison with competitive hallucination detection methods.** "Single Generation" indicates whether the method requires only one generation during inference. "Supervision" indicates whether the method requires ground-truth annotations during training or testing. We report the results of REDE with CCS and supervised probing as downstream detectors. The best results are highlighted in **bold**.

Method	Single Generation	Supervision	Qwen3-8B				DeepSeek-R1-Distill-Llama-8B			
			TruthfulQA	MATH	CodeElo	MULTIHOPQA	TruthfulQA	MATH	CodeElo	MULTIHOPQA
Lexical Similarity [38]	✗	✗	52.00	49.02	55.44	58.26	44.09	66.28	52.57	54.33
SelfCheckGPT [43]	✗	✗	53.11	47.20	51.85	52.61	58.54	51.81	63.33	50.21
Semantic Entropy [30]	✗	✗	54.35	40.69	53.76	66.83	49.43	51.11	72.52	50.13
EigenScore [8]	✗	✗	49.50	55.31	53.88	60.27	45.31	51.17	55.20	58.12
P(True) [29]	✓	✗	64.47	75.16	55.62	75.59	42.93	45.64	53.88	65.25
HaloScope [15]	✓	✗	67.22	61.43	57.19	61.03	54.19	64.18	54.22	51.31
TSV [47]	✓	✓	69.91	71.88	59.42	67.43	52.59	68.75	59.28	59.71
CED [32]	✓	✗	70.29	68.42	60.23	72.18	54.51	65.30	61.72	57.32
HaMI [45]	✓	✓	71.42	70.05	60.44	70.63	58.72	67.41	60.57	61.74
LRM-based										
RHD [54]	✓	✗	58.08	64.15	62.21	58.16	57.16	57.63	65.20	63.02
RACE [59]	✗	✗	80.82	70.55	77.14	74.58	61.23	62.62	59.02	64.62
HalluGuard [69]	✓	✗	51.67	53.58	56.52	55.73	51.15	57.57	51.86	53.55
ARS (CCS) [70]	✓	✗	81.92	77.03	80.05	70.31	75.42	72.18	73.97	68.40
REDE (CCS)	✓	✗	**87.32**$^{\pm 0.83}$	81.33$^{\pm 2.14}$	82.17$^{\pm 1.76}$	73.41$^{\pm 2.53}$	**78.30**$^{\pm 1.91}$	74.24$^{\pm 2.37}$	79.09$^{\pm 0.62}$	72.85$^{\pm 2.85}$
REDE (Probing)	✓	✓	86.44$^{\pm 0.47}$	**87.06**$^{\pm 1.23}$	**87.19**$^{\pm 0.95}$	**82.94**$^{\pm 1.67}$	74.37$^{\pm 2.08}$	**83.91**$^{\pm 0.71}$	**82.55**$^{\pm 1.42}$	**78.71**$^{\pm 1.89}$

4.3. Inference-time filtering and hallucination detection

During inference, REDE performs step filtering without requiring either attention scores or ground-truth labels. Given a test trace $\bar{\mathbf{C}} = \{\bar{\mathbf{c}}_1, \ldots, \bar{\mathbf{c}}_{\bar{K}}\}$, we first extract the step embeddings $\{\bar{\mathbf{e}}_i\}_{i=1}^{\bar{K}}$ using Eq. 2 and project them into the learned space: $\bar{\mathbf{z}}_i = f_\phi(\bar{\mathbf{e}}_i)$. We then compute a distance score S_i for each step as its k-nearest neighbor cosine distance with respect to all other steps within the same trace: $S_i = 1 - \bar{\mathbf{z}}_i^\top \bar{\mathbf{z}}_i^{(k)} / (\|\bar{\mathbf{z}}_i\|_2 \|\bar{\mathbf{z}}_i^{(k)}\|_2)$, where $\bar{\mathbf{z}}_i^{(k)}$ denotes the k-th nearest neighbor of $\bar{\mathbf{z}}_i$ among $\{\bar{\mathbf{z}}_j\}_{j \neq i}$. Intuitively, informative steps cluster together in the shaped space and thus have small k-NN distances, while noisy steps are scattered and yield larger distances. We remove the top-$\zeta\%$ steps with the largest distance scores and retain the remaining steps as the filtered trace $\bar{\mathbf{C}}^{\text{fil}} \subseteq \bar{\mathbf{C}}$. The filtered trace is passed to a downstream hallucination detector G, which outputs $\hat{y} = G(\bar{\mathbf{p}}, \bar{\mathbf{C}}^{\text{fil}}, \bar{\mathbf{a}})$. Importantly, REDE is an agnostic denoising module for diverse downstream detection pipelines.

5. Experiments

Datasets and Models. We evaluate REDE on four reasoning benchmarks: TruthfulQA [37], MATH, constructed by combining MATH500 [18], AIME 2024 [26], and AIME 2025 [46], CODEELO [49], and MULTIHOPQA, which is built following prior work [54] by sampling from HotpotQA [65], 2WikiMultihopQA [19], MuSiQue [56], and Bamboogle [48]. These datasets cover open-domain question answering, mathematical reasoning, code generation, and multi-hop question answering, respectively. For all datasets, we use 25% of the available data for testing, reserve 100 examples for validation, and use the remaining data for training. We evaluate all methods using AUROC, and all reported results are averaged over three random seeds.

We evaluate our method on two widely used LRM families: Qwen3-8B/32B [64] and DeepSeek-R1-Distill-Llama-8B/DeepSeek-R1-Distill-Qwen-32B [17]. Correctness labels are obtained using a strong external judge model Qwen3-32B following prior work [66] (robustness to alternative labeling methods is verified in Appendix C.5). By default, model outputs are generated with greedy decoding. Additional prompt, dataset, sampling strategies, and implementation details are provided in Appendices A.1, A.2, C.3, and A.4,

Table 2 | **Comparison of detecting hallucination using the original reasoning trace vs. our filtered one.** All values are percentages (AUROC) on different detection methods. The best results are highlighted in **bold**.

Model	Dataset	CCS [7]		Supervised Probing [3]		Perplexity [50]		Verbalized Certainty [36]	
		Original	Filtered	Original	Filtered	Original	Filtered	Original	Filtered
Qwen3-8B	TruthfulQA	68.63	**87.32**	80.42	**86.44**	58.17	**63.46**	50.37	**61.10**
	MATH	63.27	**81.33**	81.05	**87.06**	65.16	**71.03**	57.19	**65.13**
	CodeElo	61.64	**82.17**	82.82	**87.19**	61.78	**72.03**	78.89	**83.89**
	MULTIHOPQA	58.19	**73.41**	77.84	**82.94**	52.79	**65.84**	63.75	**69.50**
DeepSeek-R1-Distill-Llama-8B	TruthfulQA	54.14	**78.30**	70.02	**74.37**	52.79	**67.72**	56.61	**61.94**
	MATH	63.62	**74.24**	77.19	**83.91**	65.85	**69.14**	69.71	**76.13**
	CodeElo	55.16	**79.09**	75.07	**82.55**	69.35	**78.49**	79.13	**84.49**
	MULTIHOPQA	60.77	**72.85**	70.05	**78.71**	55.13	**64.45**	50.87	**59.56**

respectively. We also report qualitative case studies, results on GSM8K [11], the impact of filtering on task accuracy, and additional evaluation metrics in Appendices C.1, C.6, C.9, and C.10, respectively.

Baselines. We first compare detection performance using the original unfiltered reasoning trace and our filtered one under four representative hallucination detectors: probing-based detector (Supervised Probing [3] and Contrast-Consistent Search, CCS [7]), uncertainty-based detector (Perplexity [50]), and verbalized detector (Verbalized Certainty [36]). We then compare our method with a comprehensive set of baselines, including: (1) embedding-based methods: Truthfulness Separator Vector (TSV) [47], HaloScope [15], CED [32], and HaMI [45]; (2) consistency-based methods: Lexical Similarity [38], SelfCheckGPT [43], Semantic Entropy [30] and EigenScore [8]; and (3) verbalized methods: P(True) [29]. We further include methods specifically designed for LRMs, including Reasoning and Answer Consistency Evaluation (RACE) [59], Reasoning Hallucination Detection (RHD) [54], HalluGuard [69], and Answer-agreement Representation Shaping (ARS) [70]. For fair comparison, all baselines are evaluated on the same test sets under the default settings reported in their original papers. Implementation details of all baselines are provided in Appendix A.4.

5.1. Main results

Effect of reasoning step filtering. Table 2 compares hallucination detection using the original unfiltered reasoning trace versus our filtered trace across four representative detectors. Across all datasets and two LRMs, our step selection consistently yields substantial improvements, indicating that the benefit stems from filtering noisy steps rather than from any particular downstream detector. For example, on Qwen3-8B over TruthfulQA, AUROC improves from 68.63 to 87.32 for CCS, and from 80.42 to 86.44 for supervised probing. Similar trends hold on DeepSeek-R1-Distill-Llama-8B, confirming that REDE generalizes across different LRM families.

Comparison with competitive baselines. Table 1 compares REDE against a comprehensive set of baselines spanning embedding-based, consistency-based, verbalized, and LRM-specific detectors. REDE achieves the best performance across nearly all settings. With supervised probing, REDE reaches 87.06% on MATH, 87.19% on CodeElo, and 82.94% on MULTIHOPQA with Qwen3-8B. Even without supervision, REDE with CCS substantially outperforms all unsupervised baselines, achieving 87.32% on TruthfulQA. Notably, REDE surpasses the strongest and most relevant representation-shaping baseline ARS [70] by a

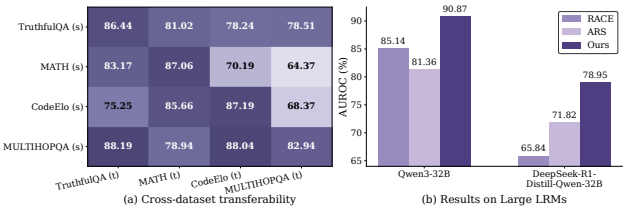

Figure 5 | (a) **Cross-dataset transferability of REDE.** Each row corresponds to a source dataset, and each column corresponds to a target dataset. (b) **Scalability to larger LRMs.** On Qwen3-32B and DeepSeek-R1-Distill-Qwen-32B, REDE outperforms state-of-the-art baselines on TruthfulQA.

large margin (*e.g.*, 81.33 vs. 77.03 on MATH), demonstrating that reasoning-step denoising is a key strategy for hallucination detection. Computational cost is analyzed in Appendix A.7.

Generalization across data distributions. Following prior work [47], we train REDE on a source dataset s and apply the learned projection f_ϕ to a different target dataset t for downstream detection. As shown in Figure 5 (results are based on Qwen3-8B with supervised probing as the downstream detector), REDE transfers reliably across diverse datasets: for example, training on MULTIHOPQA and testing on CodeElo achieves 88.04% AUROC, slightly surpassing the in-domain result of 87.19%. These results indicate that the learned projection captures a general denoising signal that is largely invariant to dataset-specific surface form.

Scalability to larger LRMs. To assess scalability, we evaluate REDE on Qwen3-32B and DeepSeek-R1-Distill-Qwen-32B. As shown in Figure 5(b) (results are based on TruthfulQA with supervised probing as the downstream detector), REDE consistently outperforms the two strongest baselines across both models: on Qwen3-32B, REDE achieves 90.87% AUROC, improving over RACE by 5.73%; on DeepSeek-R1-Distill-Qwen-32B, REDE reaches 78.95%, outperforming RACE by 13.11%. The consistent gains confirm that reasoning-step denoising remains effective in larger LRMs. Additional results over other benchmarks are provided in Appendix C.2.

5.2. Analysis

We provide analysis to understand REDE. Due to space limitations, we defer additional experiments to the Appendix, including (1) further ablations on design choices (Appendix B), (2) qualitative case studies (Appendix C.1), (3) step selection accuracy against human annotations (Appendix C.4), and (4) further analyses on the black-box applicability, and robustness across sampling, labeling, and additional metrics (Appendices C.3–C.10).

Ablation on selection strategies. We compare REDE with six alternative strategies in Table 3: random selection, four existing selection methods [12, 34, 35, 68], and an LLM-as-Judge baseline using Qwen3-32B [64]. Random selection degrades all detectors below the original baseline, confirming that naive subsampling discards useful signal. The other methods also fail to yield consistent improvements over the original reasoning trace, and in some cases even degrade detection performance. This is because they are designed to improve generation quality and efficiency rather than to filter the proper noisy steps for improved detection, whereas REDE specifically shapes representations with final-answer attention signal for this purpose. The LLM-as-Judge baseline also struggles because the long reasoning trace impairs its ability to reliably filter noisy steps. REDE avoids this by operating in the representation space where noisy steps are identified via the LRM's own internal attention signal.

Table 3 | **Comparing step selection strategies.** Results are on TruthfulQA using Qwen3-8B.

Selection Strategy	CCS [7]	Probing [3]	PPL [50]	Verb. [36]
Original	68.63	80.42	58.17	50.37
Random	63.18	74.57	54.23	47.12
STEP [35]	80.17	84.92	60.52	58.33
SPIRIT [12]	83.22	78.14	57.93	56.78
ASAP [68]	76.39	79.18	54.33	52.57
Step Entropy [34]	82.33	**87.31**	61.49	60.03
LLM-as-Judge [64]	70.11	74.81	59.17	50.94
REDE	**87.32**	86.44	**63.46**	**61.10**

Can attention scores alone identify noisy steps? As shown in Table 4, direct attention-based filtering consistently underperforms REDE across all detectors (e.g., 80.51 vs. 87.32 for CCS). Although attention scores provide a useful signal for distinguishing informative from noisy steps (e.g., 80.51 for filtering by final-answer attention signal vs. 58.08 for RHD [54]), they might be sensitive to the heterogeneous reasoning traces and suboptimal as a direct filtering rule for detection. In contrast, REDE uses attention as supervision to learn a projection that reshapes step embeddings, such that informative steps form a compact region while noisy steps become outliers. This enables more reliable distance-based filtering at test time, while avoiding attention computation during inference.

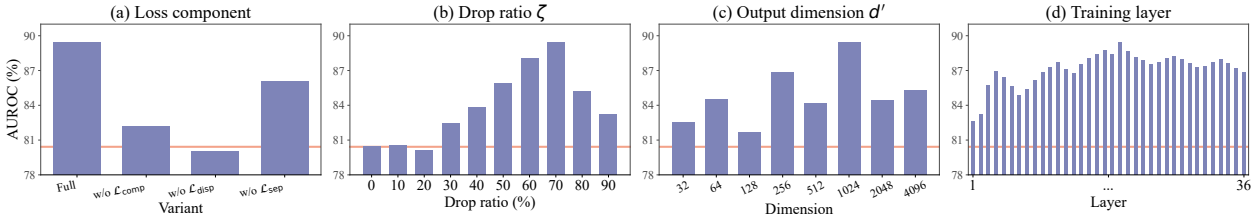

Figure 6 | **Ablation studies.** (a) Effect of individual loss components, (b) effect of drop ratio ζ, (c) effect of output dimension d', and (d) effect of training layer. Results are based on TruthfulQA using Qwen3-8B with supervised probing. The orange horizontal line denotes the performance of using the unfiltered reasoning trace.

Effect of individual loss components. We ablate each loss component by removing it from the full objective (Equation 3). As shown in Figure 6 (a) (results are based on TruthfulQA using Qwen3-8B with supervised probing), removing any single component degrades performance. Removing $\mathcal{L}_{\text{disperse}}$ causes the largest drop (*e.g.*, 86.44→80.06), as noisy steps cluster together and become indistinguishable from informative ones with kNN. Removing $\mathcal{L}_{\text{compact}}$ causes informative steps to spread out and overlap with noisy ones, and removing $\mathcal{L}_{\text{separate}}$ reduces the distance between the two groups. We visualize the projected embeddings via t-SNE in Appendix B.2, confirming that each loss induces a distinct embedding shaping effect.

Table 4 | Comparison of direct attention-based filtering and REDE on TruthfulQA.

Detector	Attention	Ours
CCS	80.51	**87.32**
Supervised Probing	84.97	**86.44**
Perplexity	60.11	**63.46**
Verbalized Certainty	59.73	**61.10**

Effect of the drop ratio ζ. We vary the drop ratio ζ during inference, which controls the fraction of reasoning steps removed as noisy. As shown in Figure 6 (b), performance improves steadily from $\zeta = 0$ (no filtering) and peaks at $\zeta = 70\%$, then declines as ζ increases further. The results suggest that a large fraction of reasoning steps are noisy, while overly filtering will remove informative steps.

