arXiv:2602.01378v1 [cs.CL] 1 Feb 2026

自回归语言模型中的上下文依赖性与可靠性。

Poushali Sengupta 奥斯陆大学信息学系 奥斯陆，挪威 poushals@ifi.uio.no Shashi Raj Pandey 奥尔堡大学电子系统系 奥尔堡，丹麦 srp@es.aau.dk Sabita Maharjan 奥斯陆大学信息学系 奥斯陆，挪威 sabita@ifi.uio.no Frank Eliassen 奥斯陆大学信息学系 奥斯陆，挪威 frank@ifi.uio.no

2026年2月3日

摘要

大语言模型（LLMs）通过以广泛的上下文为条件来生成输出，这些上下文通常包含冗余指令、重复事实、检索段落和交互历史。在关键应用中，必须识别哪些上下文元素——提示、检索结果、先前交互和记忆——真正被用于生成输出，以及模型的推理如何中介它们的影响。这对于构建可信人工智能系统至关重要。然而，注意力、梯度或扰动等标准解释方法难以处理冗余，尤其是当记忆引入重叠或过时信息时。即使是微小的输入变化，例如改写过去的指令或重排序检索内容，也可能在不改变输出的情况下使归因分数发生不可预测的偏移，从而削弱在富含记忆和历史的场景中的可解释性。这种敏感性不仅削弱了解释的可靠性，也引发了对提示注入等新兴风险的担忧：微妙的上下文操纵可以在看似合法的先前内容表象下劫持模型行为。在我们的研究中，我们探讨了与自回归大语言模型可信性相关的一个关键问题：区分对输出生成至关重要的上下文元素与仅与之相关联的上下文元素的能力。我们提出，依赖感知——明确考量每个上下文元素相对于其他元素的贡献——对于在大语言模型中生成可靠解释至关重要。为实现这一点，我们提出了RISE（冗余不敏感解释评分），用于量化每个输入元素在其余上下文条件下的独特影响。该设计自然地抑制冗余贡献，并产生更清晰的归因。在冗余提示和检索增强生成（RAG）上的实验表明，与传统基于扰动的方法相比，RISE能产生显著更稳定、更稳健的解释。这些发现凸显了条件信息作为可信解释以及在推理过程中监控大语言模型行为的原则性且高效基础的作用。

引言

大语言模型是理解和生成人类语言的强大工具。它们越来越多地应用于医疗决策、法律建议以及辅助完成各种任务的自动化系统等重要领域 Bommasani 等人（2021）；OpenAI（2023）。从本质上说，大语言模型通过预测下一个词（或

词元）来运作，预测依据是不断扩展的上下文，其中包括系统指令、用户问题、先前对话以及其他资源 Brown 等人（2020）。随着这些模型日益复杂、获得更多自主权，确保它们可理解且值得信赖至关重要 Doshi-Velez 和 Kim（2017）。阐明这些挑战有助于让受众认识到，他们的专业知识对于解决信任问题至关重要。然而，解释这些模型如何得出结论并非易事。当前的方法，例如可视化模型关注哪些词，或分析调整输入后输出发生的变化，往往会得出不一致的结果 Wiegreffe 和 Pinter（2019）；Sundararajan 等人（2017）。其中一个原因是，大语言模型处理的长上下文中往往会重复相似信息。例如，指令可能会被换一种说法重述，文档可能含义相近，先前的讨论也可能被反复提及 Liu 等人（2024）。即使整体回答保持不变，输入结构的变化也可能导致解释大不相同。这种不一致会削弱信任，并使监控和评估模型性能的过程更加复杂 Jacovi 和 Goldberg（2021）。一个简单的现实类比是向几个人问路，有些人重复相同的建议，另一些人则提出相互矛盾的建议。设想有人问如何去一家咖啡馆：

- **人物 A**说：“直走两个街区，然后左转。”
- **人物 B** 重复了相同的建议：“对——直走，然后左转。”
- **人物 C** 说了不同的话：“直走，然后在第二个路口右转。”

这个人仍然左转了，依据的是 A 提供的信息（以及 B 中冗余的信息）。现在假设后来有人问：“哪条建议导致了左转？”如果某个解释声称*A*、*B* 和 *C* 对这一决策的影响完全相同，那显然是错误的。B 并未提供超出 A 之外的新信息，而 C 则与之矛盾。因此，问题不在于冗余本身，而在于**错误归因**：它可能造成一种错误印象，让人以为多个独立来源都是必要的，甚至暗示错误的建议也促成了正确的行动。更糟糕的是，如果 C 的指令被记录下来并在之后被重复（例如作为“记忆”），审计员可能会错误地得出结论，认为“右转”属于推理过程，或者仅仅因为解释系统将其视为有影响因素，就认为遵循该指令是安全的。相比之下，不受冗余影响的解释会采取不同的处理方式。由于 B 没有提供超出 A 之外的新信息，它应获得接近于零的归因，而解释应主要将左转归因于 A。此外，**C 也应因另一种原因获得（近似的）零归因**：在其余上下文给定的情况下，它无助于解释左转。一旦 A（以及 B）已经明确指示“左转”，C 相互冲突的指令就不会为观察到的行动提供额外的预测信息；在以 A 为条件的情况下，该决策已经确定，因此 C 不具有独特贡献。换言之，C 不仅仅是冗余的——它与该决策无关（或呈负向一致），因此，依赖感知解释器应当抑制它，而不是因其存在而给予奖励。

这种区别在实践中很重要。首先，它能避免产生“许多来源意见一致”的错觉：将重复出现的陈述都计入贡献，可能会错误地让人以为结果稳健，尽管这些信息不过是被重复了 Jacovi & Goldberg（2020）。其次，它能提升稳定性：即使输出保持不变，对冗余敏感的归因也可能因细微的释义或重排序而发生变化，而依赖感知归因则能保持一致 Kindermans 等人（2019）。第三，它能降低对抗风险：如果恶意指令被注入检索上下文或记忆中，随后又被重复，忽视冗余的解释器就可能过度归功于这些指令，给受操纵的行为赋予不应有的正当性 Wei 等人（2023）。简而言之，抑制冗余并不会移除关键信息；它能防止重要性随着重复次数增加，从而生成体现独特依赖而非出现频率的解释。更广泛地说，当多个上下文元素传达重叠信息时，将贡献分配给各项输入的归因方法可能系统性地高估冗余内容的价值。这可能会**误导**用户，使其无法正确理解模型的运作方式，并造成对可靠性或可控性的错误印象 Miller（2019）。在实践中，这会降低可审计性（真实的决策路径被掩盖）、误导信任（解释看似稳定，实际却很脆弱），并在提示注入或记忆投毒情境下增加风险（无关内容被赋予不可忽视的影响力） Liet al.（2023）。

表 1：自回归大语言模型的解释方法比较。

方法	约简-感知	稳定	上下文-层级	推理-time
注意力可视化	×	×	×	✓
梯度	×	×	×	✓
显著性	×	×	✓	×
扰动	✓	✓	×	×
机制探查				
RISE（本文）	✓	✓	✓	✓

大语言模型在清晰解释其决策过程方面面临重大挑战。一个主要问题是，很难区分哪些内容对特定结果真正至关重要，哪些只是与之同时出现。在许多情况下，提示、检索段落和记忆信息可能会重复或改述相同的事实。如果某种解释方法对每一次重复都赋予同等重要性，就可能错误地暗示模型利用了多种独立信号，甚至更糟的是，暗示其结论基于无关或错误的信息。这

可能导致三种　具体失效模式：1. 过度归功于干扰项：当模型因　r

其实际上并未真正依赖的信息而受到赞扬时，就会出现这种情况，从而掩盖其输出的真正来源。2. 不稳定性：细微的变化，例如释义改写或调整信息顺序，都可能导致归因显著变化，而模型的输出实际上并未发生改变。3. 易受操纵性：这在提示注入等场景中尤为值得关注，因为插入上下文或记忆中的有害指令可能会不当地影响模型输出，这是因为所使用的解释方法无法区分真实依赖与信息仅仅在位置上的接近。当前的方法，例如基于注意力的分数 Jain & Wallace（2019）；Serrano & Smith（2019）、梯度归因 Sundararajan 等人（2017）；Simonyan 等人（2014），以及简单的消融或扰动技术 Li 等人（2016）；Feng 等人（2018），都难以应对这些问题。它们往往过于关注数据的表面形式，未能充分考虑语义关系或不同上下文信息片段之间的条件依赖。为应对如何为自回归语言模型提供可靠解释这一挑战，我们提出了一种结构化方法，优先考虑我们所称的“依赖感知”。这一原则为在自回归系统中生成可信的归因奠定了基础。我们的贡献是一种名为**RISE（冗余不敏感解释评分）**的新解释方法。该方法在考虑其他元素影响的同时，评估每个上下文元素的独特贡献。通过降低重叠或无关输入的影响，RISE 旨在提供更清晰、更准确的解释。进而，用户和评估者能够辨明模型真正依赖的内容，增强其抵御误解和潜在操纵的能力。

本研究的核心观点是，应从条件依赖的角度评估自回归语言模型的解释稳定性：只有当上下文元素提供了无法从其余上下文中恢复的下一词元分布相关信息时，才应将其归因于该元素。通过对提示和检索上下文进行受控变化，我们表明，这一方法能够减少冗余导致的重要性高估，并揭示现有面向稳定性的启发式方法的局限性。此外，我们的方法支持结构化上下文单元，例如提示片段、检索段落或对话轮次，从而能够在推理过程中切实开展实时监控和审计，追踪模型对上下文的依赖。本研究的主要贡献如下：

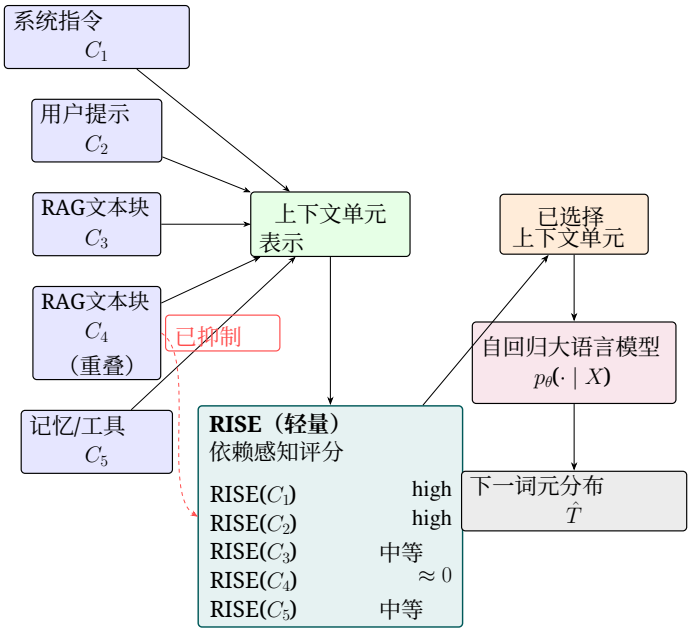

图1：使用**RISE**的轻量级依赖感知上下文设计。结构化上下文单元被映射为紧凑表示，并根据其在其余上下文条件下的独特贡献进行评分，冗余单元（例如重叠的RAG片段）在归因和生成之前被抑制。

