arXiv:2611.31190v1 [cs.AI] 23 Nov 2026

把握何时弃答：文本到 SQL 的保形置信度估计

Avinash Kumar Singh
Domyn 海得拉巴，印度
avinash.kumarsingh@domyn.com

Bhaskarjit Sarmah
Domyn 新德里，印度
bhaskarjit.sarmah@domyn.com

Stefano Pasquali
Domyn 纽约，美国
stefano.pasquali@domyn.com

摘要

文本到 SQL 系统经常会返回能够成功执行、却回答了错误问题的查询。在部署时检测这类错误很困难，因为执行正确性是根据黄金查询定义的，而黄金查询只存在于标注数据中。我们将文本到 SQL 的置信度估计表述为概率性查询验证：其中的潜在正确性变量在校准时可通过黄金 SQL 执行观测到，而在部署时则是潜在的。我们的估计器将生成与验证分开。生成器大语言模型编写并执行查询，另一个不同的评判器大语言模型则评估该查询及其执行结果。评判器的验证分数和基于执行的对齐分数，与生成器以自然语言表述的置信度、词元级不确定性及 SQL 结构特征一起输入随机森林，并通过两次顺序的大语言模型调用完成。两个校准步骤提供有限样本保证。Learn-then-Test 阈值以概率 $1-\delta$ 确保已接受查询的错误率低于用户设定的预算。保形 p 值则界定错误弃权率。在 Spider 上，我们使用 GPT-OSS-120B 作为生成器，Gemma-4-31B 作为评判器。该估计器的 AUARC 为 0.899。若不弃答，22.6% 的查询是错误的。在 15% 的风险预算 (δ=0.1) 下，所有 10 个重采样划分都能认证一个阈值；系统平均接受 52.4% 的查询，观测到的错误率为 9.1%。五种不同的评判器模型得出的结果相近。在以 Olmo-3-32B 为生成器的 Spider 受控比较中，该估计器的 AUARC 为 89.5，而最强的基于logit的基线为 81.2。将该方法应用于 BIRD 且不重新训练时，置信度分数仍具有参考价值（AUROC 为 0.72，而 Spider 上为 0.79），但正如在缺乏可交换性的情况下所预期的那样，风险保证无法迁移。

1 引言

大语言模型在文本到 SQL 基准测试中取得了很高的执行准确率（Yu 等，2018；Li 等，2023），但其中许多错误表现为查询可以执行，却返回合理但错误的结果（Gong 等，2025；Somov & Tutubalina，2025）。这类静默失败不会产生任何
执行时信号，且具有大型模式和领域专属语义的企业工作负载让这类错误更难被发现（Lei 等，2025；Chen 等，2025b）。因此，部署的系统需要为每个查询生成一个置信度分数，以决定是返回结果还是弃答。

难点在于，正确性是相对于黄金查询定义的。当生成查询的执行结果与黄金 SQL 的执行结果一致时，该查询才是正确的。在带标注基准上，可以精确验证这一点；但在部署时不存在黄金 SQL，因此正确性是潜在的。执行成功与否可以观测到，但它并不等同于正确性。因此，任何置信度都必须根据生成和执行时可用的信号推断，而任何弃权规则都必须在标注数据上拟合，再应用于没有标注的数据。

现有的文本到 SQL 置信度估计器大多依赖单一类型的信号。基于采样的方法将 K 采样输出之间的一致性视为正确性的替代指标（Wang 等人，2023；Kuhn 等，2023）。这类方法需要 K 次大语言模型调用，而且，如果模型系统性地误读模式，就可能生成一致但错误的查询（Stengel-Eskin & Van Durme，2023；Ramachandran & Sarawagi，2024）。基于Logit的方法会聚合词元概率，也可能按 SQL 子句进行聚合（En- tezari Maleki 等人，2026；Ramachandran & Sarawagi，2024）。这类方法只需单次处理，但词元概率反映的是下一词元不确定性；即使查询流畅却意图错误，这些概率仍可能很高（Stengel-Eskin & Van Durme，2023；Desai & Durrett，2020）。这些方法通常通过区分度和校准进行评估；除少数例外（Chen 等人，2025年a）外，弃权阈值并未与明确说明的统计性质挂钩。

我们将置信度估计视为概率性查询验证，并将生成与验证的职责分开。生成器大语言模型编写查询并报告自身置信度。随后，另一位评判器大语言模型会验证该查询，并检查其执行结果是否回答了问题。使用独立的评判器，意味着作出判断的不是那个因自身推理而产生错误的模型。我们将评判器的验证结果和基于执行的对齐分数，与生成器的言语化置信度、词元级不确定性及 SQL 结构特征结合起来。这些信号需要依次调用大语言模型两次，每个模型调用一次，此外还需进行本地计算。我们使用随机森林融合这些信号，并在正确性可观测的数据上进行训练。随后，我们采用先学习后检验（Angelopoulos 等，2025），在带标签的校准集上确定接受阈值。由此可保证，以高概率使已接受查询中的错误率低于用户选定的上限。根据同一组分数计算出的共形 p 值，还能界定系统对正确查询弃答的比例。

在 Spider 上，以 GPT-OSS-120B 为生成器、Gemma-4-31B 为评判器时，每个重采样划分都能认证 15% 的风险预算。在该预算下，系统接受了 52.4% 的查询，观测到的错误率为 9.1%；相比之下，不弃答时的错误率为 22.6%。在生成器相同的受控比较中，该估计器的 AUARC 为 89.5，高于 SAC-Product 的 81.2 和五样本自一致性的 82.1。我们的贡献如下：

- **形式化**。 我们针对潜在的执行正确性，对文本到 SQL 的置信度估计进行形式化，区分校准时可观测的内容与部署时的潜在内容。我们说明哪些选择性预测指标具有保证，哪些属于经验值（第 3 节）。
- **估计器**。 我们提出一种生成器-批评器估计器。一个独立的评审模型会验证生成器给出的查询及其执行结果，随机森林则融合评审模型的判断与生成器侧信号。消融实验表明，大语言模型派生信号承载了大部分信息，因此，不具备 logit 访问权限的变体仍能保留大部分性能（第 4 节、第 5.4 节）。
- **校准式弃答**。 我们使用先学习后检验来选择接受阈值，从而以高概率控制已接受查询中的错误率。我们还推导了错误弃权的保形界，并刻画由此产生的覆盖率–风险权衡（第 3.1 节、第 5.2 节）。
- **评估**。 我们将五种评判器模型与自我批评进行比较。我们还报告了受控基线比较、四种生成器骨干网络的结果、10 个随机种子的重采样、向 BIRD 和 NL2SQL-BUGs 的零样本迁移，以及一种不具备 logit 访问权限的变体（第 5 节）。

2 相关工作

文本到 SQL 的错误检测与置信度。以往研究从模型概率及其重新缩放中估计查询级置信度（Ramachandran & Sarawagi，2024；Stengel-Eskin & Van Durme，2023），从 SQL 感知型 logit 聚合和基于执行的落地中估计（En- tezari Maleki 等，2026；Somov & Tutubalina，2025），或通过带多变量 Platt 校准的子句频率进行估计（Liu 等，2025年a）。SQLens 结合数据库和大语言模型信号，以检测并纠正子句级错误（Gong 等，2025）；SQLFixAgent 通过多智能体一致性检查修复查询（Cen 等，2025）；NL2SQL-BUGs 则提供了一个用于语义错误检测的基准测试（Liu 等，2025b）。这些方法会生成分数或纠正结果。我们将多个信号系列结合为一个估计器，并附加一项具有明确校准性质的弃权规则。

大语言模型的不确定性估计。 白盒方法使用词元级概率（Hendrycks & Gimpel，2017；Malinin & Gales，2021）。黑盒方法使用采样一致性或语义等价性（Wang 等人，2023；Kuhn 等，2023；Farquhar 等人，2024），其底层假设由 Xiao 等人（2025）进行检验；或者引出语言化置信度（Kadavath等，2022年；Xiong等，2024年；Tian 等人，2023）。我们将语言化信号和 logit 信号用作输入特征，而非独立的分数。

选择性预测与保形方法。 选择性分类会对低置信度输入弃答，从而降低已接受预测中的错误率（Geifman & El-Yaniv，2017）；Wen 等人（2025）综述了大语言模型中的弃答。TrustSQL 使用基于惩罚的评分来评估文本到 SQL 的可靠性，并对弃答给予奖励（Lee 等人，2024）；Chen 等人（2025年a）则使用保形预测，在模式链接过程中弃答。保形预测在可交换性假设下提供有限样本保证（Vovk 等，2005；Angelopoulos 与 Bates，2023），并已应用于语言生成（Quach 等人，2024）以及无法访问 logit 的大语言模型（Su 等人，2024）。我们使用先学习后检验（Angelopoulos 等，2025）来控制选择性错误率，该方法将阈值选择表述为多重假设检验。我们的共形置信度采用异常值检测构造（Bates et al.，2023），并以正确查询为参考类，从而控制错误弃权。

3 问题表述

设定。 令 $x = (q, s)$ 将自然语言问题 q 与数据库模式 s配对，并令 D 为数据库实例。全文中，s表示模式，$\hat{s}$ 表示生成的 SQL 查询，s^* 表示标准 SQL 查询。涉及两个模型。生成器 θ_G 定义了自回归分布$P_{\theta_G}(\hat{s} \mid x) = \prod_{t=1}^{T} P_{\theta_G}(\hat{s}_t \mid \hat{s}_{<t}, x)$，其中 T 是$\hat{s}$中的词元数量。解码过程返回单个查询$\hat{s}$，其词元 $\tau = (t_1, \ldots, t_T)$的概率为 $\pi_1, \ldots, \pi_T$。评判器 θ_C是一个不同于 θ_G的大语言模型，它负责评估$\hat{s}$，但不生成该查询。

执行正确性作为潜变量。 令 Exec$(\hat{s}, D)$表示确定性执行。执行正确性变量为

$$Z(x, \hat{s}) = \mathbb{1}[\mathrm{Exec}(\hat{s}, D) = \mathrm{Exec}(s^*, D)]\,. \tag{1}$$

对于只读 SELECT 查询，始终可以获得结果 $r = \mathrm{Exec}(\hat{s},\ D)$。 Z 之所以是潜在的，是因为部署时无法获取s^*，而不是因为执行不可行。对于带标签的数据，Z 可通过执行匹配精确计算，我们据此获得训练标签和校准标签。

置信度函数。 置信度函数 $C\colon \mathcal{X} \times \mathcal{Y} \to [0, 1]$ 仅使用部署时信号，将 $(x,\ \hat{s})$ 映射为一个分数：q、s、$\hat{s}$、词元概率 $\{\pi_t\}$、执行结果 r、生成器语言化的置信度，以及评判器的判断。由于部署时无法观测到 Z，我们通过区分度（将正确查询排在错误查询之前）、校准（分数与经验值正确性的一致程度）和选择性预测，在带标签的测试数据上评估 C。

3.1 选择性预测及其保证

给定置信度分数 $S(x,\ \hat{s})$ 和阈值 λ，选择性预测器定义为

$$\mathcal{A}_\lambda(x) = \hat{s} \ \text{ if } S(x, \hat{s}) \geq \lambda, \qquad \mathcal{A}_\lambda(x) = \perp \text{ (abstain) otherwise.} \tag{2}$$

我们使用四个总体量：

$$\mathrm{Coverage}(\lambda) = \mathbb{P}(S \geq \lambda)\,, \qquad \mathrm{Risk}(\lambda) = \mathbb{P}(Z = 0 \mid S \geq \lambda)\,, \tag{3}$$

$$\mathrm{FAR}(\lambda) = \mathbb{P}(S < \lambda \mid Z = 1)\,, \qquad \mathrm{TAR}(\lambda) = \mathbb{P}(S < \lambda \mid Z = 0)\,. \tag{4}$$

FAR（错误弃答率）是被弃答的正确查询所占的比例，而 TAR（正确弃答率）是被弃答的错误查询所占的比例。给定基础准确率 $\rho = \mathbb{P}(Z{=}1)$，即可确定另外两个量：

$$\mathrm{Coverage}(\lambda) = \rho\,(1 - \mathrm{FAR}) + (1 - \rho)(1 - \mathrm{TAR}), \qquad \mathrm{Risk}(\lambda) = \frac{(1 - \rho)(1 - \mathrm{TAR})}{\mathrm{Coverage}(\lambda)}. \tag{5}$$

$\hat{}$ 以下两种校准规则均使用第 4.4 节中的融合估计 $p(x,\ \hat{s}) = \mathbb{P}(Z{=}1 \mid \mathbf{x})$，该估计在单独的数据划分上训练，并使用大小为 m 、观测到 Z_i 的留出校准集 $\mathcal{C}$。无法执行的查询一律拒绝。