Effect of different distance metrics. We compare alternative distance metrics (cosine, Euclidean, and Mahalanobis) for identifying noisy steps in the projected space at test time. All three metrics yield comparable results, confirming the robustness of the learned projection. We defer the detailed comparison to Appendix B.3.

Effect of the output dimension d'. We vary the output dimension d' of the projection module f_ϕ. As shown in Figure 6 (c), performance peaks at $d' = 1024$ and remains competitive across a wide range, suggesting that the denoising signal can be effectively captured at various dimensionalities.

How do different layers impact REDE? In Figure 6 (d), we train REDE using embeddings from different layers. Consistent with prior findings [8, 70], intermediate-to-late layers are most discriminative. REDE improves separability across all layers (*i.e.*, above 82.58 at all layers vs. 80.42 for the original trace), indicating that it does not rely on a specific layer depth.

6. Related work

Hallucination detection has attracted broad interest across multiple paradigms, including logit- and probability-based uncertainty [29, 50, 58], consistency-based scoring [7, 30, 38, 43], verbalized confidence [36], embedding-based approaches [3, 5, 8, 15, 25, 45, 47], and methods that reveal mechanistic distinctions between factual and reasoning-driven hallucinations [41]. While effective for standard LLMs, these methods are not designed for LRMs, whose long reasoning traces introduce additional complexity. Empirical studies show that the reasoning trace actively obscures hallucination signals [10, 66]. Recent LRM-specific detectors exploit reasoning traces more directly [39, 40, 53, 54, 59, 69], but none address the noise steps embedded within the trace itself. The most relevant work is ARS [70], which shapes answer-level representations via latent perturbations at the trace boundary but treats the reasoning trace as a monolithic context, without considering the reasoning noise steps. REDE fills this gap by filtering noisy steps before hallucination

detection.

Reasoning step selection has been explored recently to select reasoning steps for better generation quality and efficiency [14, 16, 20, 21, 23, 42, 57, 62, 63]. Representative approaches include measuring perplexity shifts upon step removal (SPIRIT [12]), using first-token surprisal to identify logical pivots (ASAP [68]), compressing traces via step-level entropy (Step Entropy [34]), training hidden-state probes to evaluate step contributions (STEP [35]), and using various step-level importance signals to identify critical or redundant steps [2, 60, 67]. All of them target generation rather than hallucination detection. As we show in Section 5.2, their selection signals do not reliably identify noisy steps for detection. More broadly, attention scores have been used for token-level filtering and KV cache compression in LLMs [9, 33, 71]. Prior work applies attention patterns to identify pivotal reasoning sentences for interpretability [6]. We explore using final-answer attention as a training and filtering signal to shape a representation space tailored for hallucination detection.

7. Conclusion

We propose REDE, a novel framework for filtering noisy reasoning steps to detect hallucination in LRMs. Our framework leverages final-answer attention to shape the step-level representation space, enabling reliable identification and removal of noisy steps. REDE is lightweight, requires no human annotation, and can be readily plugged into diverse downstream hallucination detectors. Extensive experiments show that REDE consistently improves hallucination detection performance across datasets, models, and detector families. We hope our work can inspire future research on reasoning-trace denoising and more broadly on building reliable methods for assessing LRMs.

References

[1] G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[2] S. An, R. Wang, T. Zhou, and C.-J. Hsieh. Don't think longer, think wisely: Optimizing thinking dynamics for large reasoning models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[3] A. Azaria and T. Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, 2023.

[4] F. Bao, C. Xu, and O. Mendelevitch. DeepSeek-R1 hallucinates more than DeepSeek-V3. https://www.vectara.com/blog/deepseek-r1-hallucinates-more-than-deepseek-v3, Jan. 2025. Vectara Blog, accessed 2025-04-02.

[5] R. Bhatnagar, Y. Sun, C. A. Zhang, Y. Wen, and H. Yang. HALT: Hallucination assessment via latent testing. *arXiv preprint arXiv:2601.14210*, 2026.

[6] P. C. Bogdan, U. Macar, N. Nanda, and A. Conmy. Thought anchors: Which LLM reasoning steps matter? *arXiv preprint arXiv:2506.19143*, 2025.

[7] C. Burns, H. Ye, D. Klein, and J. Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, 2023.

[8] C. Chen, K. Liu, Z. Chen, Y. Gu, Y. Wu, M. Tao, Z. Fu, and J. Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024.

[9] L. Chen, H. Zhao, T. Liu, S. Bai, J. Lin, C. Zhou, and B. Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024.

[10] J. Cheng, T. Su, J. Yuan, G. He, J. Liu, X. Tao, J. Xie, and H. Li. Chain-of-thought prompting obscures hallucination cues in large language models: An empirical evaluation. In C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*. Association for Computational Linguistics, 2025.

[11] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[12] Y. Cui, P. He, J. Zeng, H. Liu, X. Tang, Z. Dai, Y. Han, C. Luo, J. Huang, Z. Li, et al. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18581–18597, 2025.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[14] H. Do, J. Hwang, D. Han, S. J. Oh, and S. Yun. What defines good reasoning in LLMs? dissecting reasoning steps with multi-aspect evaluation. *arXiv preprint arXiv:2510.20603*, 2025.

[15] X. Du, C. Xiao, and Y. Li. HaloScope: Harnessing unlabeled LLM generations for hallucination detection. *Advances in Neural Information Processing Systems*, 37:102948–102972, 2024.

[16] Y. Fu, X. Wang, Y. Tian, and J. Zhao. Deep think with confidence. *arXiv preprint arXiv:2508.15260*, 2025.

[17] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[18] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[19] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, 2020.

[20] B. Hong, J. Liu, K. Zhang, J. Sun, M. Zhang, and Z. Huang. Pruning long chain-of-thought of large reasoning models via small-scale preference optimization. In *The Fourteenth International Conference on Learning Representations*, 2026.

[21] C. Hong, X. Guo, A. C. Singh, E. Choukse, and D. Ustiugov. Slim-SC: Thought pruning for efficient scaling with self-consistency. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34488–34505, 2025.

[22] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[23] B. Hou, Y. Zhang, J. Ji, Y. Liu, K. Qian, J. Andreas, and S. Chang. ThinkPrune: Pruning long chain-of-thought of LLMs via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

[24] Z. Hou, X. Lv, R. Lu, J. Zhang, Y. Li, Z. Yao, J. Li, J. Tang, and Y. Dong. T1: Advancing language model reasoning through reinforcement learning and inference scaling. In *Forty-second International Conference on Machine Learning*, 2025.

[25] J. Hu, G. Tu, C. Shengyu, J. Li, J. Wang, R. Chen, Z. Zhou, and D. Shan. HARP: Hallucination detection via reasoning subspace projection. In *The Fourteenth International Conference on Learning Representations*, 2026.

[26] Hugging Face H4. Huggingfaceh4/aime_2024, 2025. URL https://huggingface.co/datasets/HuggingFaceH4/aime_2024. Dataset.

[27] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney, et al. OpenAI o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[28] S. Jain and B. C. Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, 2019.

[29] S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Hatfield-Dodds, N. DasSarma, E. Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

[30] L. Kuhn, Y. Gal, and S. Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023.

[31] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[32] H. Lee, K.-H. Park, H. Byun, J. Yeom, J. Kim, G.-M. Park, and K. Song. CED: Comparing embedding differences for detecting out-of-distribution and hallucinated text. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14866–14882, 2024.

[33] Y. Li, Y. Huang, B. Yang, B. Venkitesh, A. Locatelli, H. Ye, T. Cai, P. Lewis, and D. Chen. SnapKV: LLM knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

[34] Z. Li, J. Zhong, Z. Zheng, X. Wen, Z. Xu, Y. Cheng, F. Zhang, and Q. Xu. Making slow thinking faster: Compressing LLM chain-of-thought via step entropy. In *The Fourteenth International Conference on Learning Representations*, 2026.

[35] Z. Liang, B. Huang, Z. Wang, and M. Zhang. Hidden states as early signals: Step-level trace evaluation and pruning for efficient test-time scaling. In *Findings of the Association for Computational Linguistics: ACL 2026*, San Diego, California, USA, 2026. Association for Computational Linguistics.

[36] S. Lin, J. Hilton, and O. Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

[37] S. Lin, J. Hilton, and O. Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252, 2022.

[38] Z. Lin, S. Trivedi, and J. Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

[39] H. Lu, Y. Liu, J. Xu, G. Nan, Y. Yu, Z. Chen, and K. Wang. Auditing meta-cognitive hallucinations in reasoning large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[40] H. Lu, M. Pan, R. Li, G. Nan, J. Zhuang, Z. Zhao, Z. Sun, K. Wang, and Y. Liu. Streaming hallucination detection in long chain-of-thought reasoning. *arXiv preprint arXiv:2601.02170*, 2026.

[41] W. Luo, G. Peng, W. Li, S. Wei, F. Song, L. Wang, N. Yang, X. Zhang, J. Jin, F. Wei, et al. Two pathways to truthfulness: On the intrinsic encoding of LLM hallucinations. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, San Diego, California, USA, 2026. Association for Computational Linguistics.

[42] Y. Luo, Y. Song, X. Zhang, J. Liu, W. Wang, G. Chen, W. Su, and B. Zheng. Deconstructing long chain-of-thought: A structured reasoning optimization framework for long cot distillation. *arXiv preprint arXiv:2503.16385*, 2025.

[43] P. Manakul, A. Liusie, and M. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9004–9017, 2023.

[44] P. Miralles-González, J. Huertas-Tato, A. Martín, and D. Camacho. Not all tokens are created equal: Perplexity attention weighted networks for ai generated text detection. *arXiv preprint arXiv:2501.03940*, 2025.

[45] M. Niu, H. Haddadi, and G. Pang. Robust hallucination detection in LLMs via adaptive token selection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[46] OpenCompass Team. AIME 2025 dataset, 2025. URL https://huggingface.co/datasets/opencompass/AIME2025. Problems from the American Invitational Mathematics Examination (AIME) 2025-I & II.

[47] S. Park, X. Du, M.-H. Yeh, H. Wang, and Y. Li. Steer LLM latents for hallucination detection. In *International Conference on Machine Learning*, pages 47971–47990. PMLR, 2025.

[48] O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, 2023.

[49] S. Quan, J. Yang, B. Yu, B. Zheng, D. Liu, A. Yang, X. Ren, B. Gao, Y. Miao, Y. Feng, et al. CodeElo: Benchmarking competition-level code generation of LLMs with human-comparable elo ratings. *arXiv preprint arXiv:2501.01257*, 2025.

[50] J. Ren, J. Luo, Y. Zhao, K. Krishna, M. Saleh, B. Lakshminarayanan, and P. J. Liu. Out-of-distribution detection and selective generation for conditional language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[51] T. Santosh, Y. T. Elkhayat, O. Ichim, P. Shetty, D. Wang, Z. Ma, A. Nourbakhsh, and X. Liu. CoCoLex: Confidence-guided copy-based decoding for grounded legal text generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19002–19018, 2025.

[52] T. Sellam, D. Das, and A. Parikh. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7881–7892, 2020.

[53] P. Srey, X. Wu, and L. A. Tuan. Unsupervised hallucination detection by inspecting reasoning processes. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22128–22140, 2025.

[54] Z. Sun, Q. Wang, H. Wang, X. Zhang, and J. Xu. Mechanistic detection and mitigation of hallucination in large reasoning models. In *The Fourteenth International Conference on Learning Representations*, 2026.

[55] L. Tang, W. Gao, B. Zhao, L. Ma, B. Yang, Y. Zou, et al. Thinking by subtraction: Confidence-driven contrastive decoding for LLM reasoning. *arXiv preprint arXiv:2602.18232*, 2026.

[56] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

[57] S. Tu, Y. Li, Y. Bai, L. Hou, and J. Li. DeepPrune: Parallel scaling without inter-trace redundancy. *arXiv preprint arXiv:2510.08483*, 2025.

[58] N. Varshney, W. Yao, H. Zhang, J. Chen, and D. Yu. A stitch in time saves nine: Detecting and mitigating hallucinations of LLMs by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*, 2023.

[59] C. Wang, W. Su, Q. Ai, and Y. Liu. Joint evaluation of answer and reasoning consistency for hallucination detection in large reasoning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33377–33385, 2026.

[60] S. Wang, E. Zhao, and X. Ren. Stepwise informativeness search for improving LLM reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25291–25309, 2025.

[61] S. Wiegreffe and Y. Pinter. Attention is not not explanation. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 11–20, 2019.

[62] C. Wu, Q. Cao, C. Li, Z. Wang, C. Xue, Y. Fan, W. Xi, and X. He. Beyond token length: Step pruner for efficient and accurate reasoning in large language models. *arXiv preprint arXiv:2510.03805*, 2025.

[63] H. Xia, C. T. Leong, W. Wang, Y. Li, and W. Li. TokenSkip: Controllable chain-of-thought compression in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3351–3363, 2025.

[64] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[65] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380, 2018.

[66] Z. Yao, Y. Liu, Y. Chen, J. Chen, J. Fang, L. Hou, J. Li, and T.-S. Chua. Are reasoning models more prone to hallucination? *arXiv preprint arXiv:2505.23646*, 2025.

[67] H. Yuan, B. Yu, H. Li, S. Yang, C. D. Wang, Z. Yu, X. Xu, W. Qi, and K. Chen. Not all tokens are what you need in thinking. *arXiv preprint arXiv:2505.17827*, 2025.

[68] W. Zeng, Y. Wang, C. Hu, Y. Shi, C. Wan, H. Zhang, and X. Gu. Pruning the unsurprising: Efficient code reasoning via first-token surprisal. *arXiv preprint arXiv:2508.05988*, 2025.

[69] X. Zeng, J. Lin, Y. Yan, F. Guo, L. Shi, J. Wu, and D. Zhou. HalluGuard: Demystifying data-driven and reasoning-driven hallucinations in LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026.

[70] J. Zhang, B. Guo, Y. Jiang, H. Wang, B. An, and X. Du. Harnessing reasoning trajectories for hallucination detection via answer-agreement representation shaping. *International Conference on Machine Learning*, 2026.

[71] Z. Zhang, Y. Sheng, T. Zhou, T. Chen, L. Zheng, R. Cai, Z. Song, Y. Tian, C. Ré, C. Barrett, et al. H2O: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

[72] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

[73] H. Zubkova, J.-H. Park, and S.-W. Lee. SUGAR: leveraging contextual confidence for smarter retrieval. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

A. Additional experimental details

A.1. Input prompts

We provide the detailed prompts used in our experiments for two purposes: (1) generating the original reasoning traces and final answers, and (2) evaluating the correctness of model-generated answers with an external judge model.

Generation prompts. For each dataset, we prepend a task-specific system instruction before the question. The system instructions are as follows:

- **TruthfulQA:** "*You are a factual question answering expert. Provide one concise and direct final answer.*"
- **MATH:** "*You are a mathematical reasoning expert. Solve the following problem step by step and give the final answer in the format \boxed{YOUR_ANSWER}. Answer concisely.*"
- **CodeElo:** "*You are a competitive programming expert. Solve the following Codeforces problem and provide a correct and efficient C++17 solution. Return only one final C++ code block. Answer concisely.*"
- **MULTIHOPQA:** "*You are a multi-hop question answering expert. Reason across the evidence*[1] *and provide one concise final answer.*"

These system instructions are combined with the model-specific prompt templates below.

Qwen Prompt

```
<|im_start|>user
{system_instruction}
Question: {question}
<|im_end|>
<|im_start|>assistant
```

Figure 7 | Prompt used to generate reasoning traces and final answers for Qwen models.

DeepSeek-R1 Prompt

```
<|begin_of_sentence|><|User|>
{system_instruction}
Question: {question}
<|Assistant|><think>
```

Figure 8 | Prompt used to generate reasoning traces and final answers for DeepSeek-R1 models.

Correctness evaluation prompt. Following prior work [66, 70, 72], we use Qwen3-32B as an external judge to assess whether a model-generated answer is correct, using the prompt shown in Figure 9. Answers graded as `INCORRECT` or `NOT_ATTEMPTED` are treated as hallucinations.

A.2. Dataset details

We evaluate REDE on four tasks: TruthfulQA, MATH, CodeElo, and MULTIHOPQA.

[1] Here, "evidence" refers to the multiple pieces of supporting information that must be connected to answer a multi-hop question. For example, answering "Was the director of *Jaws* born in the same country as the author of *Hamlet*?" requires combining evidence about the film and its director, the play and its author, and their birth countries.