- 我们将上下文冗余识别为解释自回归大语言模型时的一种关键失效模式：当未考虑重叠输入时，归因可能看似合理甚至稳定，但仍会将功劳归于冗余或不相关的上下文，从而掩盖模型的真实依赖关系。
- 我们提出以条件信息作为自回归模型中可靠归因的原则性基础，并引入**RISE**作为一种依赖感知评分方法，用于评估结构化上下文单元（例如提示片段、检索段落和对话轮次）的重要性。
- 通过受控的提示词与检索实验，我们证明依赖感知归因通过减少冗余导致的过度归功，产生更清晰、更有意义的解释，并为监控、调试和审计大语言模型行为提供可操作信号，包括在类似提示注入的上下文操纵情形下。

我们有意将主稿聚焦于核心的依赖感知原则及其经验后果，同时将辅助诊断、压力测试和各变体的分析留待附录，以保持研讨会论文叙事清晰。

背景、相关工作与问题缺口

上一节指出，自回归大语言模型中的解释不稳定性主要源于长上下文中未建模的依赖关系与冗余。现在，我们将这一观察置于大语言模型可解释性相关研究的背景下，并正式阐述促使我们采用该方法的尚存研究空白。自回归语言模型根据以下内容生成文本，

$$p(x_1, \ldots, x_T) = \prod_{t=1}^{T} p(x_t \mid x_{<t}), \tag{1}$$

其中，每个词元都以所有先前词元为条件 Brown 等人（2020）。在现代部署中，条件上下文 $x_{<t}$ 并非扁平序列，而是由异质元素构成的结构化组合，包括系统指令、用户提示、检索增强生成（RAG）中的检索文档、对话历史、中间推理步骤、工具输出和智能体记忆 Bommasani 等人（2021）；Lewis 等人（2020）。我们将这种结构化上下文表示为一组上下文单元，$\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$，其中每个 C_i 都代表一个具有语义意义的单元。这种抽象反映了提示在实践中的构建和使用方式，也提供了一个自然的粒度，使解释能够在部署期间接受审计、比较并据此采取行动。研究日益聚焦于提升大语言模型（LLM）的可解释性 Doshi- Velez 和 Kim（2017）；Miller（2019）。当前解释大语言模型的方法主要分为四类，包括基于注意力的解释：这类方法将注意力权重可视化，但可能产生误导，因为仅凭注意力无法捕捉上下文元素之间的所有影响 Wiegreffe 和 Pinter（2019）。基于梯度的方法衡量输出对输入变化的敏感性，但可能受到模型重新参数化的影响，也难以分离相互关联输入的贡献 Sundararajan 等人（2017）。基于扰动的方法通过遮蔽或移除词元后观察输出变化来评估词元重要性，但这可能导致对长上下文中的重要性估计过高 Hooker（2021）。机制分析探索模型内部表征，但往往计算开销巨大，不适用于实时场景 Olah 等人（2018）。为应对这些挑战，我们提出了一种评估上下文贡献的新方法，并明确考虑上下文元素之间的依赖关系，从而实现更准确、高效的解释。我们的方法包含 **RISE**，这是一种面向结构化上下文单元的轻量级、模型无关解释工具；此外还提供一个可选的学习型选择器，可在模型中纳入依赖感知。

通过 RISE 进行依赖感知解释设计

令 $\mathcal{C} = \{C_1, \ldots, C_m\}$ 表示提供给自回归大语言模型的一组 结构化上下文单元。每个单元C_i 对应提示中的一个语义连贯组成部分，例如系统指令块、用户查询、上下文示例、检索增强生成（RAG）中的检索文档块（Lewis 等人（2020））、对话轮次、工具输出或智能体记忆条目。令 X 表示按照固定模板将 $\mathcal{C}$ 序列化后得到的拼接提示。这种抽象反映了实践中大语言模型上下文的构建和操作方式（Brown 等人（2020）），并提供了一个自然的层次，使得部署期间能够使用、审查解释并据此采取行动。我们考虑多个解释目标。主要目标是下一词元分布 $\hat{T} \triangleq p_\theta(\cdot \mid X)$。必要时，同一公式也适用于 (i) 采样得到的下一词元 $\hat{t} \sim p_\theta(\cdot \mid X)$，以及 (ii) 生成的文本片段 $\hat{S} = (\hat{t}_1, \ldots, \hat{t}_H)$；后者对于智能体系统中的长时程监控尤为重要（Yao 等人（2023））。[1] 我们的目标是量化每个上下文单元 C_i在考虑其余上下文 $\mathcal{C}_{\setminus i}$ 后，对生成结果贡献了多少独特的信息。我们将 C_i 的未归一化依赖感知贡献定义为其与解释目标之间的**条件唯一依赖（CUD）**（形式上即条件互信息）：

$$\Delta_i \triangleq \mathrm{CUD}(C_i; \hat{T} \mid \mathcal{C}_{\setminus i}) \triangleq I(C_i; \hat{T} \mid \mathcal{C}_{\setminus i}). \tag{2}$$

该量衡量 C_i 所提供的信息中，无法从其余上下文中恢复的部分 Cover 与 Thomas（2006）。为了便于比较不同上下文单元的贡献，我们将这些贡献归一化，得到我们的 **RISE 评分**（冗余不敏感解释评分）：

$$\mathrm{RISE}(C_i) = \frac{\mathrm{CUD}(C_i; \hat{T} \mid \mathcal{C}_{\setminus i})}{\sum_{j=1}^{m} \mathrm{CUD}(C_j; \hat{T} \mid \mathcal{C}_{\setminus j}) + \varepsilon}, \tag{3}$$

其中，$\varepsilon > 0$ 是一个较小的常数，仅用于在退化情况下避免除以零（例如，所有 CUD 的数值估计均为零时）。在实践中，我们将 ε 设为可忽略的值（例如，10^{-12}），并明确报告分母接近零的情况。

[1] 如果跨度级生成是通过迭代采样产生的，那么 $\hat{S}$ 可以视为固定上下文 X下自回归展开所诱导的联合随机变量。

对于生成的片段 $\hat{}S$，我们类似地定义 $\Delta_i^{(S)} \triangleq \mathrm{CUD}(C_i; \hat{}S \mid \mathcal{C}_{\setminus i})$，并按照方程 3 进行归一化。这使得我们能够在句子级或工具调用级进行解释和监控。较高的 RISE(C_i) 值表明，某个上下文单元提供了关于生成结果的信息，而这些信息无法从提示的其余部分中恢复；较低的值则表明这些信息与 $\mathcal{C}_{\setminus i}$ 存在冗余。重要的是，RISE 并非旨在解释内部神经元、注意力头或机制因果路径。相反，它在结构化上下文单元层面刻画了条件依赖下的模型依赖。这种视角非常适合可信部署：它能够提供在提示冗余情况下保持稳定、从业者可直接理解，并可用于推理时监控、审计和干预的解释。

理论性质

现在，我们将形式化若干关键理论性质，以证明 **RISE** 作为一种针对在长且冗余的上下文上运行的自回归语言模型、遵循原则且具备依赖感知能力的解释设计是合理的。在全文中，我们假设所有随机变量都定义在同一个概率空间上，并且条件互信息 $\hat{}$ 已有明确定义。令 $\mathcal{C} = \{C_1, \ldots, C_m\}$ 表示结构化上下文单元的集合，令 T表示解释目标（例如，下一词元分布或生成词元）。对于每个单元 C_i，定义其条件贡献 $\Delta_i \triangleq I(C_i; \hat{}T \mid \mathcal{C}_{\setminus i})$，以及其归一化归因分数，

$$\mathrm{RISE}(C_i) = \frac{\Delta_i}{\sum_{j=1}^{m} \Delta_j + \varepsilon}. \tag{4}$$

如果某个上下文单元在已有信息之外没有增加任何新信息，那么依赖感知解释应将其重要性设为零。

引理1（条件独立下的冗余抑制）。 如果上下文单元 C_i 满足 $C_i \perp\!\!\!\perp \hat{}T \mid \mathcal{C}_{\setminus i}$，则 $\Delta_i = 0$，因此对于 $\varepsilon = 0$ 和 $\sum \Delta_j > 0$，$\mathrm{RISE}(C_i) = 0$。$_j$

一旦已知所有其他上下文，若某个上下文元素无法为模型输出提供独特信息，就应获得零归因。重复相同的指令或上下文不应增加其重要性，因为这并未引入任何额外信息。

表2： 边际归因方法与**RISE** 在自回归大语言模型上下文中的概念比较。

方面	边际归因	RISE（本文方法）
处理冗余	No	是（通过条件化）
提示变化下的稳定性	Low	High
上下文单元解释	有限	原生
推理时监控	困难	直接

推论 1（重复上下文不变性）。 假设两个上下文单元 C_i 和 C_i' 完全重复，即 $C_i = C_i'$ 几乎必然成立。那么，在理想的总体情形下，$I(C_i; \hat{}T \mid \mathcal{C}_{\setminus i}) = 0$ 和 $I(C_i'; \hat{}T \mid \mathcal{C}_{\setminus i'}) = 0$，且两个单元的***RISE***归因均为零。

重复相同的提示或指令并不会使其变得更重要；**RISE** 会正确忽略重复上下文。添加冗余上下文不应改变分配给现有信息性上下文单元的重要性。

命题 1（冗余注入下的稳定性）。 设 $\mathcal{C}' = \mathcal{C} \cup \{C'\}$，其中 C' 在给定 $\mathcal{C}$的情况下是冗余的，即$C' \perp\!\!\!\perp \hat{}T \mid \mathcal{C}$. 那么，对于所有原有单元 $C_i \in \mathcal{C}$，$\mathrm{RISE}_{\mathcal{C}'}(C_i) = \mathrm{RISE}_{\mathcal{C}}(C_i)$，在理想的总体情形下。

注入冗余上下文不会扭曲对原始输入的解释；**RISE** 在设计上保持归因稳定性。其他性质、扩展及压力测试结果留待附录。**RISE**提供依赖感知归因，但同一条件信息原理也可用于设计轻量级上下文选择器，以降低自回归系统的有效输入维度。关键观察是，自回归上下文往往高度冗余：在时间序列中，许多历史滞后项编码了重叠的季节性/惯性信息；在大语言模型中，提示片段和检索片段经常在语义上进行释义或重叠。全部输入