共形置信度与错误弃权。令 $\mathcal{C}_1 = \{i \in \mathcal{C} : Z_i = 1\}$，其中 $m_1 = |\mathcal{C}_1|$，并定义非一致性分数 $\eta = -\log p$。校准置信度即共形p值

$$\hat{c}(x, \hat{s}) = \frac{1 + \left|\{i \in \mathcal{C}_1 : \eta_i \geq \eta(x, \hat{s})\}\right|}{m_1 + 1}, \tag{6}$$

wit当 $\hat{s}$ 无法执行时的 $\mathrm{h}\,\hat{c} = 0$。保形规则 使用 $S = \hat{c}$ 和 $\lambda = 1 - \alpha$ 来进行操作 g level $\alpha \in (0, 1]$。

命题 1（错误弃权的有限样本界）。假设正确的测试查询 $(x,\hat{s})$ 与$Z = 1$ 可交换，且与 $\{(x_i,\hat{s}_i)\}_{i\in\mathcal{C}_1}$可交换，条件是训练估计器为 p。那么，对于每个 $t \in [0, 1]$，$\mathbb{P}(\hat{c} \leq t \,|Z = 1) \leq t$。因此，保形规则满足

$$\mathrm{FAR}(1 - \alpha) \;\leq\; 1 - \alpha, \qquad \textit{equivalently} \qquad \mathbb{P}\big(\hat{c} \geq 1 - \alpha \mid Z = 1\big) \;\geq\; \alpha, \tag{7}$$

其中，概率是针对校准集和测试查询而言的。

这是用于异常检测的共形 p 值的有效性（Angelopoulos & Bates，2023，第 4.4 节；另见 Bates et al.，2023），其中正确查询作为参考类（证明见附录 E）。根据公式 5，保形规则接受约占正确查询的 α，因此，按构造，较小的 α会在覆盖率较低时带来较高的精确率。它并不限定风险。

风险控制。为界定已接受查询中的错误率，用户设定风险预算 ε 和失败概率 δ，我们采用先学习后检验方法选取阈值（Angelopoulos 等，2025）。设 Λ为在校准前固定的 p 上的阈值网格。对于每个 $\lambda \in \Lambda$，令 n(λ)为满足 $p_i \geq \lambda$的校准查询数量，k(λ)为其中满足 $Z_i = 0$的查询数量。我们使用二项p值 $u_\lambda = \mathbb{P}$(Bin(n(λ) ε) $\leq k$(λ))检验 H_λ： Risk(λ) $>\hat{}\,\varepsilon$，并设 $\lambda = \min\{\lambda \in \Lambda : u_\lambda < \delta/|\Lambda|\}$。ˆ 风险控制规则 使用 $S = p$ 和 $\lambda = \lambda_\diamond$。如果没有 λ 符合条件，则对每个查询都弃答。

命题 2 (有限样本风险控制)。如果校准查询和测试查询独立同分布，且 p 是在与 $\mathcal{C}$不相交的数据上训练的，那么

$$\mathbb{P}_{\mathcal{C}}\big(\mathrm{Risk}(\hat{\lambda}) \leq \varepsilon\big) \;\geq\; 1 - \delta. \tag{8}$$

在给定 n(λ)的条件下，k(λ)服从成功概率为 Risk(λ) 的二项分布，因此 u_λ 是H_λ 的有效 p 值（Angelopoulos 与 Bates，2023，第 5.5 节）。Bonferroni 校正控制了为风险超过 ε 的任何阈值提供认证的概率（附录E）。两个命题仅需在校准集上使用黄金SQL；在部署阶段，只有当部署查询遵循校准分布时才成立（命题 1 要求可交换，命题 2 要求独立同分布）。风险受控规则下的覆盖率取决于 p 区分正确查询与错误查询的能力。校准查询数量较少时，较为严格的预算可能无法得到认证；在这种情况下，该规则会对每个查询都选择弃权。

4 方法

该估计器使用两个承担不同角色的大语言模型（图1）。生成器 θ_G 编写查询，评判器 $\theta_C \neq \theta_G$ 对其进行判断。阶段 1（生成器，大语言模型调用 1）生成带有言语化置信度和词元概率的$\hat{s}$，并执行$\hat{s}$。阶段2（评判器，大语言模型调用 2）验证$\hat{s}$ 并对其执行结果进行评分。第3阶段（本地）根据生成器的词元概率计算概率与结构特征。第4阶段（本地）融合所有特征，应用保形校准，并选择风险受控阈值。阶段2和第3阶段并行运行，因此该流水线会依次进行两次大语言模型调用，分别调用两个模型。表1列出了各特征族，以及生成每个特征的模型或阶段。

4.1 阶段 1：生成与执行（生成器）

生成器接收（q，s），其中模式包含样本行，以及一条生成提示词（附录A.1）。在给出答案之前，它会在内部检查查询是否与问题对齐、是否接地于模式、字面量值、SQLite语法、逻辑正确性、结果合理性以及极简性。其输出仅包含$\hat{s} = f_{\mathrm{gen}}$(q， s)和言语化置信度 $c_{\mathrm{gen}} \in [0, 100]$。提示词要求置信度反映查询最薄弱的关键方面，而非其语法有效性。生成器不会返回验证判定。判定仅由评判器给出。我们记录词元$\hat{s}$ 的概率 π_t，供第3阶段使用。随后，我们在 D 上执行$\hat{s}$，以获得 $r = \mathrm{Exec}(\hat{s},\ D)$。如果执行失败，则 Z=0必定成立，且查询会被赋予$\hat{c}$=0，不再进行后续处理。

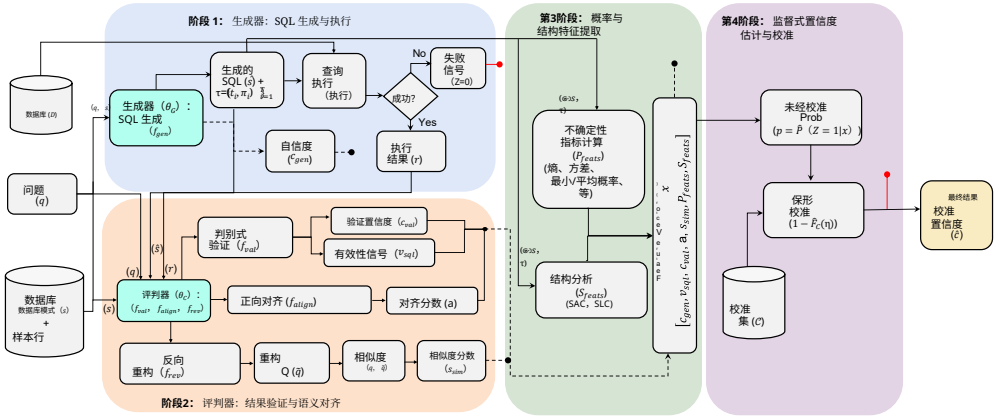

图 1：四阶段生成器-批评器估计器。阶段 1（生成器 θ_G，LLM 调用 1）生成$\hat{s}$及其词元和概率 τ 以及自信度 c_{gen}，然后在 D上执行$\hat{s}$ 。执行失败会设置 $Z{=}0$ 和$\hat{c}{=}0$。阶段 2（批评器 $\theta_C \neq \theta_G$，LLM 调用 2）接收（q、 s、$\hat{s}$、 r）。它返回带有置信度 c_{val}的有效性判定 v_{sql} 、对齐分数 a，以及 ˆ q与根据 r重构的问题 $\tilde{q}$ 之间的相似度 s_{sim} 。阶段 3（本地）根据生成器的（$\hat{s}$、 τ）计算不确定性和结构特征，并与阶段 2 并行运行。阶段 4（本地）将特征向量$\mathbf{x}$与随机森林融合，并使用离线标注的校准集$\mathcal{C}$ 将其后验概率映射为校准置信度$\hat{c}$ 。

4.2 阶段 2：批评器验证与基于执行的对齐（批评器）

评判器在单条提示词中接收（q， s， $\hat{s}$， r）（附录A.2）。它看不到 c_{gen} 或生成器的推理，也无法访问黄金SQL或黄金答案。它会根据问题评判执行结果，而不只是判断这些值看起来是否合理。言语化判断可以揭示词元概率无法体现的语义问题（Kadavath等，2022年；Xiong等，2024年）。

判别式验证。评判器会检查 r 是否正确且完整地回答了 q。检查内容涵盖所需的列和行、请求的操作或聚合、筛选条件和比较运算符、排序和前 k 项限制、分组粒度、日期和时间约束、返回值的一致性以及完整性。违反任何实质性要求都会使结果不正确。评判器会返回一个二元判定 $v_{\text{sql}} = f_{\text{val}}(q,\ s,\ \hat{s},\ r) \in \{0,1\}$ 以及对该判定的置信度 $c_{\text{val}} \in [0,100]$。

基于执行的对齐。评判器忽略 q，仅使用 r 和模式，重构出 r 所回答的最精确问题 $\tilde{q} = f_{\text{rev}}(s,\ r)$，且不会补全缺失的条件。它还会返回一个对齐分数 $a = f_{\text{align}}(q,\ s,\ \hat{s},\ r) \in [0,100]$，用于衡量结果在请求的信息、筛选条件、聚合、排序、分组、日期和完整性方面对问题 q 的回答效果。评判器还会返回语义相似度 $s_{\text{sim}} = \text{Sim}(q,\ \tilde{q}) \in [0,100]$。当 $v_{\text{sql}} = 0$ 时，提示会限制 a 和 s_{sim} 。较低的 a 或 s_{sim} 表明结果虽然主题相关，但回答的是另一个问题。

4.3 第3阶段：概率与结构性特征

根据生成器的词元概率 $\{\pi_t\}_{t=1}^{T}$ of$\hat{s}$，我们无需进一步调用大语言模型即可计算：

$$\text{FTC} = \prod_t \pi_t, \quad \pi_{\min} = \min_t \pi_t, \quad \mu_{\text{arith}} = \tfrac{1}{T}\sum_t \pi_t, \quad \mu_{\text{geom}} = \left(\prod_t \pi_t\right)^{1/T}, \tag{9}$$

$$H = -\sum_t \pi_t \log \pi_t, \quad \sigma^2 = \tfrac{1}{T}\sum_t (\pi_t - \mu_{\text{arith}})^2, \quad \ell_{\text{CLM}} = -\sum_t \log \pi_t. \tag{10}$$

我们还分别重新计算 SQL 关键字词元和非关键字词元上的前五项统计量，以免仅局限于标识符或字面量的不确定性被平均掉。我们加入概率范围和 13 箱概率直方图。词元概率无法区分模式标识符与其他词元，因此我们加入 SQL 感知特征（Entezari Maleki 等，2026；Somov & Tutubalina，2025）。令 $\mathcal{G}$ 将$\hat{s}$ 的词元划分为子句组（SELECT、WHERE、表标识符等），并令 $\mathcal{G}$ 表示命名模式元素的词元。我们计算

表1：$\mathbf{x}$中的特征、生成这些特征的模型和阶段，以及它们所针对的故障模式。 θ_G：生成器； θ_C：评判器。最后一行的完整内容列于附录表 7中。

类别	特征	来源	Dim	目标故障模式
语言化	c_{gen}	阶段 1， θ_G	1	生成器不确定性
	v_{sql}， c_{val}	阶段2， θ_C	2	结果违背问题
对齐	a, s_{sim}	阶段2, θ_C	2	合理但错误的结果
对数几率 (P_{feats})	FTC $\pi_{\min}, \mu_{\text{arith}}, \mu_{\text{geom}}, H, \sigma^2, \ell_{\text{CLM}}$	第3阶段，本地	7	词元级不确定性
SQL 结构（S_{feats}）	SAC_{geom}， SLC	第3阶段，本地	2	模式对齐错误
变体	子集统计量、范围、直方图, 模式表均值 , C3S	第3阶段，局部	27	局部化不确定性
总计			**41**	

，并令 τ_{sch} 表示命名模式元素的词元。我们计算

$$\text{SAC}_{\text{geom}} = \left(\prod_{g\in\mathcal{G}} \tfrac{1}{|g|}\sum_{t\in g}\pi_t\right)^{1/|\mathcal{G}|}, \qquad \text{SLC} = \left(\prod_{t\in\tau_{\text{sch}}}\pi_t\right)^{1/|\tau_{\text{sch}}|} \ (\text{SLC}{=}0 \text{ if } \tau_{\text{sch}}{=}\emptyset). \tag{11}$$