Judge Prompt

```
Your job is to look at a question, multiple acceptable gold targets, and a predicted answer,
    and then assign a grade of either "CORRECT", "INCORRECT", or "NOT_ATTEMPTED".

IMPORTANT: The question has multiple acceptable correct answers provided as gold targets. The
    predicted answer is CORRECT if it matches any one of the gold targets according to the
    criteria below.

A predicted answer is CORRECT if:
- It fully contains the important information from at least one gold target.
- It does not contain any information that contradicts any gold target.
- Only semantic meaning matters; capitalization, punctuation, grammar, and order do not matter.
- Hedging or guessing is allowed as long as at least one gold target is fully included and the
    response contains no incorrect or contradictory statements.

A predicted answer is INCORRECT if:
- Any factual statement in the answer contradicts all the gold targets.
- Hedging statements are still considered incorrect if they introduce contradictory content.

A predicted answer is NOT_ATTEMPTED if:
- The important information from any gold target is not included.
- No statements in the answer contradict any of the gold targets.

Additional notes:
- Numerical answers must match the last significant figure of the gold target.
- If the gold target contains more information than the question requires, the predicted answer
     only needs to match the information requested by the question.
- Do not penalize omissions that are clearly inferable from the question.
- Do not penalize small typos in names if the intended meaning is clear.

Question: {question}
Gold targets: {targets}
Predicted answer: {predicted_answer}

Return only one of the following:
A
B
C

where:
A = CORRECT
B = INCORRECT
C = NOT_ATTEMPTED
```

Figure 9 | Prompt used to evaluate the correctness of model-generated answers with Qwen3-32B.

Verbalized Uncertainty Prompt for Qwen Models

```
<|im_start|>user
A language model was asked: {question}
The model's answer was: {answer}

a. Very Certain
b. Fairly Certain
c. Moderately Certain
d. Somewhat Certain
e. Not Certain
f. Very Uncertain

Analyze its answer. How certain are you of the model's answer?
<|im_end|>
<|im_start|>assistant
```

Figure 10 | Prompt used for verbalized uncertainty estimation with Qwen models.

Verbalized Uncertainty Prompt for DeepSeek-R1 Models

```
<|begin_of_sentence|><|User|>
A language model was asked: {question}
The model's answer was: {answer}

a. Very Certain
b. Fairly Certain
c. Moderately Certain
d. Somewhat Certain
e. Not Certain
f. Very Uncertain

Analyze its answer. How certain are you of the model's answer?
<|Assistant|><think>
</think>
Answer:
```

Figure 11 | Prompt used for verbalized uncertainty estimation with DeepSeek-R1 models.

TruthfulQA. We use TruthfulQA [37], which contains 817 questions covering factual knowledge across diverse topics.

MATH. We construct the evaluation set by combining MATH500 [18], AIME 2024 [26], and AIME 2025 [46], resulting in 560 problems in total.

CodeElo. We use CodeElo [49], which contains 408 Codeforces problems requiring competition-level code generation.

MULTIHOPQA. We construct the evaluation set following prior work [54] by sampling 1,000 questions from HotpotQA [65], 2WikiMultihopQA [19], MuSiQue [56], and Bamboogle [48].

For all datasets, we reserve 25% of the available examples for testing, use 100 examples for validation, and use the remaining examples for training. Unless otherwise specified, model outputs are generated with greedy decoding.

Reasoning trace statistics. Table 5 reports the average number of reasoning tokens and reasoning steps per sample across datasets and models. Reasoning trace length varies substantially across tasks: CodeElo produces the longest traces (up to 7,808 tokens and 254.3 steps on average), while TruthfulQA and MULTIHOPQA are considerably shorter. Math problems fall in between, reflecting the iterative nature of symbolic reasoning. These statistics motivate the need for effective step filtering, as longer traces contain more opportunities for noisy steps to accumulate.

Table 5 | Average reasoning trace length (tokens) and number of reasoning steps per sample.

Model	Dataset	Avg. Tokens	Avg. Steps
Qwen3-8B	TruthfulQA	2,294	65.8
	MATH	5,606	145.0
	CodeElo	7,808	254.3
	MULTIHOPQA	2,969	86.6
DeepSeek-R1-Distill-Llama-8B	TruthfulQA	2,090	54.5
	MATH	4,251	112.4
	CodeElo	7,656	267.2
	MULTIHOPQA	2,210	50.4

A.3. Annotation details on AIME 2024

To better understand the structure of noisy reasoning traces, we annotate the reasoning steps on AIME 2024 generated by Qwen3-8B. These annotations are used for analysis, including the examples in Figure 1 and the step-selection accuracy evaluation in Section 5.2.

Step segmentation. Before annotation, we segment each reasoning trace into step-level units. Following prior work on reasoning hallucination analysis [54], we adopt a hybrid segmentation strategy that combines discourse-marker cues with formatting-based boundaries. Specifically, we split the trace at explicit discourse markers that typically indicate a transition in reasoning, such as “Wait”, “But”, “However”, “Hmm”, and “Alternatively”. We further use formatting-based boundaries, such as paragraph breaks, when they correspond to coherent changes in reasoning. This procedure yields a sequence of contiguous reasoning steps for subsequent annotation.

Annotation labels. Each reasoning step is assigned exactly one of three labels:

- **Informative**, if it provides non-redundant, problem-specific information that directly supports the final answer.

Table 6 | Statistics of manual step annotations on AIME 2024 with Qwen3-8B.

Step Category	Count
Informative	1,178
Irrelevant	1,050
Repetitive	2,127

- **Irrelevant**, if it does not contribute meaningful problem-specific information for solving the problem or verifying the answer.
- **Repetitive**, if its main content is semantically redundant with later reasoning steps and is expressed more completely or more decisively afterwards.

Annotation procedure. We adopt a two-stage procedure to ensure annotation quality. In the first stage, we conduct three successive rounds of labeling on the segmented reasoning steps, where the label guidelines are iteratively refined based on observed ambiguous cases before producing an initial label for each step. In the second stage, three graduate students with relevant expertise in mathematical reasoning and NLP *independently* verify the initial labels on a randomly sampled subset without access to each other's judgments; disagreements with the initial label are resolved by majority vote among the verifiers, and the result serves as the final label for the sampled subset. The three verifiers achieve a Fleiss' κ of 0.76 on this subset, indicating substantial agreement [31], which supports the reliability of the initial labels.

Annotation statistics. Table 6 summarizes the distribution of the final labels across the three categories. Overall, repetitive steps form the largest category (2,127 steps, 48.8%), while informative steps account for 1,178 steps (27.1%) and irrelevant steps for 1,050 steps (24.1%). Taken together, irrelevant and repetitive steps make up 3,177 steps (73.0%) of the annotated traces, so we treat them as a single noisy class for the accuracy evaluation in Section 5.2.

A.4. Implementation details

REDE. For each reasoning trace, we first segment the trace into reasoning steps and extract a step-level embedding from the hidden representations of the backbone model. Unless otherwise specified, we use the final-layer hidden states and compute each step embedding by perplexity-weighted averaging over its tokens, as described in the main paper. The projection module f_ϕ is implemented as a two-layer MLP with ReLU activation. We train it using Adam with learning rate $1\mathrm{e}{-4}$, weight decay $1\mathrm{e}{-5}$, cosine learning rate decay, and batch size 128. The projection dimension, the loss weights, the top/bottom selection ratio, and the step drop ratio are selected on the validation set. At test time, noisy steps are identified using kNN distance in the projected space with $k = 15$, and the remaining steps are used to construct the final detection representation. For the step score in Eq. 1, we uniformly average the per-head attention probabilities over all heads in the chosen layer.

Downstream detectors. We evaluate REDE with four downstream detectors: supervised probing [3], CCS [7], verbalized uncertainty [36], and perplexity [10, 50]. For supervised probing, we follow prior work [3] and adopt an MLP classifier with a 512-dimensional hidden layer. We use dropout 0.6, Gaussian input noise with $\sigma = 0.008$, and BatchNorm momentum 0.05. The classifier is trained with `BCEWithLogitsLoss` using SGD with learning rate 8×10^{-3}, momentum 0.9, and weight decay 5×10^{-2} for 100 epochs with batch size 128. A ReduceLROnPlateau scheduler is used with factor 0.5, patience 7, and minimum learning rate 10^{-4}. Unless otherwise specified, we extract representations from layers 0–35 and use the last-token representation for classification. For CCS, we follow its original setup [7] and train a lightweight CCS classifier instantiated as a single linear layer, optimized with AdamW (learning rate 1e-3, weight decay 1e-2). The model is trained

on balanced positive–negative embedding pairs constructed via difference vectors for 1000 epochs and with a logical consistency loss. We repeat training for 10 random initializations and retain the best-performing checkpoint. For verbalized uncertainty, we follow prior work [36] and prompt the backbone model to self-assess its confidence using a six-level certainty scale and use the resulting score for AUROC evaluation. For perplexity-based detection, we follow prior work [10, 50] and use the mean perplexity over answer tokens as the detection score.

Baselines. For lexical similarity [38] and semantic entropy [30], we reproduce them based on the original papers using multiple sampled responses from the same backbone model. SelfCheckGPT [43] is implemented with its NLI-based variant, which uses a fine-tuned DeBERTa-v3-large model[2] to estimate the contradiction or entailment relation between the most-likely answer and sampled responses. For P(True) [29], we reproduce it following the original paper. For TSV [47], we follow the default settings described in the original paper and train it on the same dataset using embeddings extracted from the same layer as in our main experiments. For HaloScope [15], we train it on the same unlabeled training set as our method and use pseudo-labels generated via PCA to train the lightweight MLP classifier. For EigenScore [8], we search the optimal number of sampled generations on the validation set and report the best AUROC. For RACE [59], we follow the original implementation and adopt the setting without pre-extracted chains of thought, where reasoning steps are summarized automatically by the built-in extractor. For RHD [54], we reproduce it based on the official code. For HalluGuard [69], we follow the default configuration described in the original paper. For ARS [70], we use the officially released codebase with the default hyperparameters reported in the paper. All baselines are evaluated on the same test sets under the default settings reported in their respective papers.

A.5. Clustering details for Section 3

We describe the procedure for clustering semantically similar reasoning steps and retaining representative steps from each cluster. The full procedure is summarized in Algorithm 1.

Given a reasoning trace $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, we first extract step embeddings $\{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$ using perplexity-weighted averaging (Eq. 2). We then iteratively assign each step to an existing cluster if its cosine similarity to the cluster centroid exceeds a threshold τ_c, or create a new cluster otherwise. After all steps are assigned, we retain either the earliest or latest step from each cluster based on its position in the trace. The retained steps are concatenated with the original question and answer, and the resulting sequence is fed through the LRM to extract the last-token hidden state of the answer as the answer embedding. We set $\tau_c = 0.7$.

A.6. Training and inference algorithms of our REDE

We provide the complete training and inference procedures of REDE in Algorithm 2 and Algorithm 3, respectively. During training, REDE computes step-level attention scores from the LRM's own attention mechanism and uses them as supervision to train a lightweight projection module f_ϕ. During inference, REDE applies the learned projection to identify and remove noisy steps without requiring attention scores or ground-truth labels, and passes the filtered reasoning trace to a downstream hallucination detector.

A.7. Compute resources and time

Software and hardware. We conduct all experiments using Python 3.10 and PyTorch 2.3.1 on NVIDIA A100 GPUs with 80 GB memory.

[2] https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli

Algorithm 1: Step Clustering and Retention

Input: Reasoning trace $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, similarity threshold τ_c, retention mode $m \in \{\texttt{earliest}, \texttt{latest}\}$
Output: Retained reasoning trace $\mathbf{C}^{\text{ret}}$
Extract step embeddings $\{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$ via Eq. 2
Initialize cluster list $\mathcal{G} \leftarrow \emptyset$
for $i = 1, \ldots, K$ **do**
  assigned $\leftarrow$ False
  for *each cluster* $G \in \mathcal{G}$ **do**
    Compute centroid $\bar{\mathbf{e}}_G = \frac{1}{|G|} \sum_{j \in G} \mathbf{e}_j$
    if $\cos(\mathbf{e}_i, \bar{\mathbf{e}}_G) \geq \tau_c$ **then**
      $G \leftarrow G \cup \{i\}$
      assigned $\leftarrow$ True
      break
  if assigned $=$ False **then**
    Append singleton cluster $\{i\}$ to $\mathcal{G}$
Initialize retained index set $\mathcal{I}^{\text{ret}} \leftarrow \emptyset$
for *each cluster* $G \in \mathcal{G}$ **do**
  if $m = \texttt{earliest}$ **then**
    $\mathcal{I}^{\text{ret}} \leftarrow \mathcal{I}^{\text{ret}} \cup \{\min(G)\}$
  else
    $\mathcal{I}^{\text{ret}} \leftarrow \mathcal{I}^{\text{ret}} \cup \{\max(G)\}$
Sort the indices in $\mathcal{I}^{\text{ret}}$ in ascending order
Construct $\mathbf{C}^{\text{ret}}$ by retaining $\{\mathbf{c}_i \mid i \in \mathcal{I}^{\text{ret}}\}$ in the original trace order
return $\mathbf{C}^{\text{ret}}$

Training and inference time. We report the training and inference time of REDE on TruthfulQA with Qwen3-8B. Based on tracked runs, training REDE takes 681 seconds, and inference on the TruthfulQA test set takes 49 seconds. For comparison, Semantic Entropy [30] requires 1141 seconds, RACE [59] requires 1507 seconds, and SelfCheckGPT [43] requires 2077 seconds under the same setting. These results show that REDE has a clear efficiency advantage over these baselines while achieving better hallucination detection performance.

A.8. Licenses of existing assets

We list the licenses of the datasets and pre-trained models used in this work. All assets are used in accordance with their respective licenses and intended research use.

Datasets. TruthfulQA [37], 2WikiMultihopQA [19], and CodeElo [49] are released under Apache License 2.0; MATH [18], AIME 2024 [26], AIME 2025 [46], Bamboogle [48], and GSM8K [11] are released under MIT License; HotpotQA [65] is released under CC BY-SA 4.0; and MuSiQue [56] is released under CC BY 4.0.

Pre-trained models. Qwen3-8B and Qwen3-32B [64] are released under Apache License 2.0. DeepSeek-R1-Distill-Llama-8B and DeepSeek-R1-Distill-Qwen-32B [17] are released under MIT License, with their

Algorithm 2: Training Procedure of REDE

Input: Training set $\mathcal{D}_{\text{train}} = \{(\mathbf{p}^{(i)}, \mathbf{C}^{(i)}, \mathbf{a}^{(i)})\}_{i=1}^{N}$, frozen LRM with parameters θ, selection ratio ρ, loss weights $\lambda_{\text{disperse}}$ and $\lambda_{\text{separate}}$, number of epochs E
Output: Trained projection module f_ϕ
Initialize projection module $f_\phi \colon \mathbb{R}^d \to \mathbb{R}^{d'}$
`// Step 1: Construct proxy sets from final-answer attention`
for *each training sample* $(\mathbf{p}, \mathbf{C}, \mathbf{a})$ *in* $\mathcal{D}_{\text{train}}$ **do**
  Let $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$
  for $i = 1, \ldots, K$ **do**
    Compute PPL-weighted step embedding $\mathbf{e}_i$ via Eq. 2
    Compute step-level attention score s_i via Eq. 1
  $\mathcal{T} \leftarrow$ indices of the top $\rho\%$ steps ranked by $\{s_i\}_{i=1}^{K}$
  $\mathcal{B} \leftarrow$ indices of the bottom $\rho\%$ steps ranked by $\{s_i\}_{i=1}^{K}$
  Store $(\{\mathbf{e}_i\}_{i=1}^{K}, \mathcal{T}, \mathcal{B})$ for training
`// Step 2: Train the projection module`
for *epoch* $= 1, \ldots, E$ **do**
  for *each mini-batch of stored* $(\{\mathbf{e}_i\}, \mathcal{T}, \mathcal{B})$ **do**
    Compute projected embeddings $\mathbf{z}_i = f_\phi(\mathbf{e}_i)$ for all steps in the mini-batch
    Compute $\mathcal{L}_{\text{compact}}$, $\mathcal{L}_{\text{disperse}}$, and $\mathcal{L}_{\text{separate}}$ from the projected embeddings and proxy sets
    Compute total loss $\mathcal{L} = \mathcal{L}_{\text{compact}} + \lambda_{\text{disperse}}\mathcal{L}_{\text{disperse}} + \lambda_{\text{separate}}\mathcal{L}_{\text{separate}}$ via Eq. 3
    Update ϕ by gradient descent on $\mathcal{L}$
return f_ϕ

base models additionally subject to the Llama 3.1 Community License and Apache License 2.0, respectively. The DeBERTa-v3-large-mnli-fever-anli-ling-wanli model used for the NLI variant of SelfCheckGPT [43] is released under MIT License.

B. Additional ablation studies

We conduct additional ablation studies to validate the key design choices of REDE. Unless otherwise specified, all experiments are conducted on TruthfulQA using Qwen3-8B with supervised probing as the downstream detector.