表 3： 轻量级依赖感知上下文选择的面向部署的优势（强制采用近期窗口 + 个唯一锚点）。

设计成果	在部署中实现的功能
缩减上下文规模（$L + K \ll m$）	通过消除.....降低延迟和方差 冗余上下文单元，同时保留关键信息 。
依赖感知	保留长程或外部信息
锚点	，避免冗余造成重要性分散。
稳定的解释	冗余注入下的稳健归因， 释义改写以及轻微的上下文偏移。
监测信号	所选锚点集合的漂移可以表明检索 变化、提示路径偏差或分布 推理时偏移。

模型的上下文可能会稀释快速变化信号、增加方差并使解释不稳定。我们不截断上下文，也不按边际重要性选择单元，而是在强制保留“近期”上下文的前提下，选取一个能最大化独特信息量的小型子集。由此得到的紧凑上下文能够保留

对近期变化的响应能力，同时保留少数长程锚点（例如每周周期性的滞后项或检索到的关键证据）。

设 $\mathcal{C} = \{C_1, \ldots, C_m\}$ 为有序的上下文单元，其中排序对应于近期性（时间序列滞后项）或提示词位置（大语言模型序列化）。我们定义：$\mathcal{R} \triangleq \{C_1, \ldots, C_L\}$（强制保留的近期窗口），$\mathcal{H} \triangleq \{C_{L+1}, \ldots, C_m\}$（候选项）。轻量级选择器会输出精简后的上下文，$\tilde{\mathcal{C}} = \mathcal{R} \cup \mathcal{S}$，$\mathcal{S} \subset \mathcal{H}$，$|\mathcal{S}| = K$，其中 $\mathcal{S}$ 包含能在 $\mathcal{R}$之外提供最多独特信息的前 K 个上下文单元。对于每个候选项 $C_i \in \mathcal{H}$，我们定义一个以强制保留的近期窗口为条件的依赖性感知评分：

$$\Delta_i^{(\mathcal{R})} \triangleq I(C_i; \hat{T} \mid \mathcal{R}). \tag{5}$$

等价地，我们可以使用相应的归一化**RISE 限定**评分：

$$\mathrm{RISE}_{\mathcal{H}|\mathcal{R}}(C_i) = \frac{I(C_i; \hat{T} \mid \mathcal{R})}{\sum_{C_j \in \mathcal{H}} I(C_j; \hat{T} \mid \mathcal{R}) + \varepsilon}. \tag{6}$$

然后，我们按照 $\Delta_i^{(\mathcal{R})}$（或 $\mathrm{RISE}^{\mathcal{H}|\mathcal{R}}$）从 $\mathcal{H}$ 中选取前 K 个单元作为 $\mathcal{S}$。如果 $C_i = y_{t-i}$ 表示滞后项 i，那么 $\mathcal{R}$ 对应于最近的 L 个滞后项，而 $\mathcal{S}$ 则选出少量长程锚点（例如 $t-24$、$t-168$），这些锚点在以近期窗口为条件后仍具有信息量。这样，特征向量的规模便从 m缩减至 $L+K$。如果 C_i 是提示词片段或检索片段，那么 $\mathcal{R}$ 对应于最近的对话轮次／指令（或优先级最高的控制块），而 $\mathcal{S}$ 则保留少量独特且信息丰富的检索／记忆单元，其信息无法从 $\mathcal{R}$中恢复。

引理 2 (条件冗余意味着不会被选中（理想情况）**)。** 如果候选单元 $C_i \in \mathcal{H}$ 在给定近期窗口的条件下与目标条件独立，即 $C_i \perp\!\!\!\perp \hat{T} \mid \mathcal{R}$，则 $I(C_i; \hat{T} \mid \mathcal{R}) = 0$。因此，除非正分候选项少于 K 个，否则 C_i 不可能出现在前 K 个集合 $\mathcal{S}$ 中。

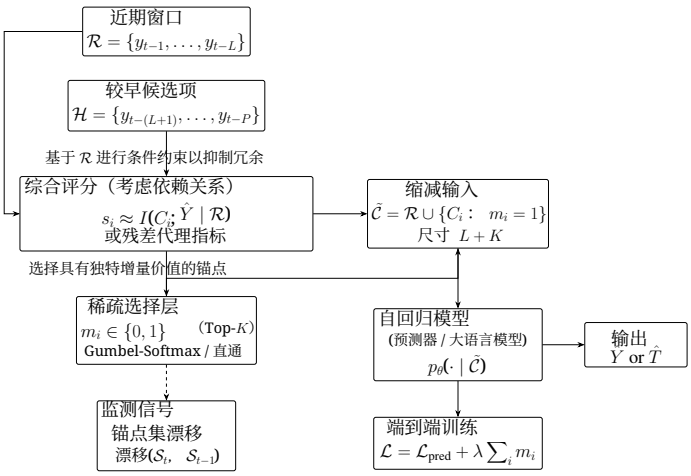

图 2： 轻量级依赖感知选择：保留最近的上下文以应对快速动态，并稀疏地选择具有独特信息价值的长程锚点，从而通过锚点漂移实现推理时监控。

简单来说，如果某个候选上下文单元没有贡献任何超出近期上下文中已有信息的内容，**RISE**绝不会选择它，除非没有更好的替代项。重复或高度重叠的上下文不应仅仅因为多次出现就获得额外的重要性。

推论 2（条件化下的重复／重叠抑制）**.** 如果两个候选 C_i和 C_j 是近似重复项，即C_i （近似地）是$(\mathcal{R}, C_j)$的确定性函数，则 $I(C_i; \hat{T} \mid \mathcal{R})$ （近似地）为零，只要 C_j 可用。因此，检索块中的重叠或滞后项之间的远程季节性冗余会被自然抑制。

这意味着，当两个被检索的块或上下文单元所传达的信息基本相同时，**RISE**至多将重要性归于其中之一，从而防止冗余引起的重要性膨胀。一个好的选择器应对快速变化的信息保持敏感，同时仍保留少量相关的远程上下文。

计算复杂度与估计

计算 **RISE** 需要估计 m 个形式为 $I(C_i; \hat{T} \mid \mathcal{C}_{\setminus i})$ 的条件互信息项。令 n表示评估提示或生成步骤的数量，令 d 表示所选上下文单元表示（例如池化隐状态或 logit 向量）的维度。总计算成本可表示为 $\mathcal{O}(m \cdot \mathrm{CostCMI}(n, d))$，其中 $\mathrm{CostCMI}(n, d)$ 取决于所选的条件互信息估计器（例如基于 kNN、变分或基于核的方法）。在实践中，**RISE** 因三个关键原因而保持轻量。首先，归因针对的是结构化上下文单元，而非原始词元，因此即使面对长提示，m 也能保持较小。

表4：冗余条件下的失效模式及预期诊断特征（输出保留）。

失效模式	基线倾向于做什么	预期特征（诊断-tic）
冗余盲目	在重复项之间拆分或重复分配功劳 / 重叠单元（边际评分）。	重复项上的归因质量 不断增加或碎片化，而模型输出保持不变。
表层形式敏感性	在释义/重排时改变评分 尽管语义等价。	排名/重叠变化而没有 任何输出变化。
结构主导	收敛到结构上必不可少的 单元（例如，问题/模板），在逐项扰动。	移除问题会导致 对数概率大幅下降，无论证据 $\Rightarrow$ 归因列-是否失效于 Q。

其次，可将单元分组（例如检索片段或对话窗口）以降低维度。第三，借助缓存的上下文表示和下一词元分布，**RISE**无需针对每个单元重新运行大语言模型，从而能够在模型运行前或事后于推理时进行轻量审计。

集成学习选择器

我们通过两种互补方式实现**RISE** 背后的依赖感知原则：(i) 面向结构化上-下文单元的轻量级、模型无关*RISE*-Lite 评分模块，以及 (ii) 在生成流水线内 强制执行依赖感知上下文保留的集成学习选择器 。在学习式选择器变体中，一个小型门控网络作用于上下文单元表示$\mathbf{z}_i$，生成选择分数，并通过标准的可微松弛将其转换为二元保留决策。随后，生成器仅以保留的上下文单元子集为条件。选择器和生成器通过端到端训练，并结合稀疏正则项，以促使模型依赖最小上下文集。与**RISE**-Lite 相比，该集成选择器引入了额外的训练成本和超参数，但实现了更紧密的流水线集成，并支持在推理时进行摊销式依赖感知选择。我们使用**RISE**-Lite 进行透明审计和监控，并在需要端到端上下文缩减时使用集成选择器。

设计原则：面向可信人工智能的依赖感知解释

我们提出了人工智能系统，尤其是大语言模型中可信解释的一项关键设计原则：解释必须具备依赖感知能力，体现每个组成部分对模型行为的独特 贡献，而非相关性或出现频率。这使归因从表面关联转向真实依赖。传统方法在存在冗余或重叠时会错误估计重要性；依赖感知解释只重视无法从其他组成部分中恢复的信息，从而避免因重复而获得归因。核心要求之一是冗余不变性：重复相同内容不应提高其归因重要性。由于实际部署中经常会过滤或截断上下文，可信解释在选择上下文后仍应有效；依赖感知归因即使只基于子集计算，也能保留冗余抑制效果。此外，依赖感知解释仅依赖可观察的输入输出行为，无需访问内部机制，因此适用于监控和审计。依赖感知也能提升推理时安全性：故障往往源于提示、检索与记忆之间的交互，而非孤立输入。**RISE** 仅凭输入输出轨迹即可揭示检索坍缩（来源冗余）、提示漂移（指令不一致）和记忆冗余（状态重复）等问题，因此适用于安全关键场景。重要的是，在安全的系统设计中，冗余往往是结构性的：检索可能返回相互重叠的证据，智能体记忆也可能为确保可靠性而重复存储状态。忽视这一点的解释可能会歪曲系统情况，因此需要采用既尊重有价值上下文、又能抑制重复的方法。例如，如果一条安全指令原样重复五次，模型行为通常在首次出现后便不会改变，但边际归因可能仅因重复就将其重要性估计为约五倍。这类解释衡量的是频率，而非依赖；可信解释无论同一信息被重述多少次，都应赋予其相同的重要性。

表 5：基线失效 $\rightarrow$ RISE 修正。上：SQuAD 冗余/释义改写（20）。下：GPT-2 输出保持测试（$n = 30$、 $k = 2$，重叠 $= 0.7$）。

方法	排名稳定性（斯皮尔曼 ρ）↑	前5 项重叠 ↑	Dup-Split ↓
注意力	0.68 ± 0.03	0.88 ± 0.02	0.29 ± 0.01
RISE（我们的）	0.31 ± 0.04	0.75 ± 0.03	$\mathbf{0.50 \pm 0.00}$
扰动	$\mathbf{0.81 \pm 0.05}$	0.83 ± 0.04	0.73 ± 0.06

方法	忠实性差距 ↑	重叠分割指数 ↓
注意力	-3.15 ± 6.77	0.8445 ± 0.0914
扰动（LOUO）	3.79 ± 5.48	0.0139 ± 0.0749
RISE（我们的方法）	3.79 ± 5.48	0.0139 ± 0.0749