SLC 针对模式落地这一跨领域文本到SQL中的主要失败模式（Yu 等，2018；Li 等，2023）。模式表均值和子句一致性分数（C3S）共同构成这一类别。完整向量包含 41 个特征（表1；每个特征均列于附录B）。其中一些特征几乎重复，例如 $\ell_{\text{CLM}} = -\log \text{FTC}$，而树集成不受此类冗余影响。

4.4 第4阶段：监督融合与保形校准

ˆ 随机森林 f_{RF} 将$\mathbf{x}$映射到 $p = \mathbb{P}(Z{=}1 \mid \mathbf{x})$。它使用带标签查询进行训练，这些查询的 Z 通过执行匹配计算得出。森林无需归一化即可处理不同尺度的特征，并能利用特征交互，例如高 FTC 与低对齐度。没有任何单一特征能够很好地区分这两个类别（附录 C）。在一个独立的校准划分上，我们计算正确查询的 $\eta_i = -\log\ p_i$，并使用公式6将 p 映射到$\hat{c}$。由于$\hat{c}$ 是 p的非递减函数，校准后查询的排序（并列情况除外）保持不变；我们报告 p的 AUROC 和 AUARC。三个阶段的划分如下：训练和校准使用黄金SQL计算 Z。部署时仅使用生成器、评判器、第3阶段的局部特征、冻结的森林以及存储的校准分数。

5 实验

5.1 实验设置

数据和模型。 我们使用来自 Spider 开发集和测试集（Yu 等，2018）的 3,181 个问题–SQL对，涵盖 200 个数据库。移除无法执行其黄金SQL的问题–SQL对后，根据生成运行的不同，剩余 3,175–3,179 个。对于零样本迁移，我们使用 BIRD 开发集（Li 等，2023）（1,534 个样本）和 NL2SQL-BUGs（Liu 等人，2025b）（1,019 个正确查询和 999 个语义错误查询）。模型详情见附录G.1。

生成器和评判器模型。 第1阶段和第2阶段始终使用不同的模型，除非另有说明，评判器均为 Gemma-4- 31B-it。主配置使用 GPT-OSS-120B 作为生成器。四骨干研究使用 GPT-OSS-120B、Olmo-3-32B-Think、Qwen2.5-72B-Instruct 和 Llama-3.3-Nemotron-Super-49B-v1.5 作为生成器。受控比较和消融实验使用 Olmo-3-32B-Think。为检验评判器的选择，我们还使用 Qwen3.8-27B、DeepSeek-V4.1- Flash、Nemotron-3-Super 和 Qwen2.5-72B-Instruct 作为 GPT-OSS-120B 的评判器。作为参照，我们报告自我批评，即生成器兼任自身的评判器（$\theta_C = \theta_G$）。

实验协议。 Spider 按 Z分层划分为训练集（60%）、校准集（20%）和测试集（20%），例如分别包含 1,907 / 636 / 636 个查询。对于每种配置，都会使用根据该配置输出计算出的特征训练一个独立的随机森林，并通过在训练划分上进行 5 折交叉验证随机搜索来选择超参数（附录 D）。保形校准使用校准划分，所有指标均在测试集上计算。对于零样本迁移，在 Spider 上拟合的森林和

表 2：Spider 测试集划分上的评判器比较（n=636，生成器为 GPT-OSS-120B）。选择性指标采用 α=0.1 下的保形规则，命题 1 保证其期望假接受率不超过 ≤ 0.9。自我批评实验使用单独的生成轮次，因此其基础准确率有所不同。

	自我批评	评判器				
	($\theta_C = \theta_G$)	Gemma-4-31B	Qwen3.8-27B	DeepSeek-V4.1	Nemotron-3-S	Qwen2.5-72B
基础准确率 ρ	0.786	0.774	0.774	0.774	0.774	0.774
期望校准误差 ↓	0.149	0.158	0.155	0.156	0.142	0.166
AUROC ↑	0.781	0.785	0.840	0.811	0.779	0.826
AUARC ↑	0.913	0.899	0.922	0.916	0.901	0.915
覆盖率	0.077	0.079	0.063	0.080	0.074	0.082
条件准确率 = 1−风险	0.980	0.940	1.000	1.000	0.957	0.981
TAR	0.993	0.979	1.000	1.000	0.986	0.993
假接受率（界：≤ 0.9）	0.904	0.904	0.919	0.896	0.909	0.896

校准分数将不作更改地应用于 BIRD 和 NL2SQL-BUGs（第 5.6 节）。训练或校准时不会使用任何目标域数据。

指标。区分度：AUROC 和 AUARC（准确率-拒绝率曲线下面积，是我们的主要指标）。校准：期望校准误差（10 个等宽分箱）、自适应校准误差（等质量分箱）和布里尔分数。选择性预测：覆盖率、风险、真接受率和假接受率（公式 3–4），在测试集划分上估算；以及条件准确率 CondAcc $= 1 -$ 风险，即已接受查询中的准确率。对于风险控制规则，风险由命题 2 覆盖。对于保形规则，假接受率由命题 1 覆盖。除非另有说明，我们使用 $\delta = 0.1$ 和网格 $\Lambda = \{0.01, 0.02, \ldots, 0.99\}$。附录 F 给出完整定义。每次查询延迟见附录 G.9。

5.2 生成器–判别器结果

表 2 比较了五种判别器与自我批评。AUARC 范围为 0.899（Gemma-4-31B）至 0.922（Qwen3.8-27B），自我批评的结果为 0.913。这些差异与 10 个随机种子的变异性（第 5.5 节）处于同一量级，因此使用单独的判别器既能保持区分度，又能避免自我判断。根据 α=0.1 处的保形规则，每种配置接受 6.3–8.2% 的查询，其中 94.0–100% 的查询正确，假接受率（0.896–0.919）接近其 0.9 的界。部分配置见附录 G.4。对于搭配 Gemma 判别器的四种生成器骨干网络，AUARC 范围为 0.895 至 0.918（附录表 11）。

风险受控弃答。按照构造，α=0.1 处的保形规则只接受少量查询，因为它将正确查询的接受比例固定在约 α（第 3.1 节）。风险控制规则则采用一个最宽松的阈值，前提是校准数据能够证明风险不高于 $\leq \varepsilon$。表 3 报告了该规则在 10 个重采样划分上的结果。在 ε=0.15 处，八种配置的全部 80 个划分均能认证一个阈值，系统平均接受 38.8–64.1% 的查询。搭配 Gemma 判别器时，系统接受 52.4% 的查询，而保形规则下这一比例为 7.9%。在 ε=0.20 处，覆盖率达到 67.5–87.7%。在全部 194 次认证运行中，测试集风险从未超过 ε。只有 0–8 个划分（共 10 个）能够认证 ε=0.10 的预算。校准查询数量为 636 时，二项界对于严格端的小型接受集合而言过宽。因此，保形规则的低覆盖率反映的是规则本身，而非估计器（扫描结果见附录 G.3）。

5.3 与基线进行受控比较

表 4 中的所有方法均在 Olmo-3-32B-Think 上重新实现，并使用相同的数据划分和评估代码。基线方法包括 SAC-Product（Entezari Maleki 等，2026）和汇聚式令牌级（Ramachandran 与 Sarawagi，2024）（基于对数几率），自一致性（Wang 等人，2023），采用 K=3,5 次采样，以及单次思维链验证；后者使用相同的生成器和评判器调用，但不进行特征融合或保形校准。原论文报告的数值采用了其他骨干网络和实验协议。我们仅在附录 G.10 中列出这些数值以供参考。

我们的估计器取得了 89.5 的 AUARC，而 SAC-Product 为 81.2，自一致性为 82.1（K=5）。对测试集划分进行自助法抽样（B=1，000）后，得到的 95% 区间为 (86.7，91.9)，其下界高于表中所有基线。单次思维链验证使用相同的两次调用，取得 78.9。因此，这 10.6 分的差距源于特征融合，而非增加大语言模型

表 3：Spider 上 10 个重采样划分的风险受控弃答（命题 2，δ=0.1）。认证：阈值已获认证的数据划分；否则，系统对每个查询均弃答。覆盖率：覆盖率（%），在全部 10 个数据划分上取均值，未认证划分计为零。风险：已接受查询中的测试划分误差（%），在已认证划分上取均值（最大值）。判别器各行使用生成器 GPT-OSS-120B；生成器各行使用 Gemma 判别器。 *四骨干研究的生成运行。

	$\varepsilon = 0.10$			$\varepsilon = 0.15$			$\varepsilon = 0.20$		
	认证	覆盖率	Risk	认证	覆盖率	Risk	认证	覆盖率	Risk
判别器：Gemma-4-31B	0/10	0.0	—	10/10	52.4	9.1 (11.7)	10/10	74.2	14.1 (16.1)
评判器：Qwen3.8-27B	7/10	27.6	4.7 (8.0)	10/10	64.1	9.9 (14.6)	10/10	79.8	14.2 (16.0)
评判器：DeepSeek-V4.1	6/10	21.9	4.9 (9.1)	10/10	55.4	8.7 (11.9)	10/10	75.3	14.0 (15.9)
评判器：Nemotron-3-Super	6/10	19.6	5.0 (8.2)	10/10	54.2	9.3 (11.4)	10/10	72.7	14.6 (17.2)
生成器：Olmo-3-32B	1/10	1.9	7.4 (7.4)	10/10	38.8	8.2 (12.1)	10/10	67.5	14.1 (17.8)
生成器：Qwen2.5-72B	4/10	14.4	5.9 (7.8)	10/10	62.2	9.7 (12.6)	10/10	87.7	15.5 (17.0)
生成器：Llama-49B	2/10	7.2	7.0 (7.1)	10/10	51.9	8.6 (11.5)	10/10	74.1	14.3 (16.0)
生成器：GPT-OSS-120B*	8/10	23.4	4.1 (6.9)	10/10	57.1	9.3 (12.7)	10/10	76.6	14.2(16.4)

表 4：Spider 上的受控比较。所有方法均使用 Olmo-3-32B-Think 作为生成器，并共享数据划分和评估代码。 †：基于Logit。调用次数：每次查询的大语言模型调用次数。我们流水线的每次查询延迟见附录 G.9。

方法	**调用次数**	**AUARC**	**AUROC**	**期望校准误差**
自一致性（K=3）	3	79.3	68.4	0.198
自一致性 (K=5)	5	82.1	71.2	0.183
单次思维链验证	2	78.9	63.6	0.030
SAC-Product†(Entezari Maleki 等，2026)	1	81.2	72.4	0.161
汇聚式令牌级† (Ramachandran & Sarawagi，2024)	1	79.7	70.8	0.174
Ours	**2**	**89.5**	**81.6**	0.126

调用次数。使用五次调用的自一致性仍低 7.4 分，这与样本间的一致性无法检测系统性错误的情况相符（Kuhn 等，2023；Ramachandran & Sarawagi，2024）。单次思维链验证的期望校准误差最低，但对查询的排序效果较差；我们的方法旨在优化排序并控制风险。

5.4 消融研究

表5每次移除一个特征族，并重新训练森林。移除语言化或对齐特征族，在 10 个随机种子上会使 AUARC 分别降低 0.9 和 1.1 个百分点。移除 logit 特征族或结构特征族所导致的 AUARC 变化小于随机种子波动；在单次划分中，移除 logit 特征族甚至会使其升高。采用单独的评判器时，源自大语言模型的特征族承载了大部分信号。保形校准不会改变排序（它是一种单调映射），但会提供命题1和命题2中的阈值。由于 $\hat{c}$ 是共形p值，而非概率，我们仅报告 p 的期望校准误差。

在表 6 中，融合三个语言化分数使 AUARC 从 78.9 提升至 84.5；基于词元的特征和结构特征增加了 3.4 个百分点，对齐增加了 1.6 个百分点。

无 Logit 变体。 封闭式 API 通常不会公开 logits。仅保留五个大语言模型派生分数（不含基于词元的特征），AUARC 为 88.6，比完整模型低 0.9 个百分点（10 个随机种子的波动为 1.6 个百分点），比 SAC-Product 高 7.4 个百分点；而 SAC-Product 本身需要 logits。我们模拟了这种设置，但未直接评估封闭式 API 模型。

5.5 随机种子波动与信号可靠性

在超参数固定的情况下，我们对表 3 中的全部八种配置（附录 G.5）的数据划分进行了 10 次重新采样。每种情况下，10个随机种子的均值 AUARC 与单次数据划分结果的差值均不超过 0.013。对于 Gemma 判别器，AUARC 为 0.900 ± 0.010。单次数据划分的 AUROC 与 10 个随机种子的均值最多相差 0.029，因此不应过度解读这一幅度的 AUROC 差异。在 α=0.1 处，均值假接受率为 0.889–0.907，接近命题 1 的界。每个大语言模型派生分数都有