B.1. Effect of training selection ratio

During training, we select the top $\rho\%$ and bottom $\rho\%$ of steps by attention score to form the informative and noisy sets, respectively. Table 7 reports detection performance under varying ρ across four downstream detectors. A small ρ provides insufficient contrast between the two groups, weakening the contrastive signal; a large ρ reduces the separation between the two ends of the attention distribution, leading to less discriminative contrastive samples. Performance consistently peaks in the range $\rho \in [20, 25]$, confirming that well-separated contrastive samples are important for learning a discriminative projection. The training-time ratio ρ complements the inference-time drop ratio $\zeta = 70\%$: ρ selects the most contrastive samples at the two ends of the attention distribution for effective supervision, while ζ enables thorough denoising at test time.

Algorithm 3: Inference Procedure of REDE

Input: Test sample $(\bar{\mathbf{p}}, \bar{\mathbf{C}}, \bar{\mathbf{a}})$, trained projection f_ϕ, number of nearest neighbors k, drop ratio ζ, downstream detector G

Output: Hallucination prediction $\hat{y} \in \{0, 1\}$

Let $\bar{\mathbf{C}} = \{\bar{\mathbf{c}}_1, \ldots, \bar{\mathbf{c}}_{\bar{K}}\}$

`// Step 1: Extract and project step embeddings`

for $i = 1, \ldots, \bar{K}$ **do**

  Compute PPL-weighted step embedding $\bar{\mathbf{e}}_i$ via Eq. 2

  Compute projected embedding $\bar{\mathbf{z}}_i = f_\phi(\bar{\mathbf{e}}_i)$

`// Step 2: Compute within-trace` k`NN distance scores`

for $i = 1, \ldots, \bar{K}$ **do**

  Let $\bar{\mathbf{z}}_i^{(k)}$ denote the k-th nearest neighbor of $\bar{\mathbf{z}}_i$ among $\{\bar{\mathbf{z}}_j\}_{j \neq i}$ within the same trace

  Compute distance score $S_i = 1 - \bar{\mathbf{z}}_i^\top \bar{\mathbf{z}}_i^{(k)} / (\|\bar{\mathbf{z}}_i\|_2 \|\bar{\mathbf{z}}_i^{(k)}\|_2)$

`// Step 3: Filter noisy steps`

$\mathcal{I}_{\text{drop}} \leftarrow$ indices of the top-$\zeta\%$ steps with the largest scores in $\{S_i\}_{i=1}^{\bar{K}}$

Construct $\bar{\mathbf{C}}^{\text{fil}}$ by retaining $\{\bar{\mathbf{c}}_i \mid i \notin \mathcal{I}_{\text{drop}}\}$ in the original trace order

`// Step 4: Downstream hallucination detection`

$\hat{y} \leftarrow G(\bar{\mathbf{p}}, \bar{\mathbf{C}}^{\text{fil}}, \bar{\mathbf{a}})$

return $\hat{y}$

Table 7 | Effect of training selection ratio ρ (%) on hallucination detection (AUROC, %) on TruthfulQA with Qwen3-8B. The best result in each column is highlighted in **bold**.

ρ (%)	**CCS** [7]	**Supervised Probing** [3]	**Perplexity** [50]	**Verbalized Certainty** [36]
10	82.19	84.06	59.11	54.42
15	84.52	82.29	61.05	57.82
20	**87.32**	83.60	**63.46**	**61.10**
25	87.05	**86.44**	62.74	59.18
30	85.53	85.51	63.02	53.67
35	84.41	86.07	60.31	57.93
40	86.07	84.05	60.85	54.13

B.2. Visualization of loss ablation

Figure 12 provides t-SNE visualizations corresponding to the loss ablation in Section 5.2. In the original embedding space, informative and noisy steps are heavily mixed. The full model produces a compact informative cluster clearly separated from scattered noisy steps, while removing any single loss leads to increased overlap.

B.3. Effect of different distance metrics

We compare alternative distance metrics for the kNN-based filtering in the projected space at test time. Our default method uses cosine distance. We additionally evaluate Mahalanobis distance and Euclidean distance. As shown in Table 8, all three metrics yield strong and comparable results, confirming that the learned projection produces a well-separated score distribution where noisy steps can be reliably identified regardless of the specific distance metric.

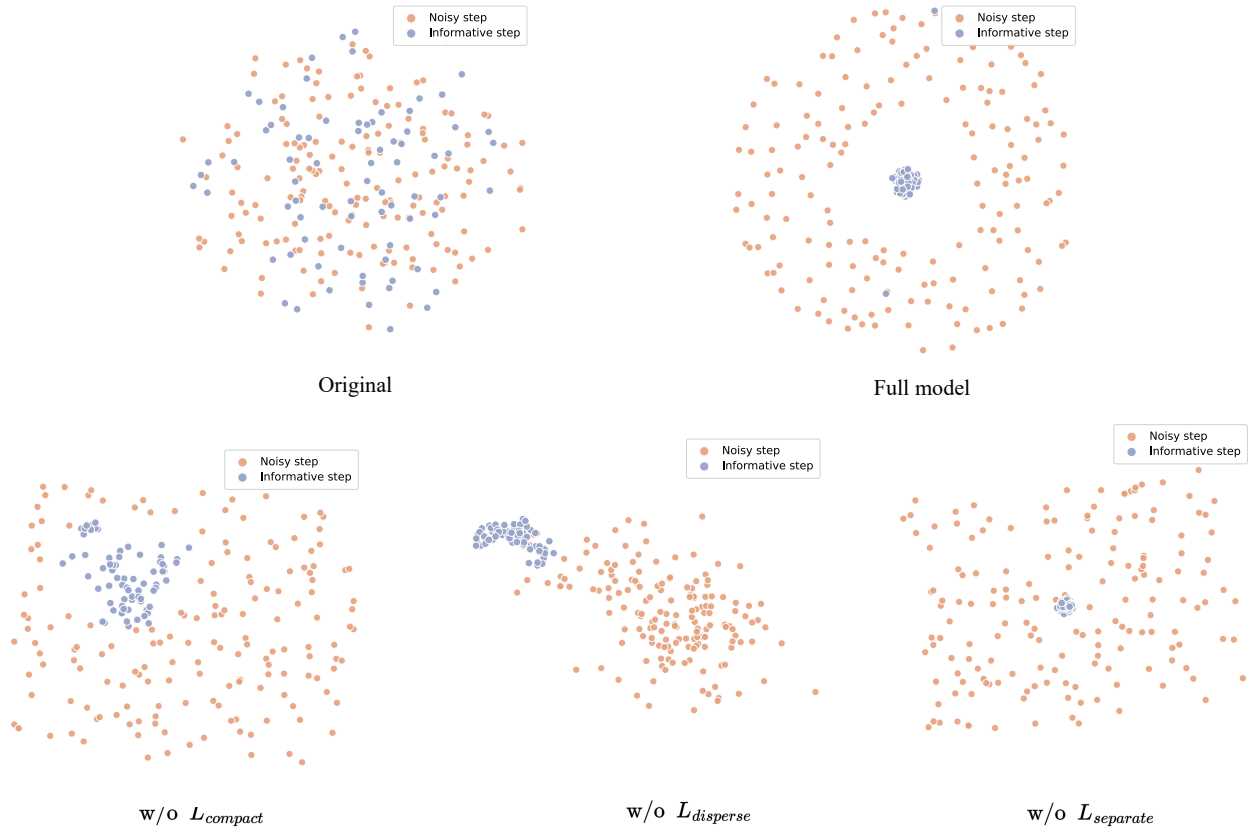

Figure 12 | t-SNE visualization of step embeddings under each ablated loss variant on AIME 2024 [26] with Qwen3-8B. Orange and blue points denote noisy and informative steps, respectively.

Table 8 | Effect of distance metrics for kNN-based filtering on hallucination detection on TruthfulQA with Qwen3-8B using supervised probing.

Distance Metric	**Supervised Probing** [3]
Cosine (Ours)	**86.44**
Mahalanobis	81.51
Euclidean	83.02

B.4. Effect of loss weights

REDE uses two weighting hyperparameters $\lambda_{\text{disperse}}$ and $\lambda_{\text{separate}}$ to balance the three loss components (Eq. 3). We independently vary each weight while fixing the other to its default value, and report detection performance with supervised probing on TruthfulQA with Qwen3-8B in Figure 13. Note that $\lambda = 0$ corresponds to removing the respective loss component entirely. Performance remains stable across a moderate range of both weights, indicating that REDE is not overly sensitive to the precise choice of loss weights. A small weight weakens the corresponding geometric constraint, reducing the quality of the learned projection; a large weight dominates the objective and suppresses the contributions of the other two losses.

B.5. Effect of the number of nearest neighbors k

At inference time, we compute each step's distance score as its k-NN cosine distance with respect to all other steps within the same trace. Table 9 reports detection performance under varying k across four downstream

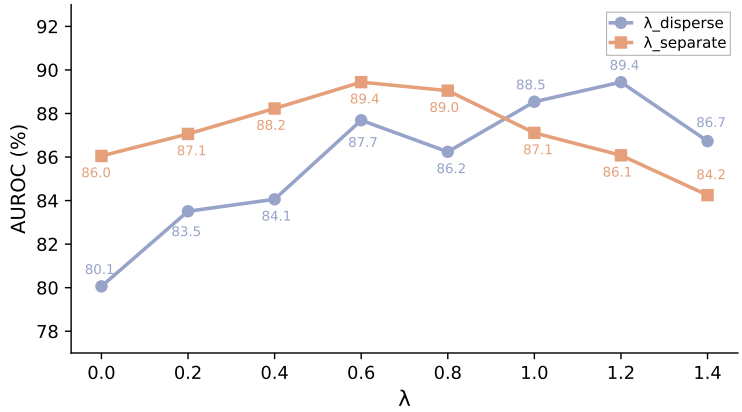

Figure 13 | Effect of loss weights $\lambda_{\text{disperse}}$ and $\lambda_{\text{separate}}$ on hallucination detection on TruthfulQA with Qwen3-8B, evaluated with supervised probing. $\lambda = 0$ indicates the corresponding loss is removed.

detectors. Across all values of k, REDE consistently outperforms the original unfiltered reasoning trace, demonstrating that the learned projection produces a stable density structure that is robust to the choice of k. Performance peaks at $k = 15$ across all four detectors, confirming that this setting best captures the local density signal induced by the learned projection.

Table 9 | Effect of the number of nearest neighbors k on hallucination detection (AUROC, %) on TruthfulQA with Qwen3-8B. The best result in each column is highlighted in **bold**.

k	**CCS** [7]	**Supervised Probing** [3]	**Perplexity** [50]	**Verbalized Certainty** [36]
5	83.97	82.61	60.42	57.13
10	85.34	85.78	61.89	60.27
15	**87.32**	**86.44**	**63.46**	**61.10**
20	86.18	85.39	62.07	59.45
25	85.91	84.02	62.83	58.74
30	84.05	84.71	60.51	58.06

B.6. Effect of step score aggregation

In Eq. 1, we compute the step-level attention score s_i by aggregating attention weights from the final answer token to each reasoning step. We ablate two design choices for this aggregation: the position of the answer token used as the query, and the layer from which the attention is extracted.

Answer token position. We compare four choices: the first token, the middle token (1/2), mean pooling over all answer tokens, and the last token. As shown in Table 10, using the last answer token yields the best performance across all downstream detectors, suggesting that it accumulates the most complete contextual information from the answer sequence and therefore provides the most informative relevance signal for each reasoning step.

Layer. Fixing the query to the last answer token, we further vary the transformer layer from which the attention score is computed. As shown in Figure 14, early layers (0–7) are generally less effective ($\sim 83\%$ AUROC), while intermediate-to-late layers yield stronger results, with local peaks around layer 10, 20, and 22. Layers 27–28 show a mild drop to around 84.5%, after which performance recovers and peaks at the final layer (86.44%). This supports our default choice of computing the step score from the last answer token at the final transformer layer.

Table 10 | Effect of answer token position for step score computation on TruthfulQA with Qwen3-8B. The best result in each column is highlighted in **bold**.

Token Position	CCS [7]	Supervised Probing [3]	Perplexity [50]	Verbalized Certainty [36]
First token	86.17	82.15	54.09	52.96
Middle token (1/2)	85.09	85.35	57.48	55.72
Mean pooling	86.71	86.34	61.30	60.27
Last token (Ours)	**87.32**	**86.44**	**63.46**	**61.10**

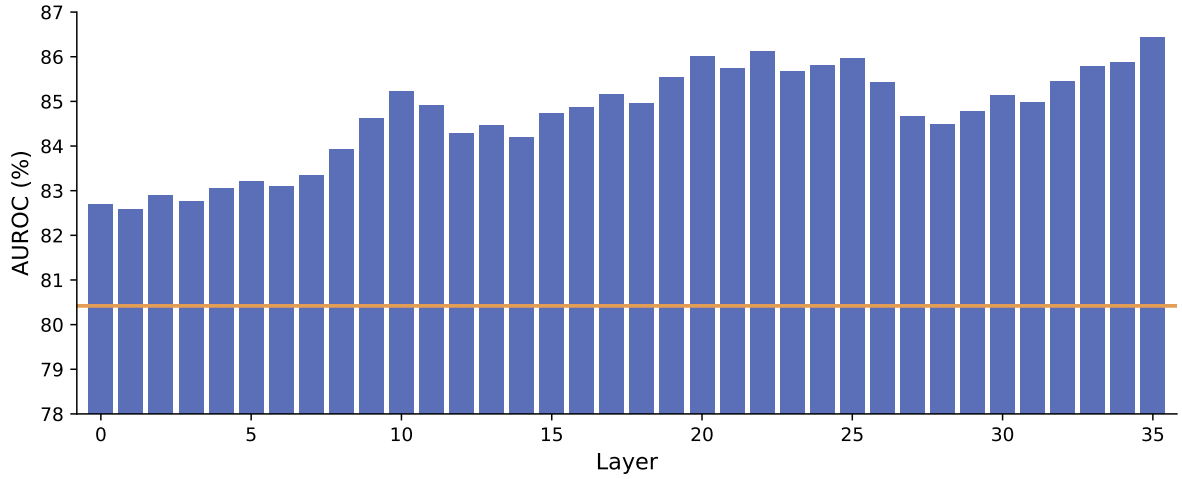

Figure 14 | Effect of the layer used to compute the attention score in Eq. 1. Results are on TruthfulQA with Qwen3-8B, evaluated with supervised probing. The orange horizontal line denotes the performance of using the unfiltered reasoning trace.

B.7. Effect of step embedding weighting

In Eq. 2, we compute the step-level embedding $\mathbf{e}_k$ by applying perplexity (PPL)-weighted averaging over the token hidden states within each step. Specifically, the token-level perplexity is defined as $\mathrm{PPL}(x_{k_j}) = \exp(-\log p_\theta(x_{k_j} \mid x_{<k_j}))$, where $p_\theta(x_{k_j} \mid x_{<k_j})$ is the model's predicted probability for token x_{k_j} given its preceding context. Tokens with higher perplexity receive larger weights, as they are less predictable and therefore more likely to carry discriminative information for hallucination detection. Table 11 compares three step embedding strategies: using only the last token, uniform averaging over all tokens in the step, and PPL-weighted averaging. The results show that PPL-weighting consistently achieves the best performance across all detectors, while the last-token representation is generally stronger than uniform averaging but remains inferior to PPL-weighting.

Table 11 | Effect of step embedding weighting strategy (AUROC, %) on TruthfulQA with Qwen3-8B. The best result in each column is highlighted in **bold**.

Weighting Strategy	CCS [7]	Probing [3]	Perplexity [50]	Verbalized [36]
Uniform average	82.56	85.41	60.88	59.08
Last token	86.18	86.05	63.31	60.09
PPL-weighted (Ours)	**87.32**	**86.44**	**63.46**	**61.10**

B.8. Effect of embedding extraction location

Following prior work [15, 47], we investigate the effect of the multi-head attention (MHA) architecture on step-level representation quality. Specifically, the MHA can be conceptually expressed as:

$$f_{i+1} = f_i + Q_i \,\mathrm{Attn}_i(f_i), \tag{4}$$

where f_i represents the output of the i-th transformer block, $\mathrm{Attn}_i(f_i)$ denotes the output of the self-attention module in the i-th block, and Q_i is the output projection matrix of the attention sub-block. We evaluate hallucination detection performance using step embeddings extracted from three different locations within the MHA architecture, as shown in Table 12. The results show that the block output f is a favorable choice for detecting hallucinations across both LRM architectures.

Table 12 | Effect of embedding extraction location on hallucination detection (AUROC, %) based on supervised probing [3]. The best result in each column is highlighted in **bold**.