实验设置

我们开展实验，以探究当模型输出保持不变时，解释方法是否会避免对冗余或主导上下文给予过高归因。我们的目标并非提高准确率，而是在长上下文变化下对解释行为进行压力测试 Bommasani 等人（2021）。在 SQuAD v1.1 验证集上使用 GPT-2 Radford 等人（2019）；Rajpurkar 等人（2016），并以开放权重自回归模型作为 API- 的代理

的系统（例如 GPT-3.5）Brown 等人（2020），我们聚焦于 输入–输出 在严格**输出保持契约**下的解释行为，仅在模型输出固定时对解释进行评分。在该契约下，任何归因变化都是解释不稳定性，而非模型行为变化，从而使冗余不变性可测量。从基础提示 X 出发，我们在保持答案不变的同时引入冗余上下文，方式为：**(i)复制**，(ii)

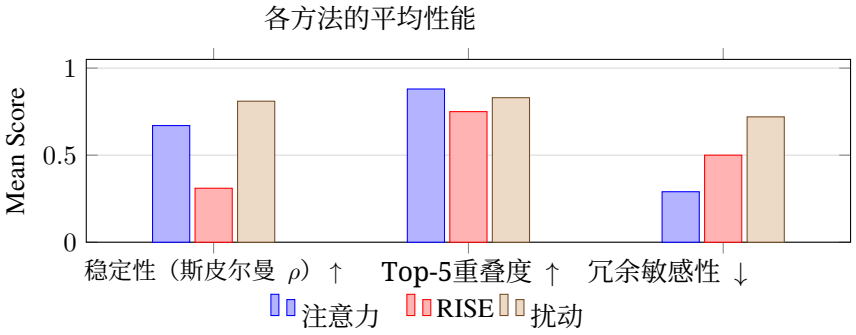

图 3： 跨方法的总体归因行为。**RISE** 表现出更强的冗余抑制能力（更低的冗余敏感性），尽管其原始秩稳定性较低，这反映了重叠情况下对依赖关系感知的评分机制。

释义改写，以及 (iii)**重排序/重叠**。这些方法反映了现实中的提示工程和记忆效应。我们将 **RISE** 与两类基线方法进行比较：**基于注意力的启发式方法**和**基于扰动的归因方法**。在存在冗余的情况下，基线方法可能表现出：(1)冗余盲视，即重复项之间的分数膨胀；(2) 表层形式敏感性，即重排序会影响归因，而不改变输出；

以及 (3) 结构主导，即必要单元会扰乱生成。我们使用**秩稳定性**（斯皮尔曼 ρ）、**前K 重叠**和**冗余敏感性（Dup-Split）**进行评估，其中 Dup-Split 用于衡量复制条件下的归因片段。优秀的解释器在输出不变的情况下能够保持较低的 Dup-Split 值，这表明其具有冗余不变性。Dup-Split 是冗余场景下可靠的诊断工具。

结果

我们的首要目标是检验解释方法在冗余情况下是否仍然忠实：当模型输出不变时，归因是否不会对重复或重叠的上下文给予额外奖励，也不会集中到结构上占主导地位的单元上？图3和表5等图表报告了跨示例的总体表现。我们观察到两种截然不同的基线失效模式；当提示中包含重复项、释义或重叠检索内容时，这些模式会愈发明显。首先，**基于注意力的启发式方法** 表面上可能很稳定（例如，Top-K 重叠度较高），但这种稳定性往往是由扩散造成的：重要性分散到重复或相关单元上，产生看似一致的支撑集，却无法分离出唯一必需的上下文。因此，在冗余丰富的提示中，注意力仍可能显得“稳定”，却依然过度归功于重复项。其次，**基于扰动的归因** 往往能取得更高的斯皮尔曼排名稳定性，但这可能具有误导性，因为扰动容易受到结构主导的影响。移除问题（或类似的结构性单元）会扰乱生成，不论语义证据如何，因此即使有信息的证据位于检索到的上下文或支撑上下文中，扰动也可能将压倒性的重要性归给问题。在存在复制的情况下，扰动也可能夸大重要性，因为重复单元会形成多个移除点，从而增加边际效应，却没有增加任何新信息。RISE针对另一项标准：条件依赖下的冗余不变性。RISE只归因于无法从其余上下文中恢复的信息，从而抑制冗余引起的重要性膨胀，并防止重要性在近似重复项之间分散。这些趋势体现了**具备冗余意识的行为**：在复制和重叠情况下，RISE能更有效地抑制冗余（Dup-Split较低）；而其原始秩稳定性较低则在意料之中，因为它有意 将重要性从冗余副本重新分配给唯一信息单元，这正是冗余压力测试所要检验的特性，而非缺陷。相比之下，基线方法看似稳定，却无法分离出独特依赖。我们还观察到扰动在冗余情况下的一种结构性失效模式（表6）：由于问题具有结构必要性，扰动会集中到问题上，从而混淆可移除性 与信息贡献。RISE(-Lite)通过以其余上下文为条件，将权重重新分配给仍然包含信息的上下文单元，从而得到与唯一信息证据相一致的紧凑支撑集。我们进一步仅在保持输出不变的 干预条件下评估解释（解码后的答案/片段相同，或下一个词元的 argmax 相同）；在这一约束下，归因变化意味着解释不稳定性，而不是模型行为变化。表5明确展示了基线方法的失效以及 → RISE的修正：**(i)** 在RAG重叠情况下，注意力会在重叠的分块之间分散重要性（SplitIndex较高），而RISE会抑制重叠（SplitIndex较低）；**(ii)** 在忠实性移除测试中，注意力的顶底差距为负，而RISE则产生正差距，这表明移除排名靠前的单元对答案似然的降低幅度，远大于移除排名靠后的单元。逐例分析和额外的压力测试（附录）证实了这些效应

并非由一小部分实例驱动，并且在不同模型间持续存在。

讨论与局限性

RISE 为自回归语言模型如何依赖上下文提供了一种考虑依赖关系的解释。我们的结果表明，在存在冗余时，标准的“解释稳定性”概念可能具有误导性：较高的一致性或许仅仅反映了重要性在相关或重复单元之间的分散（扩散），而非模型依赖唯一信息上下文。通过以其余上下文为条件，**RISE** 能够区分哪些内容不可或缺、哪些只是重复，从而在提示、检索和记忆流水线中给出更清晰、更具可操作性的解释。表6突显了另一种互补的失效模式：基于扰动的归因可能会集中到问题上，因为无论语义如何，移除问题都会扰乱生成，从而混淆结构必要性 与信息贡献。

Unit	RISE-Lite	扰动
系统（Sys）	0.27	0.00
上下文1（C1）	0.21	0.00
上下文2 (C2)	**0.39**	0.08
上下文3 (C3)	0.12	0.00
问题 (Q)	≈0.00	**0.92**

表6： 提示冗余情况下具有代表性的归因。由于结构性支配，扰动会集中到问题上，而 RISE-Lite 则将归因集中在独特且有信息量的上下文上。

相比之下，**RISE-Lite**会抑制结构上占主导地位但仍可恢复的单元，并将权重重新分配给经过条件化处理后仍有信息量的上下文成分。**RISE** 的一个局限是，它处理的是结构化上下文单元 （提示块、检索片段、对话轮次），而非单个词元，因此提升了解释性和可部署性，但牺牲了词元级粒度。此外，我们关注的是冗余和语义重叠；对于这些目标，保留主要条件依赖结构比完美校准的依赖估计更为关键。未来的工作将开发更优的估计器、量化近似误差，并将 **RISE** 集成到面向智能体和检索增强系统的实时安全与可靠性流水线中。

伦理与可复现性： 本研究通过缓解冗余或经过操纵的上下文所导致的误导性解释，提升了大语言模型的透明度，同时没有引入新的模型能力或伦理风险。RISE 与模型无关，无需重新训练，并在公开模型和数据集上进行评估；数据、模型和代码可通过 https://anonymous.4open.science/r/LLM-Trustworthy-3448/README.md 获取。

结论

我们探讨了自回归大语言模型面临的一项核心可信性挑战：在冗余条件下，识别生成所必需的独特上下文元素。我们以条件信息作为设计原则，
RISE 可抑制冗余引起的重要性膨胀，保持可解释性，并支持推理时监控。通过聚合评估、压力测试和跨模型诊断，我们表明，表面上的稳定性可能掩盖注意力和基于扰动的启发式方法中的结构性失效。除了可解释性之外，**RISE**还能在推理时实现实用的安全防护，例如检测检索坍缩、提示漂移和智能体记忆冗余，为冗余易发的大语言模型部署中的可信提示审计、检索验证和监控奠定基础。

参考文献

Rishi Bommasani 等。基础模型的机遇与风险。*arXiv* 预印本 *arXiv:2108.07258*，2021。

Tom B. Brown 等。语言模型是少样本学习器。神经信息处理系统大会，2020。

Thomas M. Cover 和 Joy A. Thomas。《信息论基础》。John Wiley & Sons，第二版，2006。doi: 10.1002/047174882X。

Finale Doshi-Velez 和 Been Kim。迈向可解释机器学习的严谨科学。*arXiv* 预印本 *arXiv:1702.08608*，2017。

Shi Feng、Byron C. Wallace、Alvin Grissom II、Mohit Iyyer、Pedro Rodriguez 和 Jordan Boyd-Graber。神经模型的病理特征使解释变得困难。载于*EMNLP* 论文集，第 3719–3728 页，2018。

Sara Hooker。解释黑盒：什么是解释？*Queue*，2021。

Alon Jacovi 和 Yoav Goldberg。《迈向忠实可解释的自然语言处理系统：我们应如何定义和评估忠实性？》载于第*58*届计算语言学协会年会论文集（*ACL*），第4198–4205页，2020。

Alon Jacovi 和 Yoav Goldberg。《将对人工智能的信任形式化》。*ACL*论文集，2021。Sarthak Jain 和 Byron C. Wallace。《注意力并非解释》。载于*2019*年北美计算语言学协会分会年会论文集（*NAACL*），第3543–3556页，2019。Pieter-Jan Kindermans、Kristof T. Sch¨utt、Maximilian Alber、Klaus-Robert M¨uller 和 Sven D¨ahne。《显著性方法的（不）可靠性》。载于可解释人工智能：深度学习的诠释、解释与可视化，第267–280页。Springer，2019。Patrick Lewis 等。《面向知识密集型自然语言处理任务的检索增强生成》。载于神经信息处理系统大会进展，2020。Jiwei Li、Will Monroe 和 Dan Jurafsky。《通过表示擦除理解神经网络》。载于*NAACL-HLT*论文集，第122–131页，2016。Yifan Li 等。《针对大语言模型的提示注入攻击》。*arXiv*预印本 *arXiv:2302.12173*，2023。Nelson F. Liu 等。《迷失在中间：语言模型如何利用长上下文》。*ACL*汇刊，2024。Tim Miller。《人工智能中的解释：来自社会科学的见解》。人工智能，2019。