表5：留一族消融实验（生成器 Olmo-3-32B-Think，评判器 Gemma-4-31B，Spider）。10 个随机种子：重采样数据划分上的均值 ± 标准差。森林后验概率的期望校准误差 p。

配置	AUARC	10 个随机种子	ECE
完整模型	89.5	88.5 ± 0.8	0.126
无语言化	87.6	87.6 ± 0.8	0.124
无对齐	87.8	87.4 ± 0.8	0.134
无对数几率 P_{feats}	90.5	88.3 ± 0.9	0.143
无结构性 S_{feats}	89.0	88.1 ± 0.8	0.122
无 Logit（5 个大语言模型评分）	88.6	86.9 ± 1.2	0.187
SAC-Product（基线）	*81.2*	—	*0.161*

表 6：从 Single CoT Verify（相同设置）逐步增加内容，每行都重新训练森林。所有新增项均为局部改动。

配置	AUARC	Δ
单次思维链验证（2 次调用）	78.9	—
基于语言化分数的森林	84.5	+5.6
+Logit／结构特征	87.9	+9.0
+对齐（完整）	89.5	+10.6

一个独立的 AUROC，在全部四个生成器和全部四个判别器中均介于 0.525 和 0.771 之间，高于 0.5 的随机水平，但低于森林的 AUROC（0.765–0.840）。附录 G.7 分析了判别器未能识别的错误。

5.6 零样本跨数据集迁移

我们将针对 Spider 拟合的森林和校准分数应用于 BIRD 和 NL2SQL-BUGs，不进行重新训练或重新校准。目标数据与 Spider 并非可交换，因此两个命题都不适用。

BIRD。 BIRD 的连接更复杂、嵌套聚合更多，模式也比 Spider 更长，而且生成的查询中只有 34.2 % 正确。该估计器的 AUROC 为 0.724（域内为 0.791），AUARC 为 0.515，而随机排序和完美排序对应的数值分别为 0.343 和 0.709。其期望校准误差低于 Spider 上的数值（0.091 对 0.133），反映的是基率，而非校准效果更好（附录 G.8）。在 Spider 上认证的风险阈值无法直接迁移：在 ε=0.15 处观察到的 BIRD 风险为 30.1%，这说明了命题 2 为何要求数据独立同分布。

NL2SQL-BUGs。 该基准测试提供由其他系统编写的 SQL，因此仅适用阶段2。仅使用我们的阶段*2*评审器，在一个开放的 *31B* 模型和执行依据的支持下，便能在 *NL2SQL-BUGs* 上达到 *77.4%* 的准确率，与 *Liu*等人（*2025b*）报告的最佳模型（*76.8%*）相当，尽管提示词、模型和输入各不相同。在 Spider 上训练的、基于评审器得分的随机森林迁移效果较差（F1分数为 0.708，而判定为 0.779），因为缺少生成器侧特征，且错误率存在差异。

6 结论

我们研究了文本到 SQL 的置信度估计问题：在标注数据上可以观测执行正确性，但在部署时该信息是潜在的。生成器编写并执行查询，另一个评审模型对其进行验证，随机森林则融合两个模型的信号。Learn-then-Test 阈值以高概率控制已接受查询中的错误率，而共形p值则约束错误弃权率。以 GPT-OSS-120B 作为生成器、Gemma-4-31B 作为评审器时，在 Spider 的每个重采样划分上都能认证 15% 的风险预算，并且系统接受 52.4% 的查询，观测错误率为 9.1%。在共享生成器的受控比较中，该估计器相较于最强的基于logit的基线，AUARC 提升了 8.3 个百分点。五种评审模型取得了相近的结果，而不具备 logit 访问权限的变体使 AUARC 下降了 1.6。在 BIRD 上，Spider 训练的估计器无需重新训练仍能区分正确查询与错误查询（AUROC 为 0.72），但 Spider 的风险阈值不再成立。

局限性。**(1) 保证的适用范围**。两个命题都假设部署查询遵循校准分布，在分布偏移下均不成立。覆盖率无法得到保证，而且对于 636 个校准查询，通常无法认证 10% 的风险预算。**(2) 对大语言模型判断的依赖**。规模最大的特征族来自大语言模型判断。单独的评判器可以避免自我判断，但评判器本身也是大语言模型，而它误读的错误仍然很难被检测到（附录 G.7）。**(3) logit 访问权限和开销**。完整模型需要生成器的 token 概率，并且需要依次进行两次大语言模型调用；在我们的设置中，每个查询约需 4.3 秒（附录 G.9）。**(4) 评估范围**。我们评估了 SQLite 基准测试和开放权重模型；Spider 2.0（Lei

等，2025）及其他方言尚未经过测试。只有对齐步骤和结构特征依赖 SQL；带有单元测试的代码有执行类比，但我们尚未评估其他任务（附录 H）。

人工智能使用声明

我们使用大语言模型（LLM）修正语法并辅助写作。大语言模型未参与任何研究讨论、分析或创意构思。所有见解、贡献和知识内容均完全由作者独立完成。本文评估的语言模型作为生成器和判别器，是研究对象，而非写作工具。

伦理声明

本研究使用了不含个人数据的公开基准（Spider、BIRD、NL2SQL-BUGs）和开放权重模型。本研究旨在减少错误却未被察觉的数据库答案传递给用户的情况。若部署时将已接受查询视为已验证，剩余错误仍可能造成影响。已接受查询中的错误率仅能以概率 $1-\delta$ 为界，且这一结论依赖于独立同分布采样（第3.1节）。

可复现性声明

第3节说明了形式化设定和假设。第5.1节介绍了数据、数据划分和实验协议。附录 A.1 和 A.2 给出了完整提示词，附录 D 给出了森林超参数，附录 F 给出了指标定义，附录G.1 给出了模型检查点。论文录用后，我们将公开用于特征提取、训练、校准和评估的代码。

参考文献

Anastasios N Angelopoulos 和 Stephen Bates。《共形预测：浅显入门》。机器学习基础与趋势，16(4):494–591，2023。本文引用的章节、命题和定理编号均指 arXiv:2107.07511v6。

Anastasios N. Angelopoulos、Stephen Bates、Emmanuel J. Cand`es、Michael I. Jordan 和 Lihua Lei。《先学习后检验：校准预测算法以实现风险控制》。*The Annals ofApplied Statistics*，19(2):1641–1662，2025。

Steph en Bates、Emmanuel Cand`es、Lihua Lei、Yaniv Romano 和 Matteo Sesia。检测离群共形的值方法《统计年鉴》，51(1)：149–178，2023。

Jipeng Cen、Jiaxin Liu、Zhixu Li 和 Jingjing Wang。Sqlfixagent：通过一致性增强的多智能体协作，实现语义准确的文本到 SQL 解析。载于《AAA I 人工智能大会论文集》，第 39 卷，第 49–57 页，2025。

Kaiwen Chen、Yueting Chen、Nick Koudas 和 Xiaohui Yu。借助自适应弃答实现可靠的文本到 SQL。《ACM 数据管理学报》，3(1)：1–30，2025年a。

Peter Baile Chen、Fabian Wenz、Yi Zhang、Devin Yang、Justin Choi、Nesime Tatbul、Michael Ca-farella、C.a˘gatay Demiralp 和 Michael Stonebraker。BEAVER：面向文本到 SQL 的企业级基准测试。载于 *ACL* 表格表示学习研讨会，2025b。

Shrey Desai 和 Greg Durrett。预训练变换器的校准。在*2020* 年自然语言处理经验方法会议（*EMNLP*）论文集，第 295–302 页，2020 年。

Sepideh Entezari Maleki、Mohammadreza Pourreza 和 Davood Rafiei。大语言模型中文本到 SQL 的置信度估计。在 *AAAI* 人工智能大会论文集，第 40 卷，第 32474–32482 页，2026 年。arXiv:2508.14056。

Sebastian Farquhar、Jannik Kossen、Lorenz Kuhn 和 Yarin Gal。使用语义熵检测大语言模型中的幻觉。《自然》，630(8017)：625–630，2024。

Yonatan Geifman 和 Ran El-Yaniv。深度神经网络的选择性分类。神经信息处理系统进展，30，2017。Yue Gong、Chuan Lei、Xiao Qin、Kapil Vaidya、Balakrishnan Narayanaswamy 和 Tim Kraska。SQLens：用于文本到 SQL 错误检测与纠正的端到端框架。载于神经信息处理系统进展，第 38 卷，2025。Dan Hendrycks 和 Kevin Gimpel。用于检测神经网络中误分类样本和分布外样本的基线方法。载于国际学习表征会议，2017。Saurav Kadavath、Tom Conerly、Amanda Askell、Tom Henighan、Dawn Drain、Ethan Perez、Nicholas Schiefer、Zac Hatfield-Dodds、Nova DasSarma、Eli Tran-Johnson 等。语言模型（大多）知道自己知道什么。*arXiv* 预印本 *arXiv:2207.05221*，2022。Lorenz Kuhn、Yarin Gal 和 Sebastian Farquhar。语义不确定性：自然语言生成中用于不确定性估计的语言不变性。载于国际学习表征会议，2023。Gyubok Lee、Woosog Chay、Seonhee Cho 和 Edward Choi。TrustSQL：采用基于惩罚的评分对文本到 SQL 可靠性进行基准测试。*arXiv* 预印本 *arXiv:2403.15879*，2024。Fangyu Lei、Jixuan Chen、Yuxiao Ye、Ruisheng Cao、Dongchan Shin、Hongjin Su、Zhaoqing Suo、Hongcheng Gao、Wenjing Hu、Pengcheng Yin 等。Spider 2.0：评估语言模型在真实企业文本到 SQL 工作流中的表现。载于国际学习表征会议，2025。Jinyang Li、Binyuan Hui、Ge Qu、Jiaxi Yang、Binhua Li、Bowen Li、Bailin Wang、Bowen Qin、Ruiying Geng、Nan Huo 等。大语言模型已经能否充当数据库接口？面向大规模、以数据库为基础的文本到 SQL 任务的大型基准测试。神经信息处理系统进展，36：42330–42357，2023。Terrance Liu、Shuyi Wang、Daniel Preoţiuc-Pietro、Yash Chandarana 和 Chirag Gupta。利用子句频率校准用于文本转 SQL 解析的大语言模型。载于*2025 年自然语言处理实证方法会议论文集*，第 16958–16982 页，2025年a。Xinyu Liu、Shuyu Shen、Boyan Li、Nan Tang 和 Yuyu Luo。NL2SQL-BUGs：用于检测自然语言到 SQL 转换中语义错误的基准测试。载于第 *31* 届 *ACM SIGKDD* 知识发现与数据挖掘会议论文集，第 *2* 卷，第 5662–5673 页，2025b。Andrey Malinin 和 Mark Gales。自回归结构化预测中的不确定性估计。载于国际学习表征会议，2021。Victor Quach、Adam Fisch、Tal Schuster、Adam Yala、Jae Ho Sohn、Tommi S Jaakkola 和 Regina Barzilay。共形语言建模。载于国际学习表征会议，2024。Ashwin Ramachandran 和 Sunita Sarawagi。文本到SQL校准：无需询问——只需重新缩放模型概率。*arXiv* 预印本 *arXiv:2411.16742*，2024。Oleg Somov 和 Elena Tutubalina。文本到 SQL 系统中用于错误检测的置信度估计。载于*AAAI* 人工智能大会论文集，第 39 卷，第 25137–25145 页，2025。Elias Stengel-Eskin 和 Benjamin Van Durme。校准解释：语义解析中的置信度估计。计算语言学协会汇刊，11：1213–1231，2023。Jiayuan Su、Jing Luo、Hongwei Wang 和 Lu Cheng。API 足矣：无需对数几率访问的大语言模型共形预测。载于计算语言学协会论文集：*EMNLP 2024* 论文集，第 979–995 页，2024。Katherine Tian、Eric Mitchell、Allan Zhou、Archit Sharma、Rafael Rafailov、Huaxiu Yao、Chelsea Finn 和 Christopher D. Manning。直接要求校准：从经过人类反馈微调的语言模型中获取校准置信度分数的策略。载于*2023 年自然语言处理实证方法会议论文集*，第 5433–5442 页，2023。