Embedding Location	Qwen3-8B		DeepSeek-R1-Distill-Llama-8B	
	TruthfulQA	Math	TruthfulQA	Math
$\mathrm{Attn}(f)$	82.40	84.06	72.08	80.05
$Q\,\mathrm{Attn}(f)$	85.50	83.44	73.18	78.52
f (Ours)	**86.44**	**87.06**	**74.37**	**83.91**

C. Additional analysis

C.1. Qualitative analysis

We present step-level case studies and trace-level embedding visualizations to illustrate how ReDe improves hallucination detection in practice. All reasoning traces are generated by Qwen3-8B [64] on the Math benchmark with greedy decoding, and detection scores are computed using supervised probing [3].

Step-level filtering examples. Figures 15 and 16 show two representative traces after ReDe filtering, where each reasoning step is marked as either kept or filtered. In Figure 15, the model gives an incorrect answer due to an arithmetic error. The original 84-step trace contains substantial repetitive and tangential content, leading to a low detection score of 0.41. After ReDe removes 70% of the steps, the score increases to 0.86, making the error more salient to the detector. Figure 16 shows the complementary case, where the model reaches the correct answer through a verbose but valid derivation. The original trace receives a score of 0.77, incorrectly suggesting hallucination. After ReDe removes 71% of the steps, the score drops to 0.28, reducing this false positive. Together, these examples show that ReDe improves detection for both hallucinated and truthful responses by filtering noisy reasoning steps.

Answer embedding visualization. We further examine whether step-level denoising improves the separability of truthful and hallucinated traces. Figure 17 shows the answer-level embeddings on TruthfulQA with Qwen3-8B. In the original space (left), the two classes are heavily mixed. After ReDe filtering (right), they become more clearly separated, with only limited overlap near the decision boundary. This is consistent with Proposition E.3 in Appendix E, which suggests that removing noisy steps improves downstream discriminability by reducing within-class variance.

C.2. Additional results on larger LRMs

Figure 5(b) demonstrates the scalability of ReDe on TruthfulQA. Here we extend this evaluation to the other three benchmarks. As shown in Table 13, ReDe consistently outperforms two competitive LRM-

Sample 1 — Kept / Filtered

Problem. Tetrahedron $ABCD$: $AB=CD=\sqrt{41}$, $AC=BD=\sqrt{80}$, $BC=AD=\sqrt{89}$. Point I inside equidistant to all faces at distance $\frac{m\sqrt{n}}{p}$. Find $m+n+p$. **GT: 104**

Pred: 102 ✗

S1 Equidistant point = incenter, distance = inradius r.
S2 In a triangle, incenter equidistant from sides.
S3 2D: $r = A/s$ for area A, semiperimeter s.
S4 Incenter splits tet into 4 sub-tetrahedra.
S5 Each has base = face, height = r.
S6 $V = \frac{1}{3}Bh$. Basic formula.
S7 Sub-volumes: $\frac{1}{3}rS_i$.
S8 $V = \frac{r}{3}S_{\text{tot}}$, so $r = 3V/S$.
S9 Need volume V and surface area S.
S10 Opposite edges equal ⇒ disphenoid.
S11 All 4 faces: $\sqrt{41}$, $\sqrt{80}$, $\sqrt{89}$.
S12 $\triangle ABC$: $\sqrt{41}$, $\sqrt{89}$, $\sqrt{80}$.
S13 $\triangle ABD$: same. $\triangle ACD$: same.
S14 $\triangle BCD$: same. Confirmed.
S15 $S = 4\times$area(one face).
S16 Check right triangle? $41, 80, 89$.
S17 $41+80=121\neq 89$. Not right.
S18 $41+80=11^2$, $80+89=13^2$. Unhelpful.
S19 Need other method for area.
S20 Law of cosines at C.
S21 $41 = 169 - 2\sqrt{7120}\cos C$.
S22 $\cos C = 16/\sqrt{445}$.
S23 $\sqrt{7120} = 4\sqrt{445}$. ✓
S24 $\cos^2 C = 256/445 < 1$, acute.
S25 $\sin^2 C = 189/445$.
S26 Area $= 2\sqrt{189}$.
S27 $\sqrt{445}$ cancels in product.
S28 $\sqrt{189} = 3\sqrt{21}$. Area $= 6\sqrt{21}$.
S29 Alt verify: $16A^2 = 12096$. $A^2 = 756$. ✓
S30 $6\sqrt{21} \approx 27.5$. Reasonable.
S31 $S = 4\times 6\sqrt{21} = 24\sqrt{21}$.
S32 Now compute V via Cayley–Menger.
S33 $288V^2 = |\text{CM}|$, 5×5 det.
S34 CM with $AB^2=41, \ldots, CD^2=41$.
S35 Write out full matrix.
S36 Use row reduction.
S37 Row ops: R_2', R_3', R_4'.
S38 Verify R_2'. ✓
S39 Verify R_3'. ✓
S40 Verify R_4'. ✓
S41 CM $= -\det(M)$, 4×4 minor.
S42 Antisymmetry in rows. Disphenoid.
S43 Reduce to 3×3: N.
S44 Verify R_1''. ✓
S45 Verify R_2''. ✓
S46 Verify R_3''. ✓
S47 N symmetric as expected.
S48 Expand $\det(N)$ row 1.
S49 Minor 1: $28480 - 16384 = 12096$.
S50 $160\times 178 = 28480$. ✓
S51 Minor 2: $5696 - 6400 = -704$.
S52 $32\times 178 = 5696$. ✓
S53 Minor 3: $4096 - 8000 = -3904$.
S54 $32\times 128 = 4096$. ✓
S55 $160\times 50 = 8000$. ✓
S56 $\det(N) = -82(12096) - 32(704) + 50(3904)$.
S57 $= -991872 - 22528 + 195200 = -819200$.
S58 Verify sub-products. ✓
S59 $-991872 - 22528 = -1014400$. ✓
S60 CM $= 819200$.
S61 $819200 = 2^{14}\times 50$. Clean.
S62 $V^2 = 25600/9$. $V = 160/3$.
S63 $819200/16 = 51200$, $/18 = 25600/9$. ✓
S64 $\sqrt{25600} = 160$. ✓
S65 $V \approx 53.3$. Plausible.
S66 $r = 3V/S = 160/(24\sqrt{21})$.
S67 $\gcd(160, 24) = 8$: $r = 20/(3\sqrt{21})$.
S68 $160 = 2^5\times 5$, $24 = 2^3\times 3$. ✓
S69 Rationalize: $r = 20\sqrt{21}/(3\times 21) = 20\sqrt{21}/61$.
S70 $3\times 21 = 60 + 1 = 61$.
S71 61 prime. $\gcd(20, 61) = 1$. ✓
S72 $21 = 3\times 7$, square-free. ✓
S73 $m=20, n=21, p=61$. $m+n+p = 102$.
S74 Alt coords: $a=2, b=5/2, c=4$. Same V. ✓

	Steps	Redundancy Ratio	Score	
Original Trace	74	0% filtered	**0.41**	
REDE-filtered	22 (of 74)	70% filtered	**0.86**	+0.45 ↑

Figure 15 | **Sample 1 (incorrect answer).** REDE step-level filtering on a tetrahedron inradius problem from MATH with Qwen3-8B. Green: kept; gray: filtered. Detection scores are from supervised probing [3]. After filtering 70% of the steps, the score increases from 0.41 to 0.86, correctly flagging the hallucinated response.

specific baselines, RACE [59] and ARS [70], across all benchmarks on both Qwen3-32B and DeepSeek-R1-Distill-Qwen-32B. For example, compared with ARS, REDE improves by 14.63%, 4.27%, and 8.41% on MATH, CodeElo, and MULTIHOPQA with Qwen3-32B, respectively. Similar trends hold on DeepSeek-R1-Distill-Qwen-32B, where REDE surpasses ARS by 2.99% on MATH, 3.55% on CodeElo, and 5.47% on MULTIHOPQA. These results confirm that reasoning-step denoising remains effective across diverse reasoning tasks as LRMs scale up.

C.3. Effect of sampling strategies

We evaluate hallucination detection performance when REDE generates reasoning traces under different sampling strategies. Our main results are obtained with greedy decoding, which selects the most likely next token at each step. We additionally compare it with multinomial sampling using a temperature of 0.5. As shown in Table 14, greedy decoding consistently outperforms multinomial sampling across all detectors. The improvement is particularly notable for the Perplexity and Verbalized detectors, suggesting that more deterministic reasoning traces provide more reliable signals for downstream hallucination detection.

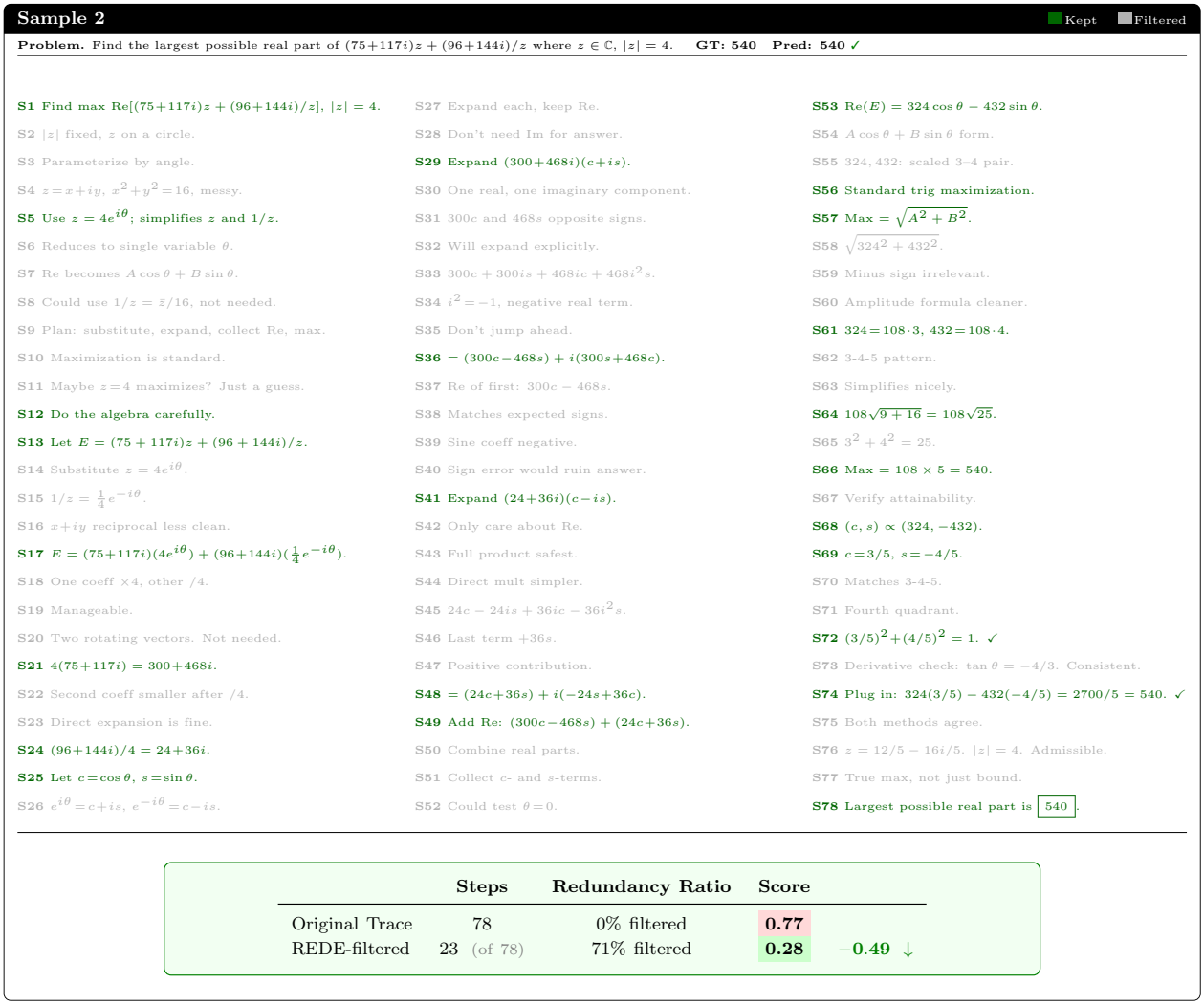

Figure 16 | **Sample 2 (correct answer).** REDE step-level filtering on a complex-number optimization problem from MATH with Qwen3-8B. Green: kept; gray: filtered. Detection scores are from supervised probing [3]. After filtering 71% of the steps, the score decreases from 0.77 to 0.28, correctly reducing the false-positive rate.

C.4. Accuracy of step selection

We further evaluate whether REDE can accurately identify noisy steps. On AIME 2024 [26] with Qwen3-8B, we compare the steps flagged as noisy by REDE against human annotations that categorize each step as informative, irrelevant, or repetitive. REDE achieves an accuracy of 86.37% in classifying steps as informative versus noisy, confirming that the learned projection reliably identifies genuinely noisy steps and that the downstream detection gains stem from accurate step filtering.

C.5. Effect of truthfulness labeling methods

In our main experiments, the correctness of model-generated answers is judged using a strong external model Qwen3-32B. In this ablation, we show that the results remain robust under different judgment methods, including ROUGE-L, BLEURT [52], and a different judge model, DeepSeek-R1-Distill-Qwen-32B [17]. For ROUGE-L, a generation is deemed truthful when the similarity score between the generation and the ground truth exceeds a threshold of 0.3. For BLEURT, we use the `bleurt-base-128` variant, a learned metric built

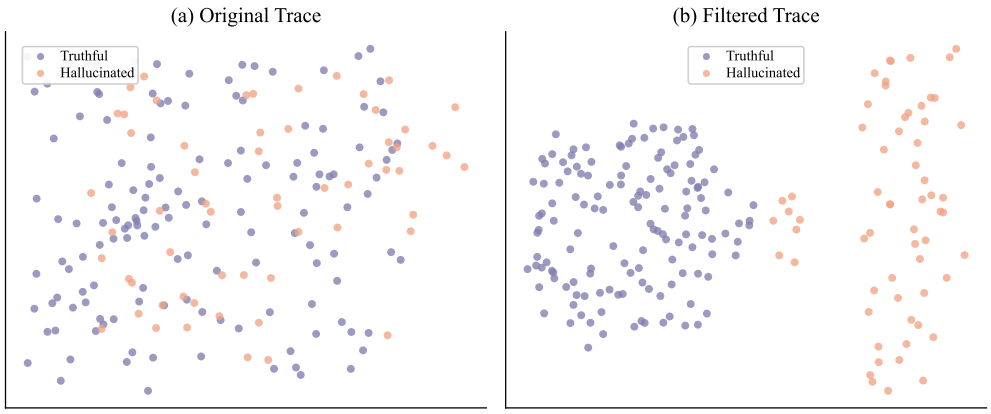

Figure 17 | Answer embeddings on TruthfulQA with Qwen3-8B via supervised probing [3]. Left: original full traces, where truthful and hallucinated responses overlap heavily. Right: after REDE denoising, the two groups become clearly more separable.

Table 13 | Additional results on larger LRMs across three benchmarks. All values are AUROC with supervised probing as the downstream detector. The best result in each column is highlighted in **bold**.

Method	Qwen3-32B			DeepSeek-R1-Distill-Qwen-32B		
	MATH	CodeElo	MULTIHOPQA	MATH	CodeElo	MULTIHOPQA
RACE [59]	77.14	80.96	77.83	67.66	63.72	67.91
ARS [70]	70.09	78.94	73.26	82.44	78.63	70.47
REDE	**84.72**	**83.21**	**81.67**	**85.43**	**82.18**	**75.94**

upon BERT [13] and augmented with diverse lexical and semantic-level supervision signals. As shown in Table 15, REDE consistently achieves strong detection performance across all labeling methods, confirming that our approach does not rely on a specific judge model or evaluation metric.

C.6. Results on an additional dataset

To further evaluate the effectiveness of REDE beyond the datasets used in the main paper, we conduct experiments on GSM8K [11]. For mathematical reasoning, we use the train split, which contains 7,473 problems. Following the same experimental setup as in our main experiments, we reserve 25% of the data for testing, use 100 examples for validation, and use the remaining examples for training. As shown in Table 16, using REDE-selected steps consistently outperforms using the original reasoning trace across all detectors and both model families. The gains are particularly substantial for CCS and remain consistent for probing, perplexity, and verbalized confidence, showing that reasoning-step denoising remains effective on this additional mathematical reasoning dataset.

C.7. Validating the attention signal

We further investigate (i) whether final-answer attention reflects a meaningful notion of step usefulness, and (ii) the advantage of REDE over simple position-based filtering strategies. Experiments are conducted on all four benchmarks with Qwen3-8B, using supervised probing as the downstream detector. All filtering operations use the same drop ratio $\zeta = 70\%$ as REDE.