Chris Olah 等人。可解释性的构建模块。*Distill*，2018。

OpenAI。GPT-4 技术报告。*arXiv* 预印本 *arXiv:2303.08774*，2023。

Alec Radford、Jeffrey Wu、Rewon Child、David Luan、Dario Amodei 和 Ilya Sutskever。语言模型是无监督多任务学习者。*OpenAI* 技术报告，2019。

Pranav Rajpurkar、Jian Zhang、Konstantin Lopyrev 和 Percy Liang。SQuAD：100,000+ 个用于文本机器阅读理解的问题。自然语言处理经验方法会议论文集，2016。

Sofia Serrano 和 Noah A. Smith。注意力是否具有可解释性？载于计算语言学协会第 *57* 届年会论文集（*ACL*），第 2931–2951 页，2019 年。

Karen Simonyan、Andrea Vedaldi 和 Andrew Zisserman。深入卷积网络：图像分类模型和显著性图的可视化。*arXiv* 预印本 *arXiv:1312.6034*，2014。

Mukund Sundararajan、Ankur Taly 和 Qiqi Yan。深度网络的公理归因。载于*ICML* 论文集，2017 年。

Jason Wei 等。越狱：大语言模型安全训练为何会失效？*arXiv* 预印本 *arXiv:2307.02483*，2023 年。Sarah Wiegreffe 和 Yuval Pinter。注意力并非解释。载于*EMNLP-IJCNLP* 论文集，2019 年。

Shunyu Yao、Jeffrey Zhao、Dian Yu、Nan Du、Izhak Shafran、Karthik Narasimhan 和 Yuan Cao。React：协同语言模型中的推理与行动。载于国际学习表征会议（*ICLR*），2023 年。网址：https://arxiv.org/abs/2210.03629。

其他理论结果与证明

在主论文中，我给出了**RISE** 的核心冗余感知保证（引理 3、推论 1、命题 1），因为这些结果直接指导了实证设计，并解释了提示重复和释义改写下观察到的行为。在本附录中，我提供辅助性质（非负性、归一化）、群组级扩展，以及（可选的）学习式选择器变体。所有证明均按步骤撰写，以明确呈现逻辑脉络，确保结果可复现。

符号与预备知识

令 $C = \{c_1, \ldots, c_m\}$ 表示上下文单元的集合（指令片段、检索片段、问题等）。令Y表示所关注的模型输出变量（例如，下一词元分布、答案字符串的对数概率，或由生成过程导出的标量代理量）。对于 $i \in \{1, \ldots, m\}$，令 $C_{-i} := C \setminus \{c_i\}$。我将条件唯一依赖 （CUD）写作

$$I(Y; c_i \mid C_{-i}) \;=\; \mathbb{E}\left[\log \frac{p(Y \mid c_i, C_{-i})}{p(Y \mid C_{-i})}\right]. \tag{7}$$

条件唯一依赖非负，且当且仅当 $Y \perp c_i \mid C_{-i}$ （条件独立性）时等于零，前提是满足标准正则性假设。**RISE** 为每个上下文单元分配一个条件唯一依赖贡献

$$\phi_i \;:=\; I(Y; c_i \mid C_{-i}), \qquad i = 1, \ldots, m, \tag{8}$$

并使用归一化归因

$$a_i \;:=\; \frac{\phi_i}{\sum_{j=1}^{m} \phi_j + \varepsilon}, \tag{9}$$

其中较小的 $\varepsilon > 0$仅用于数值稳定性。（在正文中，我将 ε 设为一个极小常数，并为简洁起见省略它。）主论文中的引理 3 使用等价关系 $I(Y; c_i \mid C_{-i}) = 0 \Leftrightarrow (Y \perp c_i \mid C_{-i})$ 来形式化冗余抑制。以下辅助引理为其余性质和实现细节提供支持。

引理 3(非　　负性)。 对于任意上下文单元 c_i，***RISE*** 贡献满足 $\phi_i = I(Y; c_i \mid C_{-i}) \geq 0$。

证明。 令 $C = (c_i, C_{-i})$ 表示完整上下文，其中 C_{-i} 汇集除 c_i 之外的所有上下文单元。随机变量 Y 表示生成目标（例如，下一个词元结果或下一个词元分布），下文所有概率均相对于由数据生成过程与待评估的自回归模型共同诱导的联合分布来取。根据定义，条件唯一依赖为

$$I(Y; c_i \mid C_{-i}) = \mathbb{E}_{(Y, c_i, C_{-i})}\left[\log \frac{p(Y, c_i \mid C_{-i})}{p(Y \mid C_{-i})\, p(c_i \mid C_{-i})}\right]. \tag{10}$$

利用 $p(Y, c_i \mid C_{-i}) = p(Y \mid c_i, C_{-i})\; p(c_i \mid C_{-i})$，方程 10 中的比值可简化为

$$\frac{p(Y, c_i \mid C_{-i})}{p(Y \mid C_{-i})\, p(c_i \mid C_{-i})} = \frac{p(Y \mid c_i, C_{-i})}{p(Y \mid C_{-i})}. \tag{11}$$

代回后，我得到标准条件形式

$$I(Y; c_i \mid C_{-i}) = \mathbb{E}_{(Y, c_i, C_{-i})}\left[\log \frac{p(Y \mid c_i, C_{-i})}{p(Y \mid C_{-i})}\right]. \tag{12}$$

方程 12 是逐点对数似然比的平均值。按条件变量 (c_i, C_{-i}) 对各项分组，可将该平均值转化为条件分布$p(Y \mid c_i, C_{-i})$ 与 $p(Y \mid C_{-i})$ 之间的期望 KL 散度。

首先以 (c_i, C_{-i}) 为条件，再对 Y 取期望：

$$\begin{aligned} I(Y; c_i \mid C_{-i}) &= \mathbb{E}_{(c_i, C_{-i})}\left[\mathbb{E}_{Y \mid c_i, C_{-i}}\left[\log \frac{p(Y \mid c_i, C_{-i})}{p(Y \mid C_{-i})}\right]\right] \\ &= \mathbb{E}_{(c_i, C_{-i})}\left[D_{\mathrm{KL}}(p(Y \mid c_i, C_{-i}) \,\|\, p(Y \mid C_{-i}))\right]. \end{aligned} \tag{13}$$

对于定义在同一支撑集上的任意分布 p 与 q，Kullback–Leibler 散度满足

$$D_{\mathrm{KL}}(p \| q) \geq 0, \tag{14}$$

当且仅当 $p = q$ 几乎处处成立时取等号。将其应用于每个实现的二元组（c_i、C_{-i}），可得

$$D_{\mathrm{KL}}(p(Y \mid c_i, C_{-i}) \,\|\, p(Y \mid C_{-i})) \geq 0 \qquad \text{for all } (c_i, C_{-i}). \tag{15}$$

由于公式 13 中的被积函数处处非负，因此其关于（c_i、C_{-i}）的期望也非负。因此，

$$I(Y; c_i \mid C_{-i}) \geq 0, \tag{16}$$

这意味着 $\phi_i \geq 0$正如所声称的那样。 □

引理 3 保证，在公式 9 的归一化步骤之前，每个未经归一化的 **RISE** 分量 ϕ_i 都是有效的非负“归因质量”先验；因此，归一化只会重新缩放这些质量，以便在不同上下文单元之间进行比较，而不会改变其符号。

引理 4 (归一化). 假设 $\sum_{j=1}^{m} \phi_j > 0$。那么，方程 *9* 中归一化后的 ***RISE*** 归因 a_i 满足 $a_i \geq 0$ 和 $\sum_{i=1}^{m} a_i = 1$。

证明。 根据方程 9，单位 i 的归一化归因为

$$a_i \;=\; \frac{\phi_i}{\sum_{j=1}^{m} \phi_j}. \tag{17}$$

假设 $\sum_{j=1}^{m} \phi_j > 0$ 确保方程 17 中的分母严格为正，因此归一化是良定义的。由于 $\{\phi_i\}_{i=1}^{m}$ 是非负的“质量”（引理 3），用其正总和相除只是将它们重新缩放为可在单纯形上解释的权重。根据引理 3，$\phi_i \geq 0$ 对所有 i均成立。由于假设分母 $\sum_{j=1}^{m} \phi_j$ 严格为正，用正分母除以非负分子仍保持非负性。因此，对每个 i，$a_i \geq 0$ 。

如果 $\sum_{j=1}^{m} \phi_j = 0$，那么所有 $\phi_j = 0$ 都成立（因为每个 $\phi_j \geq 0$），此时归一化将无定义。因此，条件 $\sum_{j=1}^{m} \phi_j > 0$ 恰好排除了这一退化情形。对 i 求和方程 17 得到

$$\sum_{i=1}^{m} a_i \;=\; \sum_{i=1}^{m} \frac{\phi_i}{\sum_{j=1}^{m} \phi_j} \;=\; \frac{\sum_{i=1}^{m} \phi_i}{\sum_{j=1}^{m} \phi_j} \;=\; 1, \tag{18}$$

其中，第二个等式将常数分母提取出来，最后一个等式成立是因为分子和分母是相同的有限正量。

□

引理 4 证明了将 **RISE** 解释视为定义在上下文单元上的概率分布是合理的： a_i 是非负权重，且权重之和为 1，因此可以在实验部分进行 Top-k 重叠和质量拆分诊断。

在某些设置中，将重要性归因于单元的组（例如，属于同一检索段落的所有词元，或所有指令行）而非单个单元会很有用。为完整起见并支持消融实验，我纳入了这一扩展。令 G 为索引集合 $\{1, \ldots, m\}$ 的一个划分，其中各组 $g \in G$互不相交。定义$c_g := \{c_i : i \in g\}$ 和 $C_{-g} := C \setminus c_g$。

引理 5 (组归因)。 组贡献 $\Phi_g := I(Y; c_g \mid C_{-g})$ 对任意 $i \in g$均满足 $\Phi_g \geq \phi_i$，并具有如下链式分解

$$I(Y; c_g \mid C_{-g}) = \sum_{i \in g} I(Y; c_i \,|\, C_{-g}, \{c_j\}_{j \in g,\, j<i}), \tag{19}$$

对于组内索引的任意固定排序。

证明。 我们分两部分进行：（i）始终成立的链式分解；（ii）仔细的不等式论证，用于解释组归因何时以及为何优于单元归因。

第一部分：链式分解（始终成立）。 固定 g 中索引的任意排序，记为 $g = \{i_1, i_2, \ldots, i_{|g|}\}$。为方便记号，定义前缀集合

$$c_{g,<k} \;:=\; \{c_{i_1}, \ldots, c_{i_{k-1}}\}, \tag{20}$$

其中 $c_{g,\,<1} = \varnothing$。对在 C_{-g} 条件下的集合 $c_g =$(${c_{i_1}}$、.....、 ${c_{i_{|g|}}}$)，应用条件唯一依赖的链式法则：

$$\begin{aligned} I(Y; c_g \mid C_{-g}) &= I\big(Y; (c_{i_1}, \ldots, c_{i_{|g|}}) \mid C_{-g}\big) \\ &= \sum_{k=1}^{|g|} I(Y; c_{i_k} \,|\, C_{-g}, c_{g,<k}). \end{aligned} \tag{21}$$