Vladimir Vovk、Alex Gammerman 和 Glenn Shafer。《随机世界中的算法学习》。施普林格，2005。

Xuezhi Wang、Jason Wei、Dale Schuurmans、Quoc V. Le、Ed H. Chi、Sharan Narang、Aakanksha Chowdhery 和 Denny Zhou。自一致性提升了语言模型中的思维链推理。载于 *ICLR 2023*，2023 年。网址：https://arxiv.org/abs/2203.11171。

Bingbing Wen、Jihan Yao、Shangbin Feng、Chenjun Xu、Yulia Tsvetkov、Bill Howe 和 Lucy Lu Wang。了解你的局限：大语言模型弃答综述。《计算语言学协会汇刊》，13：529–556，2025 年。

Quan Xiao、Debarun Bhattacharjya、Balaji Ganesan、Radu Marinescu、Katsiaryna Mirylenka、Nhan H Pham、Michael Glass 和 Junkyu Lee。大语言模型不确定性量化中的一致性假设。载于 人工智能不确定性会议（*UAI*），2025 年。

Miao Xiong、Zhiyuan Hu、Xinyang Lu、Yifei Li、Jie Fu、Junxian He 和 Bryan Hooi。大语言模型能否表达自身的不确定性？大语言模型置信度引导的经验评估。发表于 国际学习表征会议，2024。

Tao Yu、Rui Zhang、Michihiro Yasunaga 等。《Spider：用于复杂跨领域语义解析和文本到 SQL 任务的大规模人工标注数据集》。载于*EMNLP* 论文集，2018 年。

附录

A 提示词
14A.1 第一阶段提示（生成器）.14A.2 第二阶段提示（评判器）.17**B 完整功能列表**
20C

不同生成器骨干网络的特征分布**2**
0C.1 GPT-OSS-120B .21

C.2 Olmo-3-32B-Think.22C.3 Llama-3.3-Nemotron-Super-49B-v1.5 .23C.4
Qwen2.5-72B-Instruct .24**D 随机森林训练25E 证明25F 评估指标25**

G 其他实验结果26

G.1 模型详情 .26G.2 四个生成器骨干网络的完整结果 .27G.3 工作点 .27G.4 部分评判器配置 .29G.5 种子变异性 .29G.6 单个大语言模型派生分数的可靠性 .30G.7 误差分析 .30G.8 向 BIRD 的零样本迁移 .30G.9 延迟分解.31G.10 现有方法的已发表结果 .31

H 文本到 SQL 之外的适用性 **32**

A 提示词

A.1 阶段 1 提示（生成器）

生成器 θ_G 仅接收问题和包含采样行的模式。它会在内部验证其查询，但只输出查询及其置信度 c_{gen} （置信度），不提供验证判定。为便于显示，长行会自动换行。

你是一位文本到 SQL 专家，也是一名严格的 SQL 正确性审查员。

你的任务是生成一个与 SQLite 兼容的 SQL 查询，以回答给定的

自然语言问题，且只能使用所提供的数据库模式和 nd
示例。

你必须在内部同时执行 SQL 生成和严格的正确性验证，但

your 最终输出必须只包含 SQL 查询及其置信度分 re.

---------- 输入 ----------

自然语言问题:<QUESTION>

数据库模式及示例:<TABLE SCHEMA>_

SQL 生成与内部验证

生成一个兼容 SQLite 的 SQL 查询，以正确回答问题。

在评估置信度之前，请在内部验证以下所有内容:

1. 问题对齐 - SQL 必须直接回答问题的意图。- 确保所选的每一列、聚合、筛选条件、分组、排序、连接和 LIMIT 都有问题依据。- 不要回答对问题的不同解读或更狭窄的解读。- 不要引入不必要的连接、筛选条件、聚合或条件。

2. 上下文 –仅生成
- 仅使用所提供的模式和示例中明确包含的表、列、关系、值和概念。- 不要臆造表、列、关系、分类值、标识符、日期、实体或业务概念。- 不要假设所提供的上下文不支持的信息。- 不要编造字面筛选值。- 如果问题或模式/示例不支持某个必需值，则不要捏造。

3. 值与字面量验证 - WHERE、HAVING、JOIN 子句、CASE 表达式、IN 条件或其他条件中使用的每个字面量值，都必须得到问题或所提供的模式/示例的支持。- 保留上下文所支持值的确切含义和表示形式。

- 除非所提供的上下文支持，否则不要翻译、规范化或重新解释分类值。- 如果问题中包含某个值，除非模式明确要求使用有依据的表示形式，否则应忠实使用该值。- 切勿仅仅因为某个值看起来合理就擅自引入它。

4. SQL 正确性
- 查询必须符合有效的 SQLite语法。- 只能使用 SQLite 支持的函数、运算符、表达式、日期处理、聚合、连接、子查询、公用表表达式和窗口函数。- 不要使用 PostgreSQL、MySQL、SQL Server、Oracle 或其他 SQL 方言中特有、且在 SQLite 中无效的语法。- 确保表别名、列引用、GROUP BY、HAVING、ORDER BY、连接条件、子查询和公用表表达式在语法和逻辑上均有效。- 确保所引用的表和列确实存在于给定模式中

.

- SQL 必须完整且可执行。

5. 逻辑正确性
- 检查 SQL 是否计算出问题所要求的结果。- 检查连接是否符合正确的关系，并避免意外产生笛卡尔积。- 检查聚合层级和分组。- 检查筛选条件和比较运算符。- 在适用时检查排序和 LIMIT 语义。- 在相关情况下检查 NULL 处理。- 检查查询是否意外地重复或筛除记录。

6. 结果合理性
- 查询应能够生成问题所要求类型的结果 e

问题。- 不要仅仅为了让查询返回特定结果而添加限制性条件。- 不要因为查询在语法上有效，就想当然地认为它是正确的。- 如果问题和模式确实允许结果为空，则潜在的空结果并不一定是错误的。

7. 最小化且忠实的 SQL
- 优先使用能够正确回答问题的最简单 SQL。
- 不要添加不必要的复杂性。
- 问题含义明确时，不要提出其他解释。

- 不要使用所提供上下文以外的信息。

置信度校准----------

请给出 0 到 100 之间的置信度分数，表示你对生成的 SQL 同时满足以下两点的确信程度：

1. 与问题正确对应；2. 完全基于所提供的上下文。

置信度必须反映完整内部验证的结果，而不仅仅是 SQL 语法有效的置信度。

请遵循以下规则：

- 90-100：问题明确无歧义，所有必需的模式元素和值均有明确支持，SQL 与问题高度一致，且对SQLite语法和逻辑具有极高的确定性。

- 70-89：SQL 很可能正确，但在解释或模式方面存在轻微不确定性。

- 40-69：对于意图、值、关系或 SQL 逻辑存在较为明显的不确定性。

- 1-39：存在重大不确定性，缺少必需信息，或 SQL 可能包含未获支持的假设。

- 0：无法可靠地根据所提供的上下文确定 SQL 的依据，或 SQL 无法正确回答问题。

重要：
- 语法有效的 SQL 查询并不能证明其具有高置信度。- 如果 SQL 包含臆造的表、列、关系、值或未经支持的假设，置信度必须较低。- 如果 SQL 与问题明显不符，置信度必须较低。

- 如果 SQLite 兼容性不确定，置信度必须较低。- 置信度必须反映生成的 SQL 中最薄弱的关键方面，而不是对各项独立检查结果取平均值。

~~最终结果~~-------

仅返回一个有效的 JSON 对象。

JSON 必须仅包含以下两个字段：

{ "sql_query": "<生成的兼容 SQLite 的 SQL 查询>", "confidence": <介于 0 和 100>之间的整数}

请勿输出：
- 解释 - 推理 - 验证结果 - 评估分数 -
其他字段 - Markdown - 代码围栏 -
JSON 以外的注释

输出必须是有效的 JSON，且不能包含其他内容。

A.2 第二阶段提示词（评判器）

评判器 $\theta_C \neq \theta_G$ 在一次调用中接收问题、模式、生成查询和执行结果。步骤 1 得出 v_{sql}（评估分数）和 c_{val}（评估置信度）。步骤 2 得出 $\tilde{q}$（生成的问题），步骤 3 得出 a（对齐分数）和 s_{sim}（问题相似度）。

You 你是一位严格、审慎、以结果为导向的文本到 SQL 评估器。

你的任务是判断已执行SQL结果是否正确且完整地回答了原始问题。不要假定可以访问黄金SQL或黄金答案。

输入：
1. 自然语言问题 2. 数据库模式 3. 生成的 SQL 查询 4. SQL执行结果

Per在内部执行以下步骤，并且仅返回指定的 JSO N.

步骤 1：严格结果验证

判断该结果是否完整且正确地回答了问题。根据问题评估结果，而不仅仅是判断这些值是否看起来合理。

检查所有相关方面：

1. 列／行有效性 - 是否包含所需且相关的列？ - 是否缺少必需的行？ - 是否存在多余或无效的行／列？

2. 运算／聚合对齐
确认结果反映了问题明确要求的运算：计数、求和、平均值、最小值、最大值、百分比、比率、速率、差值、总计、最多、最少、最高、最低等。

例如，要求平均值却返回求和结果，或要求计数却返回单独各行，都属于不正确。

3. 筛选条件／运算符对齐
核实问题中的每个条件和运算符：>、 >=、 <、 <=、 =、!=、介于、之前、之后、至少、至多、恰好、类别筛选条件、日期、实体、与／或、排除项等。结果必须符合确切的语义和边界。

4. 排序／限制／前 K 项对齐 如果问题指定了前几项／后几项、最早／最新／最旧、排序、排列或结果数量，请核实以下两点：- 排序／排名依据正确 - 返回记录的数量正确 例如，“按收入列出前 5 项”要求返回收入最高的五条记录，而不是随便五条记录。

5. 分组／粒度
验证结果是否具有所要求的聚合层级：例如，总体总计与按公司统计的总计、按年份计算的平均值、各行业的结果等。

6. 日期／时间对齐
核实所要求的年份、日期、时间段、之前/之后的语义、范围，

最晚/最早要求及其他时间约束。

7. 值/结果一致性
验证返回的标量、聚合、排序和派生值是否
c与问题一致。

8. 完整性/空结果
验证问题的所有部分是否都已得到回答。空值、NULL 或零结果并非一定错误；应判断它们是否与问题及所应用的条件一致。

如果违反任何实质性要求，结果就是错误的。

评估分数：_- `1` = 结果正确且完整。- `0` = 结果不正确、不完整，或违反任何实质性要求。

评估置信度：_
对上述二元评估的置信度（0-100）。
- 90-100：证据明确，没有实质性歧义 - 70-89：证据充分，存在轻微不确定性 - 40-69：存在实质性不确定性 - 1-39：存在较大不确定性 - 0：无法可靠判断正确性

明确违反某项显式要求（聚合、运算符、筛选条件、排序、LIMIT、分组、日期等），应对评估分数 `= 0`._ 给出较高置信度

~~步骤 2：反向问题生成~~

忽略原始问题。

仅使用 SQL 结果和模式，生成一个结果实际回答的最精确问题。

不要修正或理想化，也不要推断缺失的筛选条件、聚合、条件、排序或其他要求。

~~步骤 3：对齐与相似度~~

将生成的问题与原始问题进行比较。

对齐_分数（0-100）衡量结果本身对原始问题的回答程度，考量以下因素：

- 请求的信息 - 筛选条件和运算符 - 聚合 - 排序 - LIMIT/前 k 项 - 分组/粒度 - 日期/时间 - 完整性

问题_相似度（0-100）衡量原始问题与生成的问题之间的语义相似度。

重要提示：- 结果可能与主题相关，但在操作层面并不正确。- 不要仅仅因为返回的数据相关，就给出较高的对齐度。

- 如果评估_分数为 = 0，则在存在实质性正确性错误时，对齐_分数通常应为 <= 30。- 如果结果只部分回答了问题，则对齐_分数通常应为 <= 50。- 如果评估_分数为 = 0，则问题_相似度不得超过 60。

输出

准确返回一个有效的 JSON 对象，包含以下五个字段：

{ "evaluation_score": <0 或 1>, "evaluation_confidence": <介于 0 和 100>之间的整数, "alignment_score": <介于 0 和 100>之间的数字, "generated_question": "<生成的问题>", "question_similarity": <介于 0 和 100>之间的数字}

不要使用 Markdown。不要解释。不要推理。不要添加其他字段。

---------- 输入 ----------

原始问题：<INSERT_QUESTION_HERE>

数据库模式：<INSERT DB SCHEMA HERE>_ _ _

生成的 SQL 查询：<INSERT_SQL_QUERY_HERE>

SQL 结果：<在此插入记录>
_ _

B 完整特征列表

表7列出了**x**的全部41个特征，以及我们在代码中使用的标识符。所有词元统计均使用生成器针对s的词元给出的概率 π_t，范围从第一个 SELECT 词元到结束分号。关键字 词元是大写的 SQL 关键字，而非关键字 词元则是其余部分（标识符、字面量、运算符）。类别列给出了消融实验（表5）中使用的分组。