Final-answer attention reflects step usefulness. We compare three filtering strategies against the original unfiltered trace: dropping the top-70% and the bottom-70% of reasoning steps ranked by their final-answer

Table 14 | Hallucination detection results under different sampling strategies on TruthfulQA with Qwen3-8B. The best result in each column is highlighted in **bold**.

Sampling Strategy	**CCS** [7]	**Probing** [3]	**Perplexity** [50]	**Verbalized** [36]
Multinomial sampling	86.49	85.07	58.11	57.14
Greedy decoding (Ours)	**87.32**	**86.44**	**63.46**	**61.10**

Table 15 | Hallucination detection results (AUROC, %) under different truthfulness labeling methods on TruthfulQA and MATH. For REDE, we report results using probing as the downstream detector. The best result in each column is highlighted in **bold**.

Labeling Method	**ROUGE-L**		**BLEURT**		**DeepSeek-R1-Distill-Qwen-32B**	
	TruthfulQA	MATH	TruthfulQA	MATH	TruthfulQA	MATH
RACE [59]	85.82	77.34	86.21	76.95	65.21	69.02
ARS (CCS) [70]	93.56	81.71	90.18	84.75	80.07	65.49
REDE (Probing)	**95.08**	**83.77**	**92.77**	**85.04**	**85.13**	**78.41**

attention score. For each strategy, we evaluate the cosine similarity between the answer embedding of the remaining trace and that of the full trace, together with the downstream detection AUROC. As shown in Table 17, dropping high-attention steps substantially degrades both metrics compared with the original trace (e.g., AUROC drops from 80.42 to 57.18 on TruthfulQA), whereas dropping low-attention steps preserves and often improves detection quality (e.g., 84.97 vs. 80.42 on TruthfulQA) while maintaining a high cosine similarity to the full-trace answer embedding. This asymmetry shows that final-answer attention reliably captures step-level usefulness for hallucination detection: high-attention steps carry information essential to the final answer, while low-attention steps are largely dispensable.

Comparison with position-based filtering strategies. We compare REDE against two position-based baselines under the same drop ratio: Drop-Latest removes the latest 70% of steps, and Drop-Earliest removes the earliest 70% of steps. As shown in Table 18, Drop-Latest performs worst across all datasets, and while Drop-Earliest is stronger, it still falls well below REDE (e.g., 86.44% vs. 77.53% on TruthfulQA). This demonstrates that our attention-based supervision captures step-level usefulness beyond simple positional patterns, and REDE effectively leverages this signal to identify informative steps throughout the reasoning trace.

C.8. Applicability in a black-box setting

Our main setup assumes white-box access to the target LRM in order to compute final-answer attention and step embeddings. We further study whether REDE can be applied when the target LRM is a black box, by using a separate white-box LRM as a proxy. Specifically, for each input generated by the black-box target, we feed the same prompt and its generated reasoning trace into the proxy to compute the final-answer attention and step embeddings, and then apply REDE to obtain the filtered trace. The filtered trace is passed back to the target model for verbalized certainty estimation, which does not require internal access. We use Qwen3-32B [64] as the proxy and evaluate on two target models: DeepSeek-R1-Distill-Llama-8B [17] and Qwen3-8B [64].

As shown in Table 19, the proxy-based variant consistently improves detection over the original trace across all four benchmarks and both target models, improving over the original by 5.92% on MultiHopQA with DeepSeek and 8.04% on TruthfulQA with Qwen3-8B, recovering most of the gain obtained with white-box REDE. This suggests that REDE remains effective in black-box settings where a suitable proxy LRM is available.

Table 16 | **Comparison of hallucination detection using the original reasoning trace versus REDE-selected steps on GSM8K.** All values are percentages (AUROC). The best results are highlighted in **bold**.

Model	CCS [7]		Probing [3]		Perplexity [50]		Verbalized [36]	
	Original	Filtered	Original	Filtered	Original	Filtered	Original	Filtered
Qwen3-8B	60.08	**91.01**	87.50	**92.44**	55.21	**60.03**	50.07	**60.09**
DeepSeek-R1-Distill-Llama-8B	58.31	**84.22**	78.61	**84.66**	52.31	**58.07**	53.81	**62.30**

Table 17 | Effect of dropping steps ranked by final-answer attention score on Qwen3-8B, compared against the original unfiltered trace. "Cos Sim" denotes the cosine similarity between the answer embedding of the remaining trace and that of the full trace (1.000 for the original trace by definition); AUROC is computed with supervised probing [3].

	TruthfulQA		MATH		CodeElo		MULTIHOPQA	
Method	Cos Sim	AUROC	Cos Sim	AUROC	Cos Sim	AUROC	Cos Sim	AUROC
Original trace (no filtering)	1.000	80.42	1.000	81.05	1.000	82.82	1.000	77.84
Drop top-70% attention	0.427	57.18	0.361	52.84	0.483	61.07	0.394	50.62
Drop bottom-70% attention	0.872	84.97	0.815	83.62	0.846	85.11	0.829	79.48

C.9. Impact of step filtering on downstream task accuracy

Our main experiments measure hallucination detection performance (AUROC). Here we examine a complementary aspect: whether the filtered reasoning trace preserves the LRM's ability to produce correct answers. This evaluation targets *task-level accuracy*, which is distinct from hallucination detection. Concretely, for each reasoning trace generated by Qwen3-8B on MATH, we remove the steps identified as noisy by REDE and regenerate the final answer conditioned on the retained steps. Task accuracy is evaluated using the same judge model as in our main experiments. After filtering, the reasoning traces are reduced by approximately 70% in length. Despite this substantial reduction, the filtered traces achieve an accuracy of 80.00%, compared to 81.43% for the original traces, maintaining a comparable level of task performance. This confirms that the vast majority of discarded steps are uninformative for solving the task, and that REDE's filtering preserves the reasoning capability of the LRM even with a significantly compressed trace.

C.10. Results with additional metrics

Our main experiments use AUROC as the primary evaluation metric, as it measures discrimination ability across all decision thresholds. To provide a more complete evaluation, we additionally report Accuracy and F1 score. As shown in Table 20, using REDE-Filtered steps consistently improves over using the Original reasoning trace across all three metrics for both CCS and probing. Notably, beyond the substantial gains in AUROC, REDE also achieves strong improvements in ACC and F1, further confirming the effectiveness of reasoning-step denoising.

D. Broader impact and limitations

Broader Impact. Large reasoning models (LRMs) are increasingly used in applications where users may rely on the factuality of generated answers. Our work proposes REDE, a lightweight framework that denoises reasoning traces for hallucination detection, and we expect it to have a positive impact on the reliability of LRM-based systems, such as question answering, educational assistants, and coding support. By filtering noisy reasoning steps before downstream detection, REDE can help flag potentially unreliable outputs prior to user-facing deployment. At the same time, REDE is not a guarantee of truthfulness. Like other detection

Table 18 | Comparison of REDE against position-based filtering baselines, all using drop ratio $\zeta = 70\%$ on Qwen3-8B. Drop-Latest removes the latest 70% of steps (retaining the earliest 30%), and Drop-Earliest removes the earliest 70% of steps (retaining the latest 30%). AUROC is computed with supervised probing [3]. The best result in each column is highlighted in **bold**.

Method	TruthfulQA	MATH	CodeElo	MULTIHOPQA
Drop-Latest	68.42	65.71	70.83	66.94
Drop-Earliest	77.53	75.68	79.14	74.29
REDE (Ours)	**86.44**	**87.06**	**87.19**	**82.94**

Table 19 | Applicability of REDE in a black-box setting with Qwen3-32B as the white-box proxy. Detection uses verbalized certainty, which does not require internal access to the target. "REDE (proxy)" uses Qwen3-32B for attention scoring and step embedding, while "REDE (white-box)" has direct access to the target and is shown as an upper-bound reference. All values are AUROC (%). The best result among black-box methods is highlighted in **bold**.

	DeepSeek-R1-Distill-Llama-8B				Qwen3-8B			
Method	TruthfulQA	MATH	CodeElo	MULTIHOPQA	TruthfulQA	MATH	CodeElo	MULTIHOPQA
Original trace	56.61	69.71	79.13	50.87	50.37	57.19	78.89	63.75
REDE (proxy)	**59.85**	**74.26**	**82.07**	**56.79**	**58.41**	**62.83**	**82.14**	**67.62**
REDE (white-box)	61.94	76.13	84.49	59.56	61.10	65.13	83.89	69.50

methods, it may still make incorrect predictions, which could either over-trust hallucinated answers or suppress correct but unconventional reasoning traces. Therefore, we view REDE as a reliability-enhancing component rather than a substitute for human oversight, especially in high-stakes settings.

Limitations. REDE currently operates as a filtering module on reasoning traces. It does not intervene during the generation process itself. A promising direction for future work is to integrate the denoising signal into the LRM, for example by incorporating the step-level informativeness objective into model training so that the LRM learns to suppress noisy steps during generation.

Table 20 | **Hallucination detection results under additional evaluation metrics on TruthfulQA with Qwen3-8B.** We compare using the Original reasoning trace versus REDE-Filtered steps. The best result in each group is highlighted in **bold**.

Setting	CCS [7]			Probing [3]		
	AUROC	ACC	F1	AUROC	ACC	F1
Original	68.63	58.91	65.03	80.42	66.57	63.08
Filtered	**87.32**	**70.47**	**73.95**	**86.44**	**71.09**	**68.82**

E. Theoretical analysis

In this section, we provide a theoretical perspective on why REDE improves hallucination detection. Our analysis follows the practical inference rule in Section 4.3, where steps are filtered by their intra-trace κ-NN distances in the shaped space. At a high level, we formalize the following story: (1) the oracle informative trace provides a more discriminative representation for hallucination detection than the full trace containing noisy steps; (2) after representation shaping, informative steps become locally dense while noisy steps remain locally sparse, which separates their κ-NN distance scores; (3) under rank-based removal of the top-$\zeta\%$ largest κ-NN distances, the resulting filtering errors are small; and (4) small filtering errors imply that the hallucination detection risk on the filtered trace is close to that on the oracle informative trace.

To keep the analysis analytically tractable, we introduce in Assumption E.1 an ε-approximate form of conditional label-uninformativeness of the noisy perturbation, parameterized by two scalar quantities $(\varepsilon_\mu, \varepsilon_c)$. Proposition E.3 is correspondingly stated as an additive perturbation bound whose slack decays continuously with $(\varepsilon_\mu, \varepsilon_c)$ and vanishes in the strict-independence limit, so that our bounds remain meaningful across a broad range of noisy-step behaviors.

E.1. Setup and definitions

Consider a test-time prompt-answer pair $(\bar{\mathbf{p}}, \bar{\mathbf{a}})$ with reasoning trace $\bar{\mathbf{C}} = \{\bar{\mathbf{c}}_1, \ldots, \bar{\mathbf{c}}_{\bar{K}}\}$, following the test-time notation in Section 4.3. Let $\bar{\mathbf{C}}^{\text{info}} \subseteq \bar{\mathbf{C}}$ denote the subset of informative reasoning steps, and let $\bar{\mathbf{C}}^{\text{noise}} = \bar{\mathbf{C}} \setminus \bar{\mathbf{C}}^{\text{info}}$ denote the subset of noisy steps. At test time, REDE produces a filtered trace $\bar{\mathbf{C}}^{\text{fil}} \subseteq \bar{\mathbf{C}}$ by removing steps that are identified as noisy in the learned representation space.

Let $\boldsymbol{\psi}(\bar{\mathbf{c}}) \in \mathbb{R}^{d_\Phi}$ denote the representation of reasoning step $\bar{\mathbf{c}}$ used by the downstream hallucination detector, and let the trace-level representation be

$$\boldsymbol{\Phi}(\bar{\mathbf{p}}, \bar{\mathbf{C}}, \bar{\mathbf{a}}) = \boldsymbol{\Phi}_0(\bar{\mathbf{p}}, \bar{\mathbf{a}}) + \frac{1}{\bar{K}} \sum_{\bar{\mathbf{c}} \in \bar{\mathbf{C}}} \boldsymbol{\psi}(\bar{\mathbf{c}}), \tag{5}$$

where $\boldsymbol{\Phi}_0(\bar{\mathbf{p}}, \bar{\mathbf{a}})$ captures the prompt-answer component independent of the reasoning steps. This form covers common downstream detectors that operate on pooled hidden states or other additive trace summaries. We normalize by the original trace length $\bar{K}$ so that filtered and oracle traces are compared under a common scale.

Let $g : \mathbb{R}^{d_\Phi} \to \mathbb{R}$ be a downstream detector, and let $\ell(g(\boldsymbol{\Phi}), y)$ be the detection loss for truthfulness label $y \in \{0, 1\}$. We define the population risk on a trace subset $\bar{\mathbf{C}}' \subseteq \bar{\mathbf{C}}$ as

$$\mathcal{R}(g; \bar{\mathbf{C}}') = \mathbb{E}\big[\ell(g(\boldsymbol{\Phi}(\bar{\mathbf{p}}, \bar{\mathbf{C}}', \bar{\mathbf{a}})), y)\big], \tag{6}$$

where the expectation is taken over the data distribution.

To quantify the quality of filtering, we define two error rates:

$$\alpha := \mathbb{E}\left[\frac{\left|\bar{\mathbf{C}}^{\text{fil}} \cap \bar{\mathbf{C}}^{\text{noise}}\right|}{\bar{K}}\right], \tag{7}$$

$$\beta := \mathbb{E}\left[\frac{\left|\bar{\mathbf{C}}^{\text{info}} \setminus \bar{\mathbf{C}}^{\text{fil}}\right|}{\bar{K}}\right]. \tag{8}$$

Here, α measures the fraction of noisy steps that are erroneously retained, while β measures the fraction of informative steps that are mistakenly removed; both expectations are taken over the data distribution, including the randomness of the trace length $\bar{K}$.

Let $\bar{\mathbf{e}}_i$ denote the pre-projection embedding of step $\bar{\mathbf{c}}_i$, and let

$$\bar{\mathbf{z}}_i = f_\phi(\bar{\mathbf{e}}_i) \in \mathbb{R}^{d'} \tag{9}$$

be the projected step embedding produced by the learned projection head f_ϕ. Consistent with the practical inference rule in Section 4.3 and the cosine-similarity-based training objective in Eq. 3, throughout this appendix we instantiate the distance over the projected space as the cosine distance

$$D(\mathbf{u}, \mathbf{v}) := 1 - \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|_2 \, \|\mathbf{v}\|_2}, \tag{10}$$

matching the score S_i used at inference in Section 4.3. To match this practical inference rule, we define the step-level filtering score by the intra-trace κ-NN cosine distance

$$S_i := D(\bar{\mathbf{z}}_i, \bar{\mathbf{z}}_i^{(\kappa)}), \tag{11}$$

where $\bar{\mathbf{z}}_i^{(\kappa)}$ denotes the κ-th nearest neighbor of $\bar{\mathbf{z}}_i$ among $\{\bar{\mathbf{z}}_j\}_{j \neq i}$ within the same trace under D. We use κ here to denote the nearest-neighbor parameter and reserve $\bar{K}$ for the number of steps in the test trace; correspondingly, the number of nearest neighbors denoted by k in Section 4.3 is written as κ throughout this appendix.

Following the practical rank-based rule, let

$$M := \lceil \zeta \bar{K} \rceil \tag{12}$$

denote the number of removed steps, where $\zeta \in (0, 1)$ is the removal ratio. The filtered trace $\bar{\mathbf{C}}^{\text{fil}}$ is obtained by removing the M steps with the largest scores $\{S_i\}_{i=1}^{\bar{K}}$.

For a radius $r > 0$, define the local neighbor count around step i by

$$N_i(r) := \sum_{j \neq i} \mathbf{1}\{D(\bar{\mathbf{z}}_j, \bar{\mathbf{z}}_i) \leq r\}. \tag{13}$$

Then the κ-NN score admits the equivalent characterization

$$S_i \leq r \iff N_i(r) \geq \kappa, \qquad S_i > r \iff N_i(r) < \kappa. \tag{14}$$

E.2. Oracle informative traces are more discriminative

We first formalize why noisy reasoning steps are harmful for hallucination detection. The key intuition is that noisy steps act as approximately label-uninformative perturbations to the trace representation: they do

not substantially improve the class-conditional mean separation relevant for truthfulness prediction, but they increase the within-class variability, thereby reducing class separability.