对求和重新编排索引后，恰好得到公式 19。这就建立了该分解。每一项 I(*Y*; $c_{i_k} \mid C_{-g}$, $c_{g,\,<k}$) 衡量的是：在固定组外其余上下文（C_{-g}）后，并且在此前的组内元素 $c_{g,\,<k}$ 已揭示之后，再揭示 c_{i_k} 所获得的关于 Y的额外信息。因此，求和累积了整个组所贡献的增量“新”信息。公式 21 中的每个加数都是条件唯一依赖。根据引理 3，每一项都非负。因此， $\Phi_g = I$(*Y*; $c_g \mid C_{-g}$) ≥ 0。

第二部分：相对于单一单元的支配性（条件与证明）。由于公式 21 中的所有加数均非负，总和至少与任一单独加数一样大：

$$\Phi_g = \sum_{k=1}^{|g|} I(Y; c_{i_k} \,|\, C_{-g}, c_{g,<k}) \;\geq\; I(Y; c_{i_k} \,|\, C_{-g}, c_{g,<k}) \quad \text{for all } k. \tag{22}$$

特别地，如果我选择一种排序，将特定的 $i \in g$ 置于首位（因此 $c_{g,<1} = \varnothing$ 和 $i_1 = i$），那么方程 22 给出了一个简洁的界

$$\Phi_g \;\geq\; I(Y; c_i \mid C_{-g}). \tag{23}$$

这个不等式无条件成立，即适用于任意 联合分布。注意， $C_{-g} = C \setminus c_g$ 会移除组中的所有 单元，而 $C_{-i} = C \setminus c_i$ 只会移除单一单元 c_i。因此，对于任意 $i \in g$，

$$C_{-g} \subseteq C_{-i}, \tag{24}$$

并且事实上， $C_{-i} = (C_{-g}) \cup (c_g \setminus c_i)$。比较 Φ_g 和 ϕ_i，需要比较不同条件集下的条件唯一依赖。一般而言，条件唯一依赖对于条件集并不具有单调性，因此在没有额外结构的情况下， $\Phi_g \geq \phi_i$ 并非普遍成立的不等式。普遍成立的是链式分解和界方程 23；在典型的提示分组机制下，额外推导至 ϕ_i 这一步也成立，因为组内成员通常被设计为彼此冗余，且不会产生抑制效应。在提示分组场景中，一个充分且自然的条件是：一旦已知 $(C_{-g}、\ c_i)$，组中其余元素对于预测 Y而言，在条件意义上是冗余的，即

$$I(Y; c_g \setminus c_i \,|\, C_{-g}, c_i) = 0. \tag{25}$$

在公式 25 下，链式法则给出

$$\begin{aligned} \Phi_g = I(Y; c_g \mid C_{-g}) &= I(Y; c_i \mid C_{-g}) + I(Y; c_g \setminus c_i \,|\, C_{-g}, c_i) \\ &= I(Y; c_i \mid C_{-g}). \end{aligned} \tag{26}$$

将公式 26 与 $C_{-i} = (C_{-g}) \cup (c_g \setminus c_i)$这一事实结合起来，公式 25 意味着，将 $(c_g \setminus c_i)$ 加入条件中不会改变 c_i 对 Y 的预测内容，因此得到

$$I(Y; c_i \mid C_{-g}) = I(Y; c_i \mid C_{-i}) = \phi_i. \tag{27}$$

因此，在公式 25 下，

$$\Phi_g \;=\; I(Y; c_i \mid C_{-g}) \;=\; \phi_i, \tag{28}$$

因此， $\Phi_g \geq \phi_i$ 成立（在这一充分条件下取等号）。在实践中，只有当同一组中的成员彼此存在重叠时，才通常会进行分组（例如，检索段落中的词元、指令块中的行，或近乎重复的检索片段）。在这种情形下，公式 25（或其近似形式）是合理的建模假设，因此组归因 Φ_g 表现为单单元贡献的上包络，同时仍可通过公式 19 加性地分解为非负增量。 □

引理 5 支持将 **RISE** 聚合为检索片段级或指令级解释，而不改变底层的依赖感知语义。

学习选择器变体

本节为可选内容，若篇幅或范围受限则可省略。我们将其放在此处，是为了把以研讨会为核心的贡献（冗余条件下的依赖感知归因）与可选的学习选择机制区分开来。在某些部署中，我们可能希望在归因之前自动选择一小部分上下文单元（例如，以降低计算量，或聚焦于最小支持集）。学习型选择器 $s(C)$ 可对单元生成二值掩码，随后将**RISE** 应用于所选子集。本附录形式化说明了此类选择在何种情况下能够保持冗余保证。算法1是一个模型前 或提示前 模块：它可对上下文单元的紧凑表示（池化隐状态、嵌入或汇总统计量）进行操作，且无需重新训练下游模型。因此，它易于审计，并可集成到推理时监控中。若某个上下文单元并未在已有信息之外提供任何新信息，则绝不应将其选为重要单元。

算法1：轻量级依赖感知上下文选择

输入： 上下文单元 $\mathcal{C} = \{C_1, \ldots, C_m\}$（有序），强制窗口 L，锚点预算 K，目标 $\hat{}_T$ **输出：** 精简上下文 $\tilde{\mathcal{C}}$

1: $\mathcal{R} \leftarrow \{C_1, \ldots, C_L\}$,
$\mathcal{H} \leftarrow \{C_{L+1}, \ldots, C_m\}$2: **对所有** $C_i \in \mathcal{H}$**执行**
3: $\Delta_i^{(\mathcal{R})} \leftarrow I(C_i; \hat{} T \mid \mathcal{R})$4: $\mathcal{S} \leftarrow$ 前K 个单元,
取自 $\mathcal{H}$并按 $\Delta_i^{(\mathcal{R})}$ **排序**5: **返回** $\tilde{\mathcal{C}} \leftarrow \mathcal{R} \cup \mathcal{S}$

命题 2（选择器一致的冗余抑制）。 假设选择器返回一个子集 $S \subseteq \{1, \ldots, m\}$，使得对于任意满足 $Y \perp c_i \mid C_{-i}$的冗余单元 c_i，要么***(i)*** $i \notin S$，要么***(ii)*** S 包含一个条件集，使 c_i 在受限上下文中成为冗余。则在所选上下文上计算的***RISE*** 保持冗余抑制： $I(Y; c_i \mid C_{S\setminus\{i\}}) = 0$ 对所有冗余 $i \in S$成立。

证明。 我们通过将冗余证书从完整上下文追踪到所选（受限）上下文来证明该陈述。令 $C = \{c_1, \ldots, c_m\}$ 表示全部上下文单元集合，令 $S \subseteq \{1, \ldots, m\}$ 为选择器返回的集合。固定一个索引 i，它在完整上下文中是冗余 的，其含义为

$$Y \perp c_i \mid C_{-i}. \tag{29}$$

我们的目标是证明：只要 $i \in S$，在受限条件集 $C_{S\setminus\{i\}}$内计算**RISE**时，同样的冗余仍然成立，即

$$I\big(Y; c_i \mid C_{S\setminus\{i\}}\big) = 0. \tag{30}$$

方程 29 表明，一旦已知完整上下文的其余部分， c_i 便不包含关于 Y的唯一 信息。只有当选择器要么丢弃 c_i，要么保留足够的相关条件信息，以便在限制之后证明相同的条件独立性时，选择才能保持这种抑制。

根据条件独立性与条件唯一依赖之间的标准等价关系，

$$Y \perp c_i \mid C_{-i} \quad \Longleftrightarrow \quad I(Y; c_i \mid C_{-i}) = 0. \tag{31}$$

因此，在方程 29 下，单元 i 在完整 上下文中的**RISE**分子消失。

我们现在传 播这一冗余性陈述，使其通过选择器。这里有两种情况，与
命题的假　　设相对应。

如果 $i \notin S$，那么所选上下文中不存在 c_i，并且不会在受限**RISE**中为 i 计算归因项。因此，“对于所有冗余的 $i \in S$”这一主张在此情况下为空真。假设 $i \in S$。根据该命题的假设，所选集合包含一个条件集，使得在受限上下文中 c_i 具有冗余性。具体而言，这意味着， $C_{S\setminus\{i\}}$ 中保留的变量足以推出

$$Y \perp c_i \mid C_{S\setminus\{i\}}. \tag{32}$$

直观地说，尽管我们不再以 C_{-i}的全部内容为条件，选择器仍保留了用于阻断 c_i 与 Y之间依赖关系所需的特定上下文单元子集。

一旦等式 32 成立，在所选上下文上计算的 **RISE** 分子也必定为零，因为条件唯一依赖恰好衡量了经过条件化后剩余的依赖。

Ap应用与等式 31 相同的等价关系，但现在使用条件集 $C_{S\setminus\{i\}}$，我们得到　　n

$$Y \perp c_i \mid C_{S\setminus\{i\}} \quad \Longleftrightarrow \quad I\big(Y; c_i \mid C_{S\setminus\{i\}}\big) = 0. \tag{33}$$

将等式 32 与等式 33 结合，即可证明对于任意冗余的 $i \in S$，等式 30 均成立。

在这两种情况下，无论选择器排除冗余单元，还是保留能够证实冗余的条件集，在所选上下文上计算的 **RISE** 都会保持冗余抑制：

$$I\big(Y; c_i \mid C_{S\setminus\{i\}}\big) = 0 \quad \text{for all redundant } i \in S. \tag{34}$$

□

命题 2 澄清了，只要选择器不会系统性地移除用于确认冗余所需的变量，选择就不会使冗余感知目标失效；换言之，能够保留（或重新引入）冗余筛查条件集的选择器，在筛选后仍能维持**RISE**的抑制行为。

命题 3（保留快速变化的动态）。 假设生成目标可以分解为短期成分和长期成分，即$\hat{T} = f(\mathcal{R}) + g(\mathcal{A}) + \eta$, 其中 $\mathcal{A} \subseteq \mathcal{H}$ 是一个较小的长程锚点集合， η 是与（$\mathcal{R}, \mathcal{A}$）独立的噪声。那么，始终保留 $\mathcal{R}$ 的选择器能够保留对 $\mathcal{R}$中所体现的快速变化的敏感性，而通过 $I(C_i;\ T \mid \mathcal{R})$ 从 $\mathcal{H}$ 中选择锚点，则可恢复 $\mathcal{A}$ 中的长程信息（在总体情形下）。