表7：全部 41 项特征。核心特征见正文（表1）及附录 C 中的密度图。

类别	符号	代码名称	定义	Core
语言化	c_{gen}	生成置信度	生成器的言语化置信度，0–100（阶段 1）	✓
	v_{sql}	评估分数	评判器对结果的二元判定（阶段2）	✓
	c_{val}	评估置信度	评判器对其判定的置信度，0–100（阶段2）	✓
对齐	a	对齐分数	结果对 q的回答程度，0–100（阶段2）	✓
	s_{sim}	问题相似度	q 与重构后的 $\hat{q}$的相似度，0–100（阶段2）	✓
对数几率 (P_{feats})	μ_{geom}	所有令牌的几何平均值	所有词元的 π_t几何平均值	✓
	FTC	所有令牌的 FTC	所有词元的 π_t乘积	✓
	μ_{arith}	均值 SQL 全部	所有词元的算术平均值 π_t	✓
	$\pi_{\min}$	最小值 SQL 全部	所有词元的最小值 π_t	✓
	H	熵 SQL 全部	$-\sum_t \pi_t \log \pi_t$，所有词元	✓
	σ^2	SQL 词元方差	π_t的方差	✓
	ℓ_{CLM}	clm	$-\sum_t$ 日志 π_t (= −日志 FTC)	✓
	*	SQL子句	geo、FTC、均值、最小值、关键词上的熵 词元（5 个特征）	
	*	非 SQL	对非关键字词元采用相同的五个特征（5 个特征）	
	min		最小值 π_t 去除空白词元后（近似 $\pi_{\min}$的重复项）	
		SQL 词元边际	概率范围 最大值$_t$ π_t − 最小值$_t$ π_t	
		直方图 分箱 1–分箱 13	over 13 概率 分箱，$[0, 0.3)$, $[0.3, 0.4)$, $[0.4, 0.5)$, 然后使用宽度为 0.05 的分箱，直到 1。	
			每个值是最小–最大归一化 分箱中词元概率的均值（为空时为 0）（13 个特征）	
SQL 结构 (S_{feats})	SAC$_{\text{geom}}$	sac	对子句组内均值取几何平均值 （组内均值 π_t ）（公式 11）	✓
	SLC	slc	模式元素词元的 π_t 几何平均值 （公式 11）	✓
		模式几何平均值	与模式匹配的词元的 π_t 几何平均值 表名	
		c3s	最弱子句几何平均值 $\times\, e^{-\text{Var}_{\text{inter}}} \times$ $e^{-\text{Var}_{\text{intra}}}$ 在各子句组上的表现	
总计：5 个大语言模型派生 + 32 对数几率 + 4 结构性				**41**（14 个核心）

某些特征在某些配置中是恒定的，因此森林永远不会使用它们。例如，生成器 GPT-OSS-120B 的分箱 1（概率低于 0.3）始终为空。

C 生成器骨干网络间的特征分布

图 2–5 展示了每种生成器骨干网络的 16 个特征的核密度估计：表 7 中的 14 个核心特征，以及在非关键字词元上计算的 FTC 和最小概率（ftc non sql、min non sql）。之所以纳入这两个非关键字变体，是因为标识符和字面量正是出现模式对齐错误的地方。为便于阅读，我们省略了其余 25 个特征。这些特征包括：在词元子集上重新计算的核心统计量、直方图分箱、近重复最小值或范围 SQL 词元间隔，以及结构变体模式几何平均值和 C3S。每个面板中，正确查询（$Z{=}1$）以橙色显示，错误查询（$Z{=}0$）以蓝色显示。密度经过联合归一化，因此每条曲线的面积与其类别大小成正比。脚本 plot feature kde.py 可根据保存的特征表重现这些图。

C.1 GPT-OSS-120B

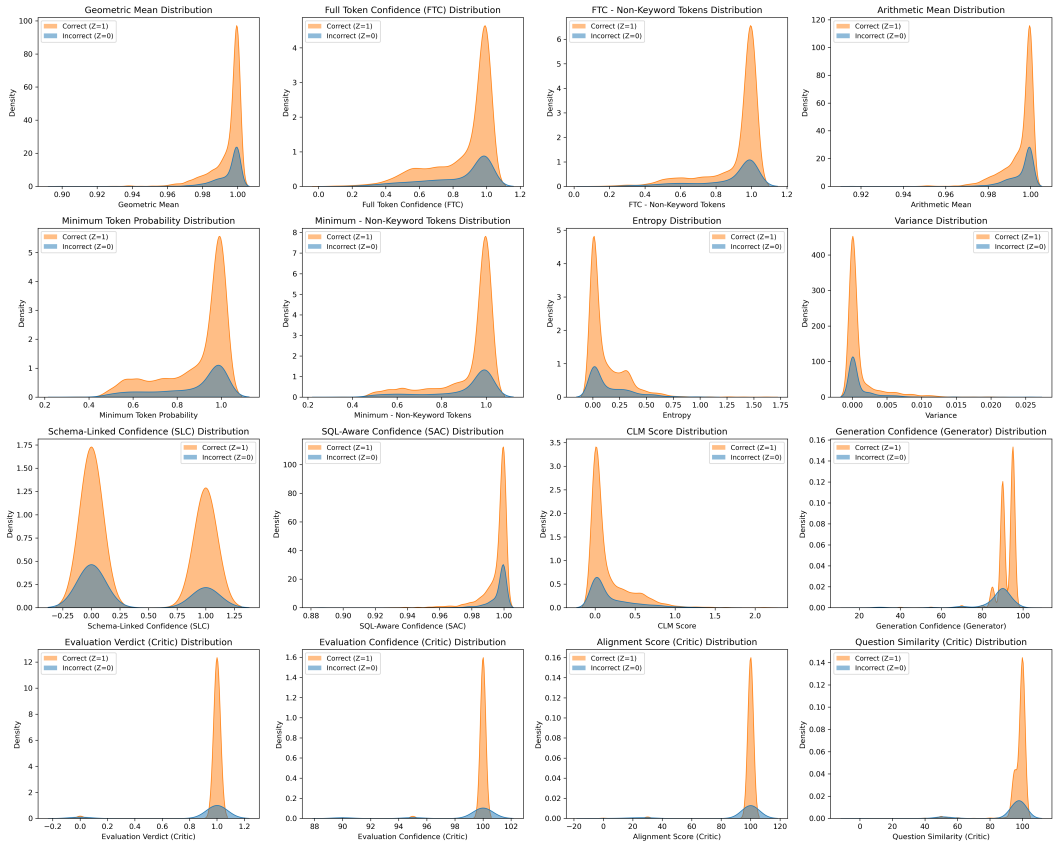

图 2：生成器 GPT-OSS-120B 在评判器 Gemma-4-31B 下的 14 个核心特征和两个非关键词变体的核密度估计，按执行结果区分（橙色：正确，蓝色：错误）。特征选择的说明见附录 C 开头。

C.2 OLMO-3-32B-THINK

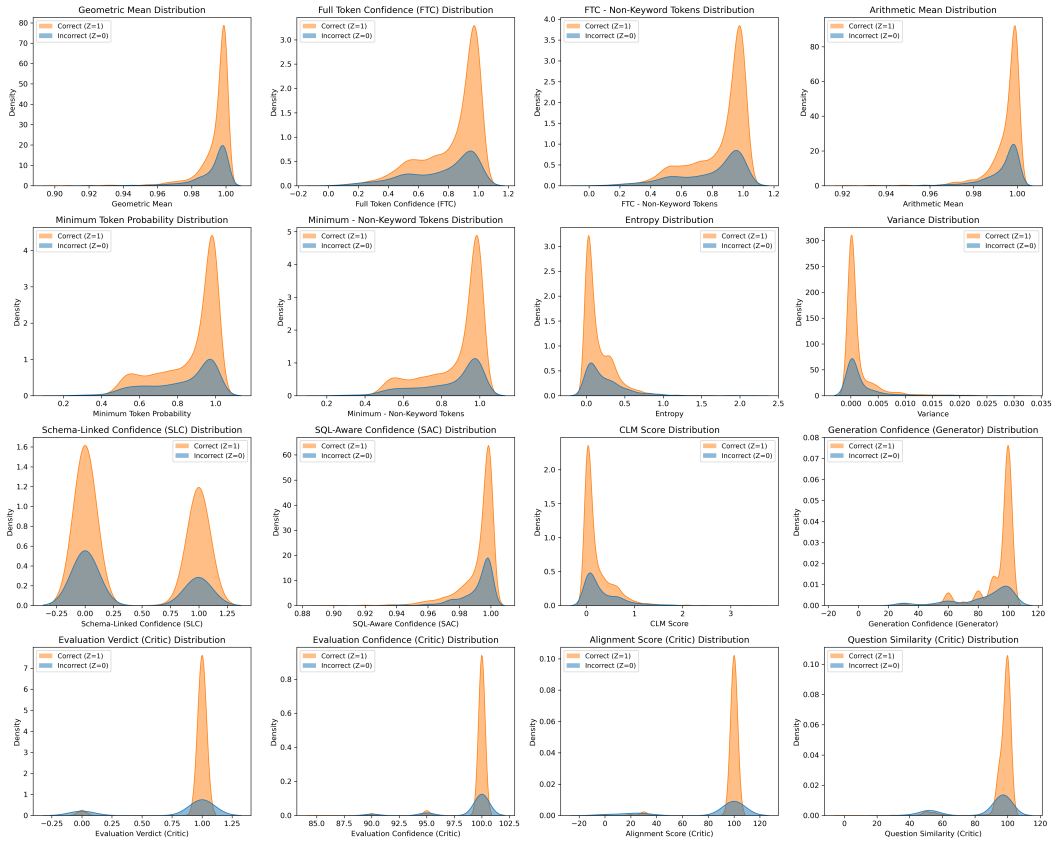

图 3：生成器 Olmo-3-32B-Think 搭配 Gemma-4-31B 评论器时，14 个核心特征和两个非关键词变体的核密度估计，并按执行结果区分（橙色：正确，蓝色：错误）。特征选择的说明见附录 C 开头。

C.3 Llama-3.3-Nemotron-Super-49B-v1.5

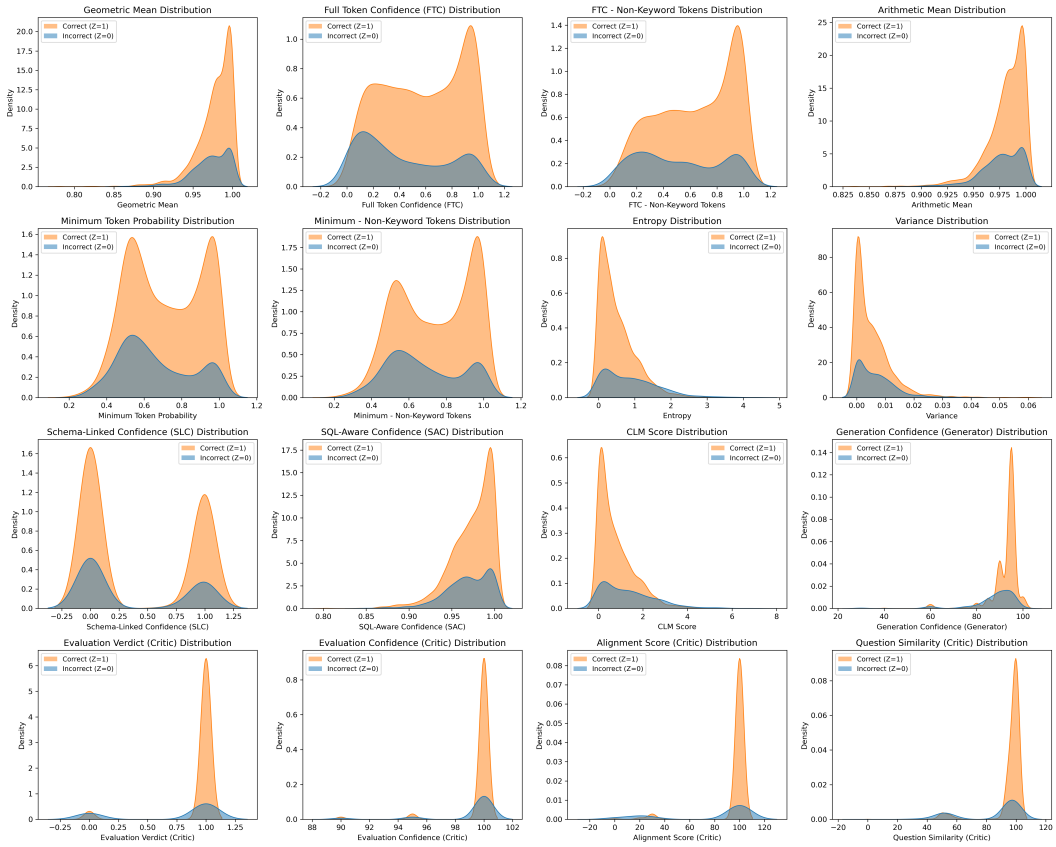

图 4：生成器 Llama-3.3-Nemotron-Super-49B-v1.5 搭配 Gemma-4-31B 评判器时，14 个核心特征和两个非关键词变体的核密度估计，按执行结果区分（橙色：正确，蓝色：错误）。特征选择详见附录 C 开头。