Let

$$\mathbf{\Phi}^\star := \mathbf{\Phi}(\bar{\mathbf{p}}, \bar{\mathbf{C}}^{\text{info}}, \bar{\mathbf{a}}), \qquad \boldsymbol{\delta} := \frac{1}{\bar{K}} \sum_{\bar{\mathbf{c}} \in \bar{\mathbf{C}}^{\text{noise}}} \boldsymbol{\psi}(\bar{\mathbf{c}}),$$

so that the full-trace representation satisfies

$$\mathbf{\Phi}(\bar{\mathbf{p}}, \bar{\mathbf{C}}, \bar{\mathbf{a}}) = \mathbf{\Phi}^\star + \boldsymbol{\delta}. \tag{15}$$

For each label $y \in \{0, 1\}$, define the class-conditional means and covariances

$$\boldsymbol{\mu}_y^\star := \mathbb{E}[\mathbf{\Phi}^\star \mid y], \qquad \mathbf{\Sigma}_y^\star := \mathrm{Cov}(\mathbf{\Phi}^\star \mid y),$$

and

$$\boldsymbol{\mu}_y := \mathbb{E}[\mathbf{\Phi} \mid y], \qquad \mathbf{\Sigma}_y := \mathrm{Cov}(\mathbf{\Phi} \mid y).$$

Let the pooled within-class covariance matrices be

$$\mathbf{\Sigma}_w^\star := \pi_0 \mathbf{\Sigma}_0^\star + \pi_1 \mathbf{\Sigma}_1^\star, \qquad \mathbf{\Sigma}_w := \pi_0 \mathbf{\Sigma}_0 + \pi_1 \mathbf{\Sigma}_1,$$

where $\pi_y := \mathbb{P}(Y = y)$.

Assumption E.1 (ε-approximate label-uninformative perturbation)**.** *There exist constants $\varepsilon_\mu, \varepsilon_c \geq 0$ such that for each $y \in \{0, 1\}$:*

(i) (Approximate mean-zero) $\left\|\mathbb{E}[\boldsymbol{\delta} \mid y]\right\|_2 \leq \varepsilon_\mu$;
(ii) (Bounded cross-covariance) $\left\|\mathrm{Cov}(\mathbf{\Phi}^\star, \boldsymbol{\delta} \mid y)\right\|_{\text{op}} \leq \varepsilon_c$;
(iii) $\mathrm{Cov}(\boldsymbol{\delta} \mid y) = \mathbf{\Sigma}_\delta(y) \succeq \mathbf{0}$, $\mathbf{\Sigma}_w^\star \succ \mathbf{0}$, *and the cross-covariance slack is moderate relative to the informative covariance, i.e.,* $\varepsilon_c \leq \lambda_{\min}(\mathbf{\Sigma}_w^\star)/4$.

The strict-independence case $\boldsymbol{\delta} \perp \mathbf{\Phi}^\star \mid y$ together with $\mathbb{E}[\boldsymbol{\delta} \mid y] = \mathbf{0}$ is recovered as the special limit $\varepsilon_\mu = \varepsilon_c = 0$.

Remark E.2 (Scope and tightness of Assumption E.1)**.** *Assumption E.1 captures a broad class of noisy-step behaviors through two nonnegative slack parameters $(\varepsilon_\mu, \varepsilon_c)$, which control, respectively, the class-conditional mean of the noisy perturbation $\boldsymbol{\delta}$ and its second-order correlation with the informative component $\mathbf{\Phi}^\star$. The mild magnitude condition on ε_c in (iii) ensures that the perturbed within-class covariance $\mathbf{\Sigma}_w$ remains well-conditioned relative to its oracle counterpart $\mathbf{\Sigma}_w^\star$, which is needed for the matrix-inverse perturbation bound used in the proof of Proposition E.3. The strict-independence case corresponds to the limit $\varepsilon_\mu = \varepsilon_c = 0$, while general $(\varepsilon_\mu, \varepsilon_c)$ allow arbitrary nonzero slack within the stated bounds. Proposition E.3 below is an additive perturbation bound whose slack $\Delta(\varepsilon_\mu, \varepsilon_c)$ decays continuously with these parameters, so the tightness of the bound scales smoothly with how close the noisy perturbation is to being label-uninformative.*

We quantify class separability of a trace representation by the Fisher discriminability

$$\mathcal{J}(\bar{\mathbf{C}}') := \big(\boldsymbol{\mu}_1(\bar{\mathbf{C}}') - \boldsymbol{\mu}_0(\bar{\mathbf{C}}')\big)^\top \mathbf{\Sigma}_w(\bar{\mathbf{C}}')^{-1} \big(\boldsymbol{\mu}_1(\bar{\mathbf{C}}') - \boldsymbol{\mu}_0(\bar{\mathbf{C}}')\big), \tag{16}$$

where

$$\boldsymbol{\mu}_y(\bar{\mathbf{C}}') := \mathbb{E}[\mathbf{\Phi}(\bar{\mathbf{p}}, \bar{\mathbf{C}}', \bar{\mathbf{a}}) \mid y], \qquad \mathbf{\Sigma}_w(\bar{\mathbf{C}}') := \pi_0 \mathbf{\Sigma}_0(\bar{\mathbf{C}}') + \pi_1 \mathbf{\Sigma}_1(\bar{\mathbf{C}}').$$

Proposition E.3 (Noisy steps reduce trace discriminability, approximate version)**.** *Under Assumption E.1, the full-trace Fisher discriminability is bounded by the oracle one up to a slack controlled by* $(\varepsilon_\mu, \varepsilon_c)$*:*

$$\mathcal{J}(\bar{\mathbf{C}}) \le \mathcal{J}(\bar{\mathbf{C}}^{\text{info}}) + \Delta(\varepsilon_\mu, \varepsilon_c), \tag{17}$$

where

$$\Delta(\varepsilon_\mu, \varepsilon_c) := \frac{4\,\varepsilon_\mu\,\|\boldsymbol{\mu}_1^\star - \boldsymbol{\mu}_0^\star\|_2 + 4\,\varepsilon_\mu^2}{\lambda_{\min}(\boldsymbol{\Sigma}_w^\star)} + \frac{4\,\varepsilon_c\,\big(\|\boldsymbol{\mu}_1^\star - \boldsymbol{\mu}_0^\star\|_2 + 2\varepsilon_\mu\big)^2}{\lambda_{\min}(\boldsymbol{\Sigma}_w^\star)^2}. \tag{18}$$

In particular, $\Delta(\varepsilon_\mu, \varepsilon_c) \to 0$ *as* $\varepsilon_\mu, \varepsilon_c \to 0$*, recovering the exact monotonicity* $\mathcal{J}(\bar{\mathbf{C}}) \le \mathcal{J}(\bar{\mathbf{C}}^{\text{info}})$ *in the strict-independence limit.*

Proof. By Eq. 15 and Assumption E.1(i),

$$\boldsymbol{\mu}_y = \boldsymbol{\mu}_y^\star + \mathbb{E}[\boldsymbol{\delta} \mid y] = \boldsymbol{\mu}_y^\star + \boldsymbol{\eta}_y, \qquad \|\boldsymbol{\eta}_y\|_2 \le \varepsilon_\mu,$$

so that

$$\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|_2 \le \|\boldsymbol{\mu}_1^\star - \boldsymbol{\mu}_0^\star\|_2 + 2\varepsilon_\mu.$$

For the covariances, write

$$\boldsymbol{\Sigma}_y = \boldsymbol{\Sigma}_y^\star + \boldsymbol{\Sigma}_\delta(y) + \mathbf{E}_y, \qquad \mathbf{E}_y := \mathrm{Cov}(\boldsymbol{\Phi}^\star, \boldsymbol{\delta} \mid y) + \mathrm{Cov}(\boldsymbol{\delta}, \boldsymbol{\Phi}^\star \mid y),$$

with $\|\mathbf{E}_y\|_{\text{op}} \le 2\varepsilon_c$ by Assumption E.1(ii). Therefore

$$\boldsymbol{\Sigma}_w = \boldsymbol{\Sigma}_w^\star + \bar{\boldsymbol{\Sigma}}_\delta + \bar{\mathbf{E}}, \qquad \bar{\boldsymbol{\Sigma}}_\delta := \pi_0\boldsymbol{\Sigma}_\delta(0) + \pi_1\boldsymbol{\Sigma}_\delta(1) \succeq \mathbf{0}, \quad \|\bar{\mathbf{E}}\|_{\text{op}} \le 2\varepsilon_c.$$

Let $\mathbf{A} := \boldsymbol{\Sigma}_w^\star + \bar{\boldsymbol{\Sigma}}_\delta \succeq \boldsymbol{\Sigma}_w^\star \succ \mathbf{0}$, so $\boldsymbol{\Sigma}_w = \mathbf{A} + \bar{\mathbf{E}}$. By operator monotonicity of inversion on the positive-definite cone, $\mathbf{A}^{-1} \preceq (\boldsymbol{\Sigma}_w^\star)^{-1}$. Using the standard perturbation bound for matrix inversion [22],

$$\big\|\boldsymbol{\Sigma}_w^{-1} - \mathbf{A}^{-1}\big\|_{\text{op}} \le \frac{\|\bar{\mathbf{E}}\|_{\text{op}}}{\lambda_{\min}(\mathbf{A})\,(\lambda_{\min}(\mathbf{A}) - \|\bar{\mathbf{E}}\|_{\text{op}})}.$$

Since $\lambda_{\min}(\mathbf{A}) \ge \lambda_{\min}(\boldsymbol{\Sigma}_w^\star)$ and $\|\bar{\mathbf{E}}\|_{\text{op}} \le 2\varepsilon_c \le \lambda_{\min}(\boldsymbol{\Sigma}_w^\star)/2$ by Assumption E.1(iii), we have $\lambda_{\min}(\mathbf{A}) - \|\bar{\mathbf{E}}\|_{\text{op}} \ge \lambda_{\min}(\boldsymbol{\Sigma}_w^\star)/2$, and therefore

$$\big\|\boldsymbol{\Sigma}_w^{-1} - \mathbf{A}^{-1}\big\|_{\text{op}} \le \frac{2\varepsilon_c}{\lambda_{\min}(\boldsymbol{\Sigma}_w^\star)\cdot\lambda_{\min}(\boldsymbol{\Sigma}_w^\star)/2} = \frac{4\varepsilon_c}{\lambda_{\min}(\boldsymbol{\Sigma}_w^\star)^2}.$$

Combining the two bounds,

$$\begin{aligned}
\mathcal{J}(\bar{\mathbf{C}}) &= (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_w^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \\
&\le (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \mathbf{A}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) + \big\|\boldsymbol{\Sigma}_w^{-1} - \mathbf{A}^{-1}\big\|_{\text{op}}\,\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|_2^2 \\
&\le (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top (\boldsymbol{\Sigma}_w^\star)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) + \frac{4\varepsilon_c\,(\|\boldsymbol{\mu}_1^\star - \boldsymbol{\mu}_0^\star\|_2 + 2\varepsilon_\mu)^2}{\lambda_{\min}(\boldsymbol{\Sigma}_w^\star)^2}.
\end{aligned}$$

For the first term, writing $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0 = (\boldsymbol{\mu}_1^\star - \boldsymbol{\mu}_0^\star) + (\boldsymbol{\eta}_1 - \boldsymbol{\eta}_0)$ with $\|\boldsymbol{\eta}_1 - \boldsymbol{\eta}_0\|_2 \le 2\varepsilon_\mu$, expanding the quadratic form, and using $\|(\boldsymbol{\Sigma}_w^\star)^{-1}\|_{\text{op}} \le 1/\lambda_{\min}(\boldsymbol{\Sigma}_w^\star)$,

$$(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top (\boldsymbol{\Sigma}_w^\star)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \le \mathcal{J}(\bar{\mathbf{C}}^{\text{info}}) + \frac{4\varepsilon_\mu\,\|\boldsymbol{\mu}_1^\star - \boldsymbol{\mu}_0^\star\|_2 + 4\varepsilon_\mu^2}{\lambda_{\min}(\boldsymbol{\Sigma}_w^\star)}.$$

Summing the two contributions recovers Eq. 17 with $\Delta(\varepsilon_\mu, \varepsilon_c)$ as in Eq. 18. □

Proposition E.3 shows that, up to a slack vanishing with $(\varepsilon_\mu, \varepsilon_c)$, noisy reasoning steps do not improve Fisher discriminability and typically degrade it by inflating within-class variation. This motivates recovering a trace close to the oracle informative trace before performing hallucination detection.

E.3. κ-NN distance separation from local density in the projected space

We next analyze the practical score in Eq. 11. The key quantity is the local neighbor count in Eq. 13: informative steps should have many nearby neighbors within a small radius, whereas noisy steps should have few nearby neighbors even within a larger radius.

Assumption E.4 (Local density separation)**.** *Conditioned on a center step $\bar{\mathbf{c}}_i$ and its type, the random indicators*

$$\mathbf{1}\{D(\bar{\mathbf{z}}_j, \bar{\mathbf{z}}_i) \le r\}, \qquad j \neq i,$$

are independent Bernoulli variables for each fixed radius r. Moreover, there exist radii $0 < r_{\mathcal{I}} < r_{\mathcal{N}}$ such that

$$\mu_{\mathcal{I}} := \mathbb{E}\Big[N_i(r_{\mathcal{I}}) \mid \bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{info}}\Big] \ge 2\kappa, \tag{19}$$

and

$$\mu_{\mathcal{N}} := \mathbb{E}\big[N_i(r_{\mathcal{N}}) \mid \bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{noise}}\big] \le \kappa/2. \tag{20}$$

Here, the non-bold $\mu_{\mathcal{I}}, \mu_{\mathcal{N}}$ denote scalar expectations of the local neighbor count, and should not be confused with the class-conditional trace means $\boldsymbol{\mu}_y$ used earlier. This assumption formalizes the geometry induced by REDE: informative steps form a compact local neighborhood, while noisy steps remain sparse and isolated.

Proposition E.5 (κ-NN distances separate informative and noisy steps)**.** *Under Assumption E.4, the practical κ-NN score in Eq. 11 satisfies*

$$\mathbb{P}\Big(S_i > r_{\mathcal{I}} \;\Big|\; \bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{info}}\Big) \le \exp(-\kappa/4), \tag{21}$$

and

$$\mathbb{P}\big(S_i \le r_{\mathcal{N}} \;\big|\; \bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{noise}}\big) \le \exp(-\kappa/6). \tag{22}$$

Proof. By Eq. 14,

$$S_i > r_{\mathcal{I}} \iff N_i(r_{\mathcal{I}}) < \kappa.$$

For informative steps, Assumption E.4 gives

$$\mu_{\mathcal{I}} = \mathbb{E}[N_i(r_{\mathcal{I}}) \mid \bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{info}}] \ge 2\kappa.$$

Therefore,

$$\{N_i(r_{\mathcal{I}}) < \kappa\} \subseteq \{N_i(r_{\mathcal{I}}) < \mu_{\mathcal{I}}/2\}.$$

Applying the multiplicative Chernoff bound to the lower tail,

$$\mathbb{P}\Big(N_i(r_{\mathcal{I}}) < \mu_{\mathcal{I}}/2 \;\Big|\; \bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{info}}\Big) \le \exp(-\mu_{\mathcal{I}}/8) \le \exp(-\kappa/4),$$

which proves Eq. 21.

Similarly, by Eq. 14,

$$S_i \le r_{\mathcal{N}} \iff N_i(r_{\mathcal{N}}) \ge \kappa.$$

For noisy steps, Assumption E.4 gives

$$\mu_{\mathcal{N}} = \mathbb{E}[N_i(r_{\mathcal{N}}) \mid \bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{noise}}] \le \kappa/2.$$

The event $\{N_i(r_{\mathcal{N}}) \ge \kappa\}$ is an upper-tail event at least as extreme as doubling the mean. Applying the multiplicative Chernoff bound to the upper tail yields

$$\mathbb{P}\big(N_i(r_{\mathcal{N}}) \ge \kappa \;\big|\; \bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{noise}}\big) \le \exp(-\kappa/6),$$

which proves Eq. 22. □

Proposition E.5 shows that once the learned projection creates a sufficiently dense informative region and a sufficiently sparse noisy region, informative and noisy steps become separable directly under the practical κ-NN distance used at inference time.

E.4. From κ-NN score separation to rank-based filtering error

We now analyze the practical rank-based rule that removes the top-$\zeta\%$ largest κ-NN distances within a trace.

Let

$$\bar{K}_{\mathcal{N}} := |\bar{\mathbf{C}}^{\text{noise}}| \tag{23}$$

denote the number of noisy steps in the trace, and define the score-separation event

$$\mathcal{E}_{\text{sep}} := \left\{ \max_{\bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{info}}} S_i \le r_{\mathcal{I}}, \quad \min_{\bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{noise}}} S_i \ge r_{\mathcal{N}} \right\}. \tag{24}$$

Since $r_{\mathcal{I}} < r_{\mathcal{N}}$, on $\mathcal{E}_{\text{sep}}$ every noisy step has a larger score than every informative step.