证明。 我们分别证明两个论断：（i）保留 $\mathcal{R}$ 可保持对快速变化的敏感性；（ii）由 $I(C_i; T \mid \mathcal{R})$进行的条件选择足以在总体情形下恢复 $\mathcal{A}$ 中的长程锚点。令 $\mathcal{R}$ 表示选择器所保留的强制近期窗口，令 $\mathcal{H}$ 表示额外候选的候选池（例如，较早的上下文、检索片段、记忆条目）。模型目标满足

$$\hat{T} = f(\mathcal{R}) + g(\mathcal{A}) + \eta, \tag{35}$$

其中 $\mathcal{A} \subseteq \mathcal{H}$ 是承载长程信息的（较小的）锚点变量集合， η 是独立噪声，$\eta \perp (\mathcal{R}, \mathcal{A})$。选择器始终保留 $\mathcal{R}$ ，并按条件唯一依赖分数对候选 $C_i \in \mathcal{H}$ 进行排序

$$\Delta_i \ = \ I(C_i; \hat{T} \mid \mathcal{R}). \tag{36}$$

对 $\mathcal{R}$取条件会明确地从T中尚待解释的部分去除短期贡献 $f(\mathcal{R})$。取条件之后，候选 C_i 与 T 之间任何剩余的依赖都必定来自 C_i中关于长程项 $g(\mathcal{A})$的信息（至多相差噪声），而这正是我们希望锚点所捕获的内容。根据公式 35， T的短期成分是近期窗口 $\mathcal{R}$的函数。因此，近期状态的任何变化（例如，新约束、时间序列的近期偏移，或新检索到的最新信息）都会通过 $f(\mathcal{R})$改变 T。由于选择器在设计上始终保留 $\mathcal{R}$ ，生成器始终接收到决定短期成分 $f(\mathcal{R})$的变量。因此，无论从 $\mathcal{H}$ 中选择哪些额外锚点，生成都仍对编码在 $\mathcal{R}$中的快速变化保持敏感。这就确立了第一个论断。

T 中长程项 $g(\mathcal{A})$ 引入的唯一剩余来源是 $\mathcal{A}$ 的（潜在）非平凡部分。第二部分表明，通过公式 36 进行条件选择可以恢复这种长程信息。

以 $\mathcal{R}$ 为条件，并定义残差目标

$$\hat{T}_{\text{res}} \ := \ \hat{T} - f(\mathcal{R}). \tag{37}$$

利用公式 35，可得

$$\hat{T}_{\text{res}} \ = \ g(\mathcal{A}) + \eta. \tag{38}$$

由于 $\eta \perp (\mathcal{R},\ \mathcal{A})$，以 $\mathcal{R}$ 为条件不会在 η 与任何 $C_i \in \mathcal{H}$ 之间产生依赖；噪声仍然独立于长程结构。分数 $\Delta_i = I(C_i;\ \hat{T} \mid \mathcal{R})$ 衡量的是，在 $\mathcal{R}$ 中编码的短期成分已知之后，C_i 能提供多少关于 T 的信息。根据公式 38，这等价于关于 $g(\mathcal{A})$ 的信息（独立噪声除外）。考虑任意不携带除 $\mathcal{R}$ 之外关于锚点的额外信息的候选 $C_i \in \mathcal{H}$ ，即在给定 $\mathcal{R}$ 时，它与 $\mathcal{A}$ 条件独立：

$$C_i \perp \mathcal{A} \mid \mathcal{R}. \tag{39}$$

由于 $g(\mathcal{A})$ 是 $\mathcal{A}$ 的一个（可测）函数，方程39意味着

$$C_i \perp g(\mathcal{A}) \mid \mathcal{R}. \tag{40}$$

结合 $\eta \perp (C_i,\ \mathcal{R},\ \mathcal{A})$，我们得到

$$C_i \perp \hat{T}_{\text{res}} \mid \mathcal{R}, \tag{41}$$

因此，

$$I(C_i; \hat{T} \mid \mathcal{R}) = I(C_i; \hat{T}_{\text{res}} \mid \mathcal{R}) = 0. \tag{42}$$

因此，在总体情形下，凡是无法提供 $\mathcal{R}$ 中已有信息之外的长程信息的候选项，其条件唯一依赖分数均为零。

步骤5（锚点在具有相关性时可获得正分）。现在考虑一个锚点 $A \in \mathcal{A}$。根据分解方程 35，A 通过 $g(\mathcal{A})$影响 T。如果给定 $\mathcal{R}$ 时，A 不是冗余的（即，它携带了一些关于 $g(\mathcal{A})$、而 $\mathcal{R}$ 中尚未包含的信息），那么

$$I(A; \hat{T} \mid \mathcal{R}) = I(A; g(\mathcal{A}) \mid \mathcal{R}) > 0. \tag{43}$$

因此，锚点的得分严格为正。方程 42–方程 43 表明，在理想的总体情形下存在明显区分：$\mathcal{A}$ 中的锚点具有正的条件信息分数，而非信息性候选项的得分为零。因此，按 Δ_i 排序的 top-K 选择器将会选出锚点（至多存在 $K \geq |\mathcal{A}|$，且不存在完全并列或退化的情况）。假设 $K \geq |\mathcal{A}|$，且每个 $A \in \mathcal{A}$ 均满足方程 43，则按 $I(C_i; T \mid \mathcal{R})$降序选择，便能在总体情形下恢复 $\mathcal{A}$ 中的信息性长程变量。这就确立了第二项主张。保留 $\mathcal{R}$ 可确保通过 $\hat{}f(\mathcal{R})$对快速变化的动态保持响应；同时，根据 $I(C_i; T \mid \mathcal{R})$从 $\mathcal{H}$ 中进行条件选择，可以筛选并恢复能够解释剩余长时域结构 $g(\mathcal{A})$的长程锚点，避免被冗余历史信息淹没。□

实际上，这保证了基于 **RISE** 的选择能够快速响应新的或不断演变的上下文，同时保留少量有信息量的长程信号，避免被冗余历史信息淹没。强制保留近期窗口可确保对快速演变的上下文作出响应（例如，时间序列状态发生变化或刚刚提出新的用户约束），而有条件地选择少数锚点，则能保留关键的长时域结构，同时避免向模型输入过多冗余上下文。

其他实验细节

我们将每个提示划分为明确且具有语义意义的上下文单元，包括系统指令行、任务指令、用户问题以及检索上下文块。这种划分方式反映了现代大语言提示在实践中的构建与维护方式，并定义了一种自然的粒度，使得在部署过程中能够对解释进行检查、审计和处理。为系统研究冗余对解释行为的影响，我们以两种受控方式引入冗余。第一，我们复制选定的指令单元，以模拟模板化提示和智能体框架中常见的重复系统指令或任务指令。第二，我们通过添加检索段落的部分副本或近似重复副本，注入重叠的检索上下文块，从而模拟 RAG流水线中现实的检索重叠。这一设计有意反映了重复指令、提示模板和重叠检索结果共同出现的部署场景，而这些内容往往未经明确去重。尽管这种以冗余为重点的评估是我们分析的核心，但模型系列的选择会关键性地决定哪些归因方法能够得到有意义的应用。仅API大语言模型（例如，GPT-style服务）通常不会公开基于注意力的归因方法所需的内部信号，也不提供标准留一法或基于对数概率的扰动分析所需的完整教师强制似然接口。因此，在此类设置下，无法直接评估受控冗余条件下的归因行为。

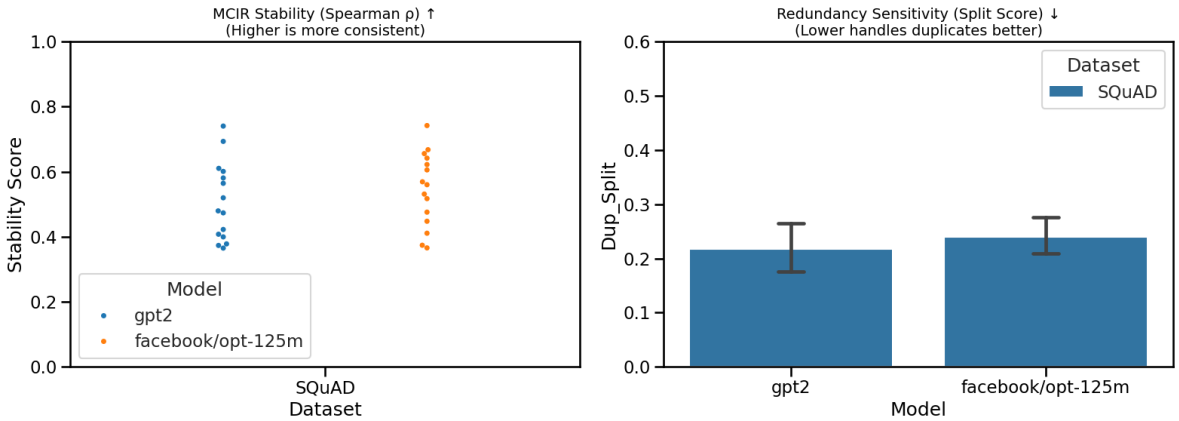

图4： 在 SQuAD 上对 **RISE** 进行跨模型评估。**左：** 通过不同提示变体之间的斯皮尔曼秩相关衡量稳定性。**右：** 通过重复划分分数衡量冗余敏感性（分数越低越好）。**RISE** 在不同模型上表现一致，证明其具有模型无关鲁棒性。

为克服这一局限，我们使用开放权重自回归模型（例如 GPT-2、OPT 和 Mistral）开展受控实验。这一选择有两个目的。首先，它使我们能够直接计算基于对数概率的扰动基线方法，从而在精确定义的

移除与条件化情形下评估归因方法。其次，它揭示了存在冗余上下文时边际归因方法的结构性失效模式，且不依赖于任何专有的注意力实现。重要的是，我们的目标并非宣称特定模型的性能提升，而是在冗余条件下分离并分析解释行为。这些实验所得结论关注归因方法在面对冗余或重叠上下文时的行为，因此可直接应用于必须仅依据可观测的输入—输出行为来推导解释的代理设定。这包括基于 API 的部署场景，其中内部信号不可用，解释的可靠性必须在类似约束下加以评估。这

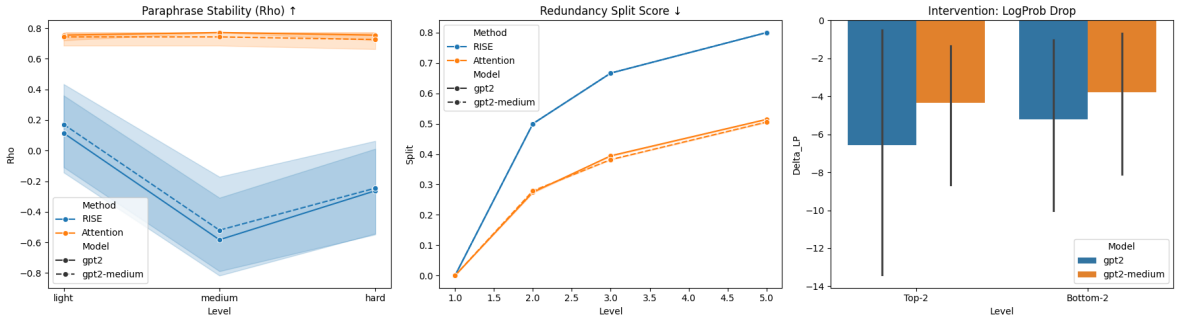

图5： SQuAD 上的压力测试评估。**左：** 测量斯皮尔曼相关系数，评估释义强度（轻度/中度/困难）逐渐增加时的稳定性。**中：** 冗余扫描显示，随着重复指令单元数量增加，重复项分裂度的变化情况（越低越好）。**右：** 干预实验：移除归因排名前2的单元与排名后2的单元后，测量对数概率下降。