C.4 QWEN2.5-72B-INSTRUCT

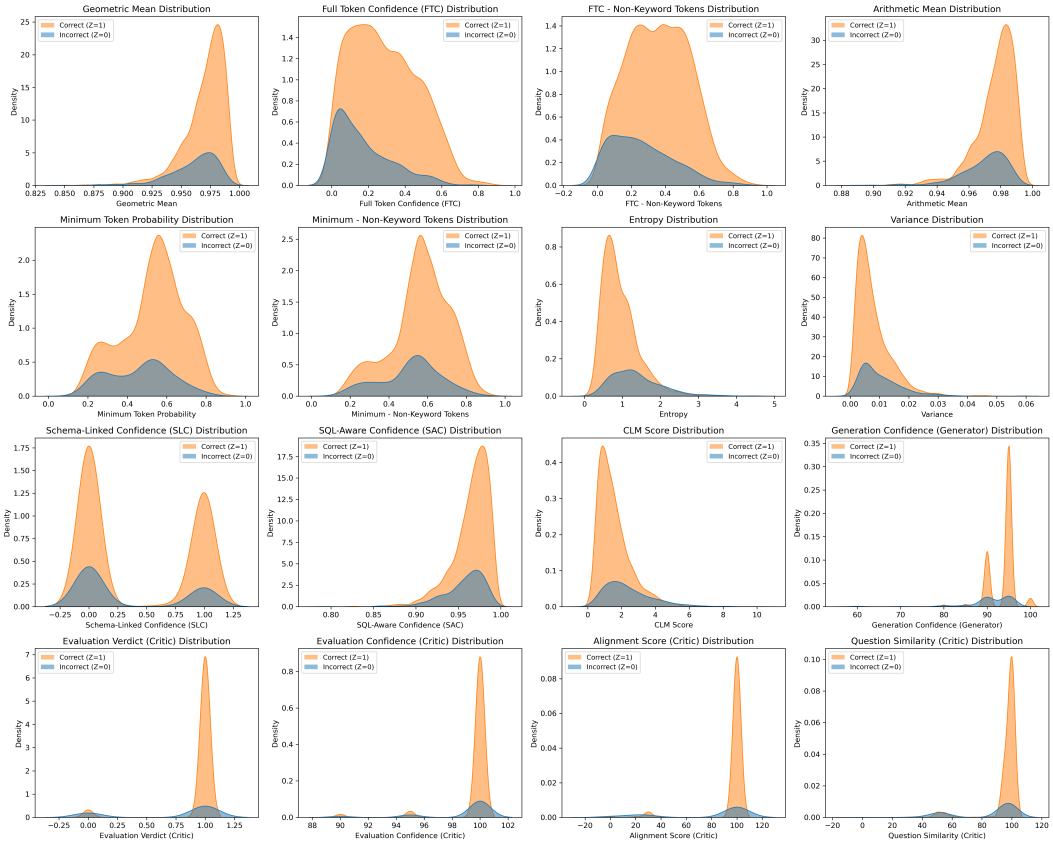

图 5：生成器 Qwen2.5-72B-Instruct 搭配 Gemma-4-31B 评判器时，14 个核心特征和两个非关键词变体的核密度估计，并按执行结果区分（橙色：正确，蓝色：错误）。特征选择详见附录 C 开头。

D 随机森林训练

超参数。 对于每种配置，在训练划分上使用 5 折交叉验证，对 40 种配置进行随机搜索，以最大化 ROC曲线下面积，从而选出超参数。始终采用自助采样和类别权重=平衡子采样，并启用袋外估计。表 8 列出了选定的值。

表 8：每种配置选定的随机森林超参数。 *四骨干模型研究（第5.1节）的生成运行。

配置	树的数量	最大深度	最小分裂样本数	最小叶节点样本数	最大特征数
生成器 GPT-OSS-120B，评判器 Gemma-4-31B	400	10	5	10	sqrt
生成器 GPT-OSS-120B，评判器 Qwen3.8-27B	400	10	5	10	sqrt
生成器 GPT-OSS-120B，评判器 DeepSeek-V4.1-Flash	400	10	5	10	sqrt
生成器 GPT-OSS-120B，评判器 Nemotron-3-Super	200	50	10	10	all
生成器 GPT-OSS-120B* ，评判器 Gemma-4-31B	800	10	10	10	all
生成器 Olmo-3-32B，评判器 Gemma-4-31B	400	15	5	10	log2
生成器 Qwen2.5-72B，评判器 Gemma-4-31B	200	20	2	1	log2
生成器 Llama-49B，评判器 Gemma-4-31B	400	10	5	10	sqrt

按配置训练。 针对每种配置分别训练一个森林。其训练集包含在 Spider 训练划分上，根据该配置的输出计算得到的特征向量，标签 Z_i 则通过与黄金SQL进行执行匹配获得。我们针对每个骨干模型分别训练，因为基于 logit 的特征（$P_{\text{feats}}, S_{\text{feats}}$）在不同骨干模型上的分布各不相同。不同模型家族和规模的词元概率范围、熵尺度和 SAC 值也各不相同（图 2–5）。对于零样本迁移（第 5.6 节），将 GPT-OSS-120B* 森林及校准分数（表 8）应用于 GPT-OSS-120B 在 BIRD 上的输出。对于 SQL 已给定的 NL2SQL-BUGs，则使用基于四个评判器分数训练的森林。整个过程中均未使用目标域标签。

保形校准。 使用$Z_i = 1$对校准查询计算校准分数 $\eta_i = -\log\ p_i$。测试时，计算公式6时将并列情况计入分子（$\eta_i \geq \eta$）。这样计入并列情况会使$\hat{c}$变大，并使假接受率界更为保守。使用 $1 - p$ 而非 $-\log\ p$ 作为非一致性分数，得到的$\hat{c}$相同，因为二者互为单调变换。

E 证明

命题 1。 令 $\eta_1, \ldots\ldots, \eta_{m_1}$ 为正确校准查询的得分，η_{m_1+1} 为正确测试查询的得分。在可交换性条件下，给定训练估计器 p，η_{m_1+1} 在这 $m_1 + 1$ 个得分中的排名在 $\{1, \ldots, m_1 + 1\}$上服从均匀分布。并列值计入公式6的分子，这只会增加$\hat{c}$。因此，$\hat{c} = (1 + |\{i \leq m_1 :\ \eta_i \geq \eta_{m_1+1}\}|)/(m_1 + 1)$ 满足$\mathbb{P}(\hat{c} \leq t \mid Z = 1) \leq \lfloor t(m_1 + 1) \rfloor/(m_1 + 1) \leq t$。取 $t = 1 - \alpha$ 可得 FAR$(1 - \alpha) = \mathbb{P}(\hat{c} < 1 - \alpha \mid Z = 1) \leq 1 - \alpha$ (Angelopoulos 与 Bates，2023，命题 3)。

命题 2. 固定 $\lambda \in \Lambda$。由于 p 是在与 $\mathcal{C}$ 不重叠的数据上训练的，且校准查询是独立同分布的，因此，给定其数量 $n(\lambda)$后，满足 $p_i \geq \lambda$ 的查询是从满足 $p \geq \lambda$的查询分布中独立同分布抽取的。因此，其错误计数满足 $k(\lambda) \mid n(\lambda) \sim \text{Bin}(n(\lambda), \text{风险}(\lambda))$。在 H_λ : 风险$(\lambda) > \varepsilon$的条件下，根据二项随机序，对于所有 s，均有 $\mathbb{P}(u_\lambda \leq s) \leq s$，因此 u_λ是有效的 p 值。该规则仅在 $u_\lambda < \delta/|\Lambda|$时认证阈值。根据并集界，认证任何满足风险$(\lambda) > \varepsilon$ 的 λ 的概率至多为 δ。在其补事件上，$\hat{\lambda}$满足风险$(\lambda) \leq \varepsilon$；否则，没有阈值获得认证，系统会对所有查询弃答。这就是带有 Bonferroni 校正的先学习后检验保证（Angelopoulos et al.，2025；Angelopoulos 与 Bates，2023，定理 A.1）。

F 评估指标

所有指标均在测试集划分上计算。对于选择性指标，如果查询的得分 $\hat{c}$ 满足规则的阈值（$\hat{c} \geq 1 - \alpha$ 适用于保形规则，$p \geq \lambda$ 适用于风险控制规则），则该查询被接受，并且

否则弃答。我们将接受视为正向预测，并记作 TP（正确且被接受）、FP（错误且被接受）、FN（正确但弃答）和 TN（错误但弃答）。

表 9：评估指标。根据命题 2（风险控制规则），风险受到界定；根据命题 1（保形规则），假接受率受到界定。

指标	定义	更好
ECE	在 10 个等宽分箱。	↓
ACE	与期望校准误差相同，使用等质量分箱。	↓
布里尔分数	置信度与 Z之间的均方误差。	↓
AUROC	随机选取的正确查询获得更高分数的概率，高于随机选取的错误查询。	↑
AUARC	准确率-拒绝率曲线下的面积：保留查询的准确率随着拒绝率从 0 变化到 1。主要指标。	↑
精确率/召回率/F1（表11）	森林在 $p \geq 0.5$ 对 $Z{=}1$ 的分类。	↑
错误检测精确率/召回率/F1 (BIRD，NL2SQL-BUGs)	通过标记具有 $\hat{c} < 1-\alpha$ 的查询，对 $Z{=}0$ 进行分类。	↑
覆盖率	已接受查询的比例，(TP + FP)/n。经验值。	—
Risk	已接受查询中的错误率，FP/(TP+FP)。由 ε界定，其中在独立同分布数据下，风险控制规则的概率为 $1-\delta$ （命题 2）。题 2）。	↓
条件准确率	已接受查询的准确率，$1-$风险 $=$TP/(TP + FP)。标识与 SSC($\lvert S\rvert{=}1$) 相同。经验值。	↑
TAR	真实弃答率，TN/(TN +FP)：错误查询中被暂缓处理的比例。经验值。	↑
FAR	假弃权率，FN/(FN +TP)：正确查询中被暂缓处理的比例—暂缓处理。在可交换性条件下，期望不超过 $1-\alpha$。	↓
SSC($\lvert S\rvert{=}0$)	弃权查询中的准确率，假负例/（假负例 + 真负例）。	—

G 补充实验结果

G.1 模型详情

表10：实验中使用的语言模型。前四种用作生成器。Gemma-4-31B-it 是默认评判器，Qwen3.8-27B、DeepSeek-V4.1-Flash 和 Nemotron-3-Super 是其他判别器。Qwen2.5-72B-Instruct 也用作评判器。主配置将生成器 GPT-OSS-120B 与评判器 Gemma-4-31B-it 配对。

模型	参数量（B）	HuggingFace模型ID
GPT-OSS-120B	120	openai/gpt-oss-120b
Olmo-3-32B-Think	32	allenai/Olmo-3-32B-Think
Llama-3.3-Nemotron-Super-49B-v1.5	49	nvidia/Llama-3_3-Nemotron-Super-49B-v1_5
Qwen2.5-72B-Instruct	72	Qwen/Qwen2.5-72B-Instruct
Gemma-4-31B-it (评判器)	31	google/gemma-4-31B-it
Qwen3.8-27B（评判器）	27	Qwen/Qwen3.8-27B
DeepSeek-V4.1-Flash（评判器）	—	deepseek-ai/DeepSeek-V4.1-Flash
Nemotron-3-Super（评判器）	120	nvidia/NVIDIA-Nemotron-3-Super-120B-A12B-BF16