Proposition E.6 (Rank-based filtering error under practical κ-NN scores)**.** *On the event $\mathcal{E}_{\text{sep}}$, removing the top-M steps ranked by $\{S_i\}_{i=1}^{\bar{K}}$ yields*

$$\left|\bar{\mathbf{C}}^{\text{fil}} \cap \bar{\mathbf{C}}^{\text{noise}}\right| = (\bar{K}_{\mathcal{N}} - M)_+, \qquad \left|\bar{\mathbf{C}}^{\text{info}} \setminus \bar{\mathbf{C}}^{\text{fil}}\right| = (M - \bar{K}_{\mathcal{N}})_+, \tag{25}$$

where $(x)_+ := \max\{x, 0\}$. Consequently,

$$\alpha \le \mathbb{E}\left[\frac{(\bar{K}_{\mathcal{N}} - M)_+}{\bar{K}}\right] + \mathbb{P}(\mathcal{E}_{\text{sep}}^c), \tag{26}$$

$$\beta \le \mathbb{E}\left[\frac{(M - \bar{K}_{\mathcal{N}})_+}{\bar{K}}\right] + \mathbb{P}(\mathcal{E}_{\text{sep}}^c). \tag{27}$$

Proof. On $\mathcal{E}_{\text{sep}}$, all noisy steps are ranked ahead of all informative steps by the practical κ-NN scores. If $M \le \bar{K}_{\mathcal{N}}$, then the top-$M$ removed steps are all noisy. Hence exactly M noisy steps are removed, no informative steps are removed, and the number of retained noisy steps is $\bar{K}_{\mathcal{N}} - M$. If $M > \bar{K}_{\mathcal{N}}$, then all noisy steps are removed and the remaining $M - \bar{K}_{\mathcal{N}}$ removed steps must be informative. This proves Eq. 25.

Now decompose

$$\frac{|\bar{\mathbf{C}}^{\text{fil}} \cap \bar{\mathbf{C}}^{\text{noise}}|}{\bar{K}} = \frac{|\bar{\mathbf{C}}^{\text{fil}} \cap \bar{\mathbf{C}}^{\text{noise}}|}{\bar{K}} \mathbf{1}\{\mathcal{E}_{\text{sep}}\} + \frac{|\bar{\mathbf{C}}^{\text{fil}} \cap \bar{\mathbf{C}}^{\text{noise}}|}{\bar{K}} \mathbf{1}\{\mathcal{E}_{\text{sep}}^c\}.$$

On $\mathcal{E}_{\text{sep}}$, Eq. 25 gives

$$\frac{|\bar{\mathbf{C}}^{\text{fil}} \cap \bar{\mathbf{C}}^{\text{noise}}|}{\bar{K}} = \frac{(\bar{K}_{\mathcal{N}} - M)_+}{\bar{K}}.$$

On $\mathcal{E}_{\text{sep}}^c$, the normalized count is at most 1. Taking expectations yields Eq. 26. The proof of Eq. 27 is identical using the second identity in Eq. 25. □

The bounds in Eqs. 26–27 separate two sources of error: (1) *count mismatch*, because the practical rule removes exactly $M = \lceil \zeta \bar{K} \rceil$ steps while the true number of noisy steps is $\bar{K}_{\mathcal{N}}$; and (2) *score overlap*, captured by $\mathbb{P}(\mathcal{E}_{\text{sep}}^c)$.

Corollary E.7 (Filtering error under local density separation)**.** *Under Assumption E.4,*

$$\mathbb{P}(\mathcal{E}_{\text{sep}}^{c}) \le \mathbb{E}[\bar{K}]\,\big(\exp(-\kappa/4) + \exp(-\kappa/6)\big). \tag{28}$$

Consequently,

$$\alpha \le \mathbb{E}\left[\frac{(\bar{K}_{\mathcal{N}} - M)_+}{\bar{K}}\right] + \mathbb{E}[\bar{K}]\,\big(\exp(-\kappa/4) + \exp(-\kappa/6)\big), \tag{29}$$

$$\beta \le \mathbb{E}\left[\frac{(M - \bar{K}_{\mathcal{N}})_+}{\bar{K}}\right] + \mathbb{E}[\bar{K}]\,\big(\exp(-\kappa/4) + \exp(-\kappa/6)\big). \tag{30}$$

Proof. Conditioned on a trace of length $\bar{K}$, a union bound and Proposition E.5 together with the trivial bounds $|\bar{\mathbf{C}}^{\text{info}}| \le \bar{K}$ and $|\bar{\mathbf{C}}^{\text{noise}}| \le \bar{K}$ give

$$\begin{aligned}\mathbb{P}(\mathcal{E}_{\text{sep}}^{c} \mid \bar{K}) &\le \sum_{\bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{info}}} \mathbb{P}\Big(S_i > r_{\mathcal{I}} \mid \bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{info}}\Big) \\ &\quad + \sum_{\bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{noise}}} \mathbb{P}\big(S_i \le r_{\mathcal{N}} \mid \bar{\mathbf{c}}_i \in \bar{\mathbf{C}}^{\text{noise}}\big) \\ &\le \bar{K}\,\big(\exp(-\kappa/4) + \exp(-\kappa/6)\big).\end{aligned}$$

Taking expectation over $\bar{K}$ on both sides yields Eq. 28. Substituting Eq. 28 into Proposition E.6 yields Eqs. 29–30. □

E.5. Representation gap induced by imperfect filtering

We now bound the representation distortion caused by imperfect filtering, where some noisy steps may be retained and some informative steps may be lost.

Assumption E.8 (Bounded step representations)**.** *There exist constants $B_{\text{info}}, B_{\text{noise}} > 0$ such that*

$$\|\boldsymbol{\psi}(\bar{\mathbf{c}})\|_2 \le B_{\text{info}} \quad \textit{for all } \bar{\mathbf{c}} \in \bar{\mathbf{C}}^{\text{info}}, \qquad \|\boldsymbol{\psi}(\bar{\mathbf{c}})\|_2 \le B_{\text{noise}} \quad \textit{for all } \bar{\mathbf{c}} \in \bar{\mathbf{C}}^{\text{noise}}.$$

Lemma E.9 (Bounded representation gap)**.** *Let*

$$\boldsymbol{\Phi}^{\text{fil}} := \boldsymbol{\Phi}(\bar{\mathbf{p}}, \bar{\mathbf{C}}^{\text{fil}}, \bar{\mathbf{a}}), \qquad \boldsymbol{\Phi}^{\star} := \boldsymbol{\Phi}(\bar{\mathbf{p}}, \bar{\mathbf{C}}^{\text{info}}, \bar{\mathbf{a}}).$$

Under Assumption E.8, the expected representation gap between the filtered and oracle traces satisfies

$$\mathbb{E}\big[\|\boldsymbol{\Phi}^{\text{fil}} - \boldsymbol{\Phi}^{\star}\|_2\big] \le B_{\text{noise}}\,\alpha + B_{\text{info}}\,\beta. \tag{31}$$

Proof. Using the additive representation in Eq. 5, the difference decomposes as

$$\boldsymbol{\Phi}^{\text{fil}} - \boldsymbol{\Phi}^{\star} = \frac{1}{\bar{K}}\left(\sum_{\bar{\mathbf{c}} \in \bar{\mathbf{C}}^{\text{fil}} \cap \bar{\mathbf{C}}^{\text{noise}}} \boldsymbol{\psi}(\bar{\mathbf{c}}) - \sum_{\bar{\mathbf{c}} \in \bar{\mathbf{C}}^{\text{info}} \setminus \bar{\mathbf{C}}^{\text{fil}}} \boldsymbol{\psi}(\bar{\mathbf{c}})\right). \tag{32}$$

Taking norms and applying the triangle inequality and Assumption E.8,

$$\|\boldsymbol{\Phi}^{\text{fil}} - \boldsymbol{\Phi}^{\star}\|_2 \le \frac{B_{\text{noise}}}{\bar{K}}\big|\bar{\mathbf{C}}^{\text{fil}} \cap \bar{\mathbf{C}}^{\text{noise}}\big| + \frac{B_{\text{info}}}{\bar{K}}\big|\bar{\mathbf{C}}^{\text{info}} \setminus \bar{\mathbf{C}}^{\text{fil}}\big|. \tag{33}$$

Taking expectation on both sides and using the definitions of α and β in Eqs. 7–8 yields Eq. 31. □

E.6. Detection risk on filtered traces

Following common practice in representation analysis [1, 3, 70], we analyze the detection risk under a linear probe on the trace representation. We now show that a small representation gap implies that the filtered trace achieves hallucination detection risk close to that of the oracle informative trace.

Assumption E.10 (Linear probe with Lipschitz loss)**.** *The downstream detector is a linear probe, i.e., $g(\mathbf{\Phi}) = \mathbf{w}^\top \mathbf{\Phi} + b$ for some $\mathbf{w} \in \mathbb{R}^{d_\Phi}$ and $b \in \mathbb{R}$, and the loss $\ell(\cdot, y)$ is L_0-Lipschitz in its first argument. Consequently, the composed loss $\ell(g(\cdot), y)$ is Lipschitz with respect to the trace representation with constant $L := L_0\|\mathbf{w}\|_2$:*

$$|\ell(g(\mathbf{u}), y) - \ell(g(\mathbf{v}), y)| \le L\|\mathbf{u} - \mathbf{v}\|_2 \qquad \text{for all } \mathbf{u}, \mathbf{v} \in \mathbb{R}^{d_\Phi}.$$

Theorem E.11 (Risk bound on filtered traces)**.** *Under Assumptions E.8 and E.10, for any linear probe g satisfying Assumption E.10,*

$$\mathcal{R}(g; \bar{\mathbf{C}}^{\text{fil}}) \le \mathcal{R}(g; \bar{\mathbf{C}}^{\text{info}}) + L\big(B_{\text{noise}}\,\alpha + B_{\text{info}}\,\beta\big). \tag{34}$$

Moreover, under Assumptions E.4, E.8 and E.10,

$$\begin{aligned}\mathcal{R}(g; \bar{\mathbf{C}}^{\text{fil}}) \le \mathcal{R}(g; \bar{\mathbf{C}}^{\text{info}}) &+ LB_{\text{noise}}\left(\mathbb{E}\left[\frac{(\bar{K}_{\mathcal{N}} - M)_+}{\bar{K}}\right] + \mathbb{E}[\bar{K}]\left(e^{-\kappa/4} + e^{-\kappa/6}\right)\right) \\ &+ LB_{\text{info}}\left(\mathbb{E}\left[\frac{(M - \bar{K}_{\mathcal{N}})_+}{\bar{K}}\right] + \mathbb{E}[\bar{K}]\left(e^{-\kappa/4} + e^{-\kappa/6}\right)\right).\end{aligned} \tag{35}$$

Proof. By the definition of population risk,

$$\begin{aligned}\mathcal{R}(g; \bar{\mathbf{C}}^{\text{fil}}) - \mathcal{R}(g; \bar{\mathbf{C}}^{\text{info}}) &= \mathbb{E}\big[\ell(g(\mathbf{\Phi}^{\text{fil}}), y) - \ell(g(\mathbf{\Phi}^\star), y)\big] \\ &\le \mathbb{E}\big[\big|\ell(g(\mathbf{\Phi}^{\text{fil}}), y) - \ell(g(\mathbf{\Phi}^\star), y)\big|\big] \\ &\le L\,\mathbb{E}\big[\|\mathbf{\Phi}^{\text{fil}} - \mathbf{\Phi}^\star\|_2\big],\end{aligned} \tag{36}$$

where the last step applies Assumption E.10. Applying Lemma E.9 gives Eq. 34. If Assumption E.4 additionally holds, substituting the bounds from Corollary E.7 for α and β yields Eq. 35. □

Remark E.12 (Role of $(\varepsilon_\mu, \varepsilon_c)$ in the overall risk bound)**.** *Theorem E.11 bounds the excess risk of the filtered trace relative to the* oracle informative *risk $\mathcal{R}(g; \bar{\mathbf{C}}^{\text{info}})$, and does not itself depend on $(\varepsilon_\mu, \varepsilon_c)$. The slack parameters enter upstream through Proposition E.3, which controls the gap between $\mathcal{R}(g; \bar{\mathbf{C}})$ (using the full, unfiltered trace) and $\mathcal{R}(g; \bar{\mathbf{C}}^{\text{info}})$ via $\Delta(\varepsilon_\mu, \varepsilon_c)$. Combining the two yields, for a problem-dependent constant $C > 0$ relating Fisher discriminability to Bayes-optimal linear-probe risk,*

$$\mathcal{R}(g; \bar{\mathbf{C}}^{\text{fil}}) \le \mathcal{R}(g; \bar{\mathbf{C}}) + L\big(B_{\text{noise}}\alpha + B_{\text{info}}\beta\big) + C\,\Delta(\varepsilon_\mu, \varepsilon_c).$$

This makes the role of the slack parameters explicit: the smaller $(\varepsilon_\mu, \varepsilon_c)$ and the smaller the filtering errors (α, β), the larger the improvement of the filtered trace over the unfiltered one.

Corollary E.13 (Consistency under accurate practical filtering)**.** *Under Assumptions E.8 and E.10, if*

$$\alpha \to 0 \qquad \text{and} \qquad \beta \to 0,$$

then

$$\mathcal{R}(g; \bar{\mathbf{C}}^{\text{fil}}) \to \mathcal{R}(g; \bar{\mathbf{C}}^{\text{info}}).$$

Moreover, under Assumptions E.4, E.8 and E.10, if

$$\mathbb{E}\left[\frac{(\bar{K}_{\mathcal{N}} - M)_+}{\bar{K}}\right] \to 0, \qquad \mathbb{E}\left[\frac{(M - \bar{K}_{\mathcal{N}})_+}{\bar{K}}\right] \to 0,$$

and

$$\mathbb{E}[\bar{K}]\left(e^{-\kappa/4}+e^{-\kappa/6}\right)\to 0,$$

then

$$\mathcal{R}(g;\bar{\mathbf{C}}^{\mathrm{fil}})\to\mathcal{R}(g;\bar{\mathbf{C}}^{\mathrm{info}}).$$

Proof. The first claim follows immediately from Theorem E.11. For the second claim, Corollary E.7 implies $\alpha\to 0$ and $\beta\to 0$ under the stated conditions. Applying the first claim completes the proof. □

Theorem E.11 shows that the excess hallucination detection risk incurred by using the filtered trace is directly controlled by the quality of practical κ-NN-based filtering. Combined with Proposition E.3, Proposition E.5, and Proposition E.6, it gives a complete chain: the learned projection creates informative regions that are locally dense and noisy regions that are locally sparse; this local density gap separates the practical κ-NN scores; under rank-based removal of the largest scores, the resulting filtering errors are small; and small filtering errors imply that the filtered trace achieves risk close to that of the oracle informative trace.

E.7. Connection to the shaping objective

The formal results above explain why filtering quality matters. We now discuss how the shaping objective in Eq. 3 of the main paper supports the practical κ-NN filtering rule.

Remark E.14 (Effect of the shaping loss on practical κ-NN filtering)**.** *The training objective in Eq. 3 is defined on the attention-based proxy sets $\mathcal{T}$ and $\mathcal{B}$. For interpretation, we consider the population-level approximation that these proxies are reasonably aligned with the latent informative and noisy step sets $\bar{\mathbf{C}}^{\mathrm{info}}$ and $\bar{\mathbf{C}}^{\mathrm{noise}}$.*

Under this approximation, the compactness term $\mathcal{L}_{\mathrm{compact}}$ pulls projected embeddings of proxy-informative steps together, which increases the local neighbor count $N_i(r)$ for informative steps at small radii and therefore tends to decrease their practical κ-NN distances. The dispersion term $\mathcal{L}_{\mathrm{disperse}}$ discourages projected embeddings of proxy-noisy steps from forming dense neighborhoods, which decreases their local neighbor count and therefore tends to increase their practical κ-NN distances. The separation term $\mathcal{L}_{\mathrm{separate}}$ pushes the two groups apart, reducing cross-group neighborhood overlap and making the score-separation event $\mathcal{E}_{\mathrm{sep}}$ more likely.

Together, these terms shape a projected embedding space in which informative steps occupy compact high-density neighborhoods while noisy steps remain sparse and isolated. Proposition E.5 then implies improved separation under the practical κ-NN score, and Proposition E.6 together with Theorem E.11 implies lower filtering error and tighter hallucination detection risk. The ablation study in Section 5.2 provides empirical support for this interpretation.