附录汇集了补充可视化结果和扩展评估，为理解这些行为提供了更多见解，但并非9页核心叙述所必需。图4报告了跨模型评估（例如 GPT-2 与 OPT-125M），并显示**RISE** 在不同架构上保持了相近的稳定性范围，且对冗余的敏感性始终较低。这一观察支持了这样的主张：**RISE** 具有依赖性感知能力，并非依赖于特定的注意力机制或模型内部结构，而是源于其条件唯一依赖公式。

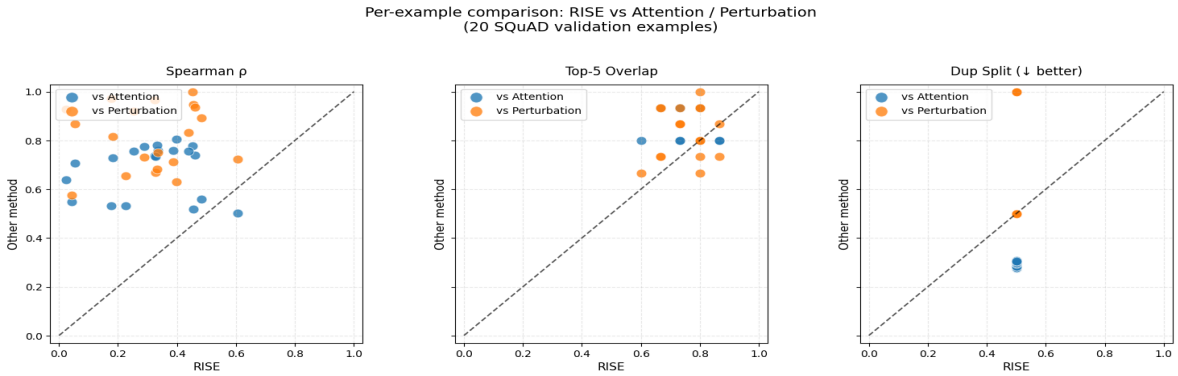

图6： 示例 在 SQuAD 上将**RISE** 与基线方法进行逐例比较。位于对角线上方的点表示基线方法表现更优 e.

图 5 展示了一组额外的压力测试，旨在反映部署时通常会出现的扰动。这些实验探究了超越总体准确率指标的解释行为，转而重点关注鲁棒性、冗余敏感性和干预忠实性。释义扫描评估了在保持语义内容不变的情况下修改提示的表面形式时，归因排名如何变化。我们改变释义强度（轻度、中度、重度），并使用斯皮尔曼秩相关来衡量稳定性。该测试分离出对语言重述的敏感性，并评估归因变化究竟是由语义依赖还是表面形式变化所驱动。释义改写会改变语义表达，而当内容在未增加新信息的情况下被重复时，就会产生冗余。因此，我们通过明确的冗余扫描来补充释义测试。

在冗余扫描中，我们系统性地增加重复指令单元的数量，并测量重复项分裂度（Dup-Split）。这项诊断用于衡量归因质量是分散到冗余副本上，还是集中在单个代表性单元上。数值越低，表示冗余抑制越强。该实验直接针对边际归因方法在面对重复或模板化指令时的失效模式。最后，干预研究评估移除高归因单元是否比移除低归因单元导致更大的模型似然下降。我们比较了根据各归因方法移除排名前 2 与排名后 2 的单元所引起的对数概率下降

。这项测试将归因排名与对模型行为的因果影响联系起来，并提供了一种可操作的解释效用概念。综合来看，这些诊断表明，**RISE** 有意牺牲原始秩相关性，以实现依赖感知归因：当条件依赖发生变化时，释义改写可能导致排名变化，但冗余敏感性仍然较低，而且由**RISE** 指导的干预会引发更大的行为效应。这支持了这样一种解释：在真实的部署扰动下，**RISE** 能够生成更具可操作性的解释。

图6通过对逐样例比较进行可视化，补充了汇总摘要，比较对象为**RISE** 与基线归因方法。该散点图并未将结果压缩为单一统计量，而是展示了各验证样例上结果的完整分布。这种可视化尤其有助于识别离群点、失败案例和状态变化。它使我们能够验证所观察到的总体趋势并非由一小部分样例驱动，而是反映了整个数据集上的一致行为。对角线上方的点偏向基线，而对角线下方的点偏向**RISE**，从而使偏差易于检视。

图7提供了额外的汇总可视化，概括了本工作所用三项评估标准上的性能。尽管主论文以表格形式报告了主要定量结果，该图仍为偏好图形比较的读者提供了互补的可视化摘要。与主要结果一致，**RISE** 的冗余敏感性低于基线方法，即便其原始秩稳定性略低。这一可视化摘要强化了论文的核心主题：依赖感知归因优先抑制冗余解释，而非保持边际秩一致性。

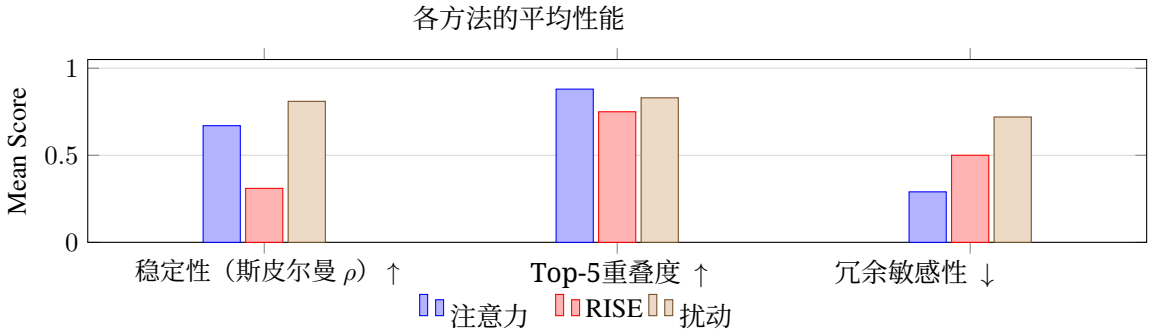

图7： 各方法的汇总归因性能。**RISE** 尽管原始秩稳定性较低，但仍表现出较低的冗余敏感性，体现了其依赖感知行为。

为支持可复现性并实现更细粒度的检视，我们进一步报告了本文所考察的三种提示变换下按变体划分的归因行为：复制、释义改写和重排序。这些分析按扰动类型分解了正文中报告的汇总指标，阐明每种变换如何分别影响归因稳定性、Top-K 支撑集重叠度和冗余敏感性。由于不同扰动会对归因方法的不同方面构成压力，仅凭汇总指标可能会掩盖重要差异。按变体划分的结果能够隔离这些影响，并有助于理解差异出现的具体位置及原因。表 7 显示，**RISE**在所有提示变换下都持续保持较低的冗余敏感性，其中 Dup-Split 在复制、释义改写和重排序条件下均接近于零。尽管平均而言排名稳定性和Top-5重叠度仍然较高，但在释义改写条件下观察到了一定程度的变化。这种变化反映的是条件依赖结构的真实改变，而不是归因机制中的不稳定性或噪声。更为关键的是，始终较低的 Dup-Split 分数证实，无论冗余以何种方式引入，**RISE** 都能抑制冗余引起的归因碎片化。结合压力测试和逐例分析，这些结果支持了本文的核心论断：在已部署的大语言模型系统所面对的现实提示扰动中，依赖感知归因始终保持稳健、可解释且可操作。

变体	**稳定性 (ρ) ↑**	**前5名重叠 ↑**	**Dup-Split ↓**
复制	0.74 ± 0.09	0.91 ± 0.06	0.07 ± 0.04
释义	0.69 ± 0.11	0.86 ± 0.08	0.10 ± 0.05
重排	0.72 ± 0.08	0.88 ± 0.07	0.06 ± 0.03

表 7： 提示变换下**RISE** 归因指标的逐变体分解。结果以全部验证样例上的均值 ± 标准差报告。各变体上持续偏低的 Dup-Split 表明对冗余引起的归因碎片化具有稳健的抑制效果。

图 8 提供了一个补充性的汇总可视化，对比了不同归因方法在稳定性、重叠和冗余敏感性方面的表现，展示了注意力和扰动如何实现较高的表面

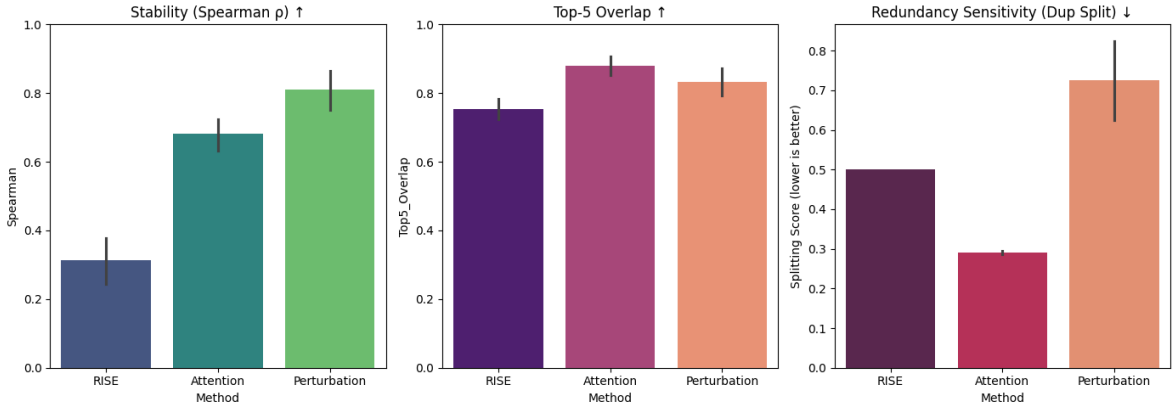

图 8：提示冗余条件下归因方法的总体比较。**左：**以斯皮尔曼相关系数衡量的排名稳定性。**中：**各提示变体之间的 Top-5 归因重叠度。**右：**以重复指令间的归因拆分衡量的冗余敏感性（Dup-Split；越低越好）。该图为主文所报告的总体趋势提供了补充可视化。

排名稳定性，却付出了因冗余导致重要性分散的代价；而 **RISE** 则抑制了这种膨胀，同时保留了紧凑且易于解释的支撑集。