G.2 四种生成器骨干网络的完整结果

表11报告了四种生成器骨干网络的结果，每种均搭配 Gemma-4-31B 评论器。α=0.1 处的混淆矩阵重现了报告的真接受率、假接受率和条件准确率。假接受率和条件准确率采用不同的分母。对于 Olmo-3-32B，假接受率以 470 个正确查询为分母（其中 416 个被拒绝，比例为 0.885）；条件准确率则以 56 个已接受查询为分母（其中 54 个正确，比例为 0.964）。

表11：在 Spider 测试集划分上，四种生成器骨干网络搭配 Gemma-4-31B 评论器的结果。选择性指标采用 α=0.1处的保形规则。GPT-OSS-120B 列使用的是四骨干研究中的生成运行，而不是表 2 中的主运行。

指标	**GPT-OSS-120B**	**Olmo-3-32B**	**Qwen2.5-72B**	**Llama-49B**
正确／错误查询	499 / 137	470 / 166	503 / 133	475 / 160
基础准确率 ρ	0.785	0.739	0.791	0.748
ECE	0.133	0.126	0.045	0.113
ACE	0.133	0.125	0.041	0.112
布里尔分数	0.163	0.158	0.137	0.153
AUROC	0.791	0.816	0.765	0.814
AUARC	0.918	0.895	0.898	0.902
精确率 (Z=1, $p \geq 0.5$)	0.889	0.847	0.815	0.854
召回率 (Z=1, $p \geq 0.5$)	0.804	0.851	0.974	0.848
F1分数 (Z=1, $p \geq 0.5$)	0.844	0.849	0.888	0.851
真正例 / 假正例 / 假负例 / 真负例	51 / 1 / 448 / 136	54 / 2 / 416 / 164	50 / 3 / 453 / 130	52 / 1 / 423 / 159
覆盖率	0.082	0.088	0.083	0.083
Risk	0.019	0.036	0.057	0.019
条件准确率 = 1−风险	0.981	0.964	0.943	0.981
TAR	0.993	0.988	0.977	0.994
FAR	0.898	0.885	0.901	0.891
SSC ($\|S\|$=0)	0.767	0.717	0.777	0.727

G.3 工作点

所报告划分上的风险控制规则。表12将风险控制规则应用于表2和表11所依据的划分。在此划分上，每个认证阈值都能将测试风险控制在 ε 以下。评判器各行使用生成器 GPT-OSS-120B；生成器各行使用 Gemma 判别器以及四骨干研究的生成运行。

ˆ 表12：在报告的 Spider 划分上应用风险控制规则（δ=0.1）。 λ：在 p 上的认证阈值（“—”：无认证阈值，因此系统对每个查询都弃答）。覆盖率和风险以百分比表示。

	$\varepsilon = 0.10$			$\varepsilon = 0.15$			$\varepsilon = 0.20$		
	$\hat{\lambda}$	覆盖	风险	$\hat{\lambda}$	覆盖	风险	$\hat{\lambda}$	覆盖	风险
评判器：Gemma-4-31B	—	0.0	—	0.58	54.4	8.4	0.44	76.4	13.4
评判器：Qwen3.8-27B	0.68	44.3	5.7	0.50	67.5	9.6	0.40	79.7	12.4
评判器：DeepSeek-V4.1-Flash	0.77	29.4	2.7	0.59	54.7	8.9	0.41	81.0	14.4
评判器：Nemotron-3-Super	—	0.0	—	0.65	47.3	7.3	0.46	71.1	13.1
生成器：GPT-OSS-120B	0.91	20.1	0.8	0.67	47.6	5.9	0.56	61.6	8.7
生成器：Olmo-3-32B	—	0.0	—	0.77	26.4	3.6	0.51	73.0	14.7
生成器：Qwen2.5-72B	0.87	42.3	8.6	0.74	72.5	10.8	0.60	89.6	16.5
生成器：Llama-49B	—	0.0	—	0.67	50.9	9.3	0.57	65.8	11.7

保形规则。 表 13 扫描保形规则的运行水平 α。根据式 5， α 将正确查询的接受比例固定在约 α，而估计器决定同时接受多少错误查询（1 − 真接受率）。使用 Gemma 判别器时，覆盖率从 7.9% 上升至 100%。风险在 α=0.6 之前保持在 5.0% 至 6.8% 之间，之后上升至无弃权误差 22.6%。在这次单次划分中，假接受率最多超出其界 0.049 （在 α=0.5 处）。所有水平共用一次校准抽样，因此其偏差彼此相关。该界在对抽样取期望时成立。

α=0.1 处的假接受率 10 个随机种子的均值为 0.907 ± 0.016，在该界的一个标准差范围内（附录 G.5）。

表 13：保形规则（命题 1）在 Spider 测试集划分上的覆盖率–风险权衡。评判器对应的行使用 GPT-OSS-120B 作为生成器。最后几行对表11中的四种生成器骨干网络取平均（Gemma 判别器）。

	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
	假接受率界 $1-\alpha$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0
评判器: Gemma-4-31B	FAR	.904	.813	.728	.640	.549	.445	.319	.181	.075	.000
	覆盖率（%）	7.9	15.3	22.2	29.6	37.3	46.1	58.2	72.3	85.7	100
	风险 (%)	6.0	5.2	5.0	5.9	6.3	6.8	9.5	12.4	16.5	22.6
评判器: Qwen3.8-27B	覆盖率 (%)	6.3	14.2	21.2	26.9	35.4	46.7	58.3	69.2	80.0	100
	风险 (%)	0.0	3.3	2.2	2.3	4.0	5.7	6.7	9.3	12.6	22.6
四个生成器（平均值）	覆盖率 (%)	8.4	15.8	23.4	31.7	39.8	51.2	61.9	72.9	84.2	100
	风险（%）	3.3	3.3	4.1	4.7	6.3	8.8	10.5	13.1	16.9	23.4

表14给出了表13中评判器行背后的混淆矩阵，这些矩阵根据每种配置的测试集预测结果计算得出（492个正确查询和144个错误查询）。

表14：α 中接受/弃权混淆矩阵（生成器 GPT-OSS-120B，Spider 测试集划分，n=636）。正类 = 表示接受。

	评判器：Gemma-4-31B						评判器：Qwen3.8-27B					
α	真正例	假正例	假负例	真负例	真接受率	假接受率	真正例	假正例	假负例	真负例	真接受率	假接受率
0.1	47	3	445	141	.979	.904	40	0	452	144	1.000	.919
0.2	92	5	400	139	.965	.813	87	3	405	141	.979	.823
0.3	134	7	358	137	.951	.728	132	3	360	141	.979	.732
0.4	177	11	315	133	.924	.640	167	4	325	140	.972	.661
0.5	222	15	270	129	.896	.549	216	9	276	135	.938	.561
0.6	273	20	219	124	.861	.445	280	17	212	127	.882	.431
0.7	335	35	157	109	.757	.319	346	25	146	119	.826	.297
0.8	403	57	89	87	.604	.181	399	41	93	103	.715	.189
0.9	455	90	37	54	.375	.075	445	64	47	80	.556	.096
1.0	492	144	0	0	.000	.000	492	144	0	0	.000	.000

生成器 Olmo-3-32B。 表 15 给出了生成器 Olmo-3-32B 搭配 Gemma 判别器时各运行水平下的混淆矩阵。正确查询召回率、$1-$假接受率，保持在接近 α 的水平，正如命题 1 所示；随着 α 增长，风险从 3.6% 升至 26.1%。图 6 绘制了主配置对应的曲线。

表 15：使用 Gemma 判别器的生成器 Olmo-3-32B-Think 在 α 上的接受/弃答混淆矩阵（Spider 测试集划分，n=636）。正类 =为接受。

α	TP	FP	FN	TN	条件准确率	召回率 = 1–假接受率	Risk
0.1	54	2	416	164	96.4%	11.5%	3.6%
0.2	90	4	380	162	95.7%	19.1%	4.3%
0.3	136	6	334	160	95.8%	28.9%	4.2%
0.4	185	9	285	157	95.4%	39.4%	4.6%
0.5	239	18	231	148	93.0%	50.9%	7.0%
0.6	293	27	177	139	91.6%	62.3%	8.4%
0.7	347	43	123	123	89.0%	73.8%	11.0%
0.8	395	68	75	98	85.3%	84.0%	14.7%
0.9	435	99	35	67	81.5%	92.6%	18.5%
1.0	470	166	0	0	73.9%	100.0%	26.1%

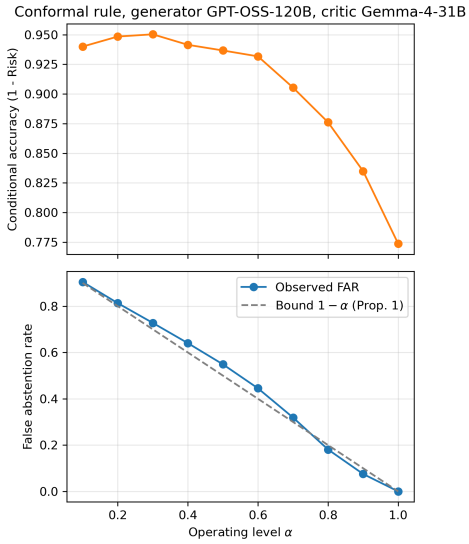

图 6：保形规则在不同运行水平 α 下的条件准确率（上）和假接受率（下）（生成器 GPT-OSS-120B，评判器 Gemma-4-31B）。虚线是假接受率界 $1-\alpha$，见命题 1。

G.4 部分评判器配置

在部分 配置中，生成器提供自身的验证判定和置信度，另一个模型仅提供对齐分数和相似度分数。表 16 将这些结果与自我评判以及完整的 Qwen2.5-72B 评判器进行了比较。四种配置下的 AUARC 均介于 0.905 和 0.915 之间。

表16：生成器–批评器分离程度（生成器 GPT-OSS-120B，Spider，α=0.1）

	自我批评	Gemma部分配置	Qwen部分配置	Qwen批评器
AUROC	0.781	0.782	0.821	0.826
AUARC	0.913	0.905	0.913	0.915
ECE	0.149	0.129	0.154	0.166
TAR	0.993	1.000	0.993	0.993
FAR	0.904	0.896	0.902	0.896

G.5 种子变异性

For 对于每种配置，我们使用随机种子 0–9 对分层训练/校准/测试划分进行重采样。我们重新拟合 t the 森林的超参数固定为表 8 中的设置，并重新计算校准分数。

表 17：10 个重采样数据划分的均值 ± 标准差（α=0.1）。评判器行使用生成器 GPT-OSS-120B；生成器行使用 Gemma 判别器和四骨干研究的生成运行。生成器 Qwen2.5-72B 的期望校准误差较低，这反映了其所选的森林（表 8）。

配置	AUROC	AUARC	ECE	TAR	FAR
评判器：Gemma-4-31B	0.772 ± 0.020	0.900 ± 0.010	0.151 ± 0.007	0.988 ± 0.012	0.907 ± 0.016
评判器：Qwen3.8-27B	0.811 ± 0.019	0.914 ± 0.008	0.136 ± 0.009	0.993 ± 0.006	0.902 ± 0.012
评判器：DeepSeek-V4.1-Flash	0.793 ± 0.021	0.907 ± 0.009	0.145 ± 0.009	0.988 ± 0.006	0.905 ± 0.016
评判器：Nemotron-3-Super	0.766 ± 0.023	0.898 ± 0.011	0.125 ± 0.011	0.984 ± 0.011	0.905 ± 0.018
生成器：GPT-OSS-120B	0.772 ± 0.023	0.909 ± 0.010	0.133 ± 0.011	0.992 ± 0.007	0.899 ± 0.015
生成器：Olmo-3-32B	0.790 ± 0.016	0.885 ± 0.008	0.128 ± 0.006	0.990 ± 0.006	0.889 ± 0.020
生成器：Qwen2.5-72B	0.785 ± 0.015	0.911 ± 0.007	0.032 ± 0.009	0.983 ± 0.012	0.895 ± 0.016
生成器：Llama-49B	0.806 ± 0.012	0.898 ± 0.007	0.131 ± 0.007	0.990 ± 0.008	0.896 ± 0.021

G.6 单个大语言模型派生分数的可靠性

表 18 报告了融合前每个大语言模型派生分数的单独 AUROC。前四列对应使用 Gemma 判别器的生成器骨干网络（四骨干研究的生成运行，因此 GPT-OSS 的 c_{gen} 与最后四列不同）。最后四列使用生成器 GPT-OSS-120B，并标明判别器。每种配置中，c_{gen} 来自生成器，其余四个分数来自判别器。全部 40 个值均高于 0.5（范围为 0.525–0.771）。

表18：大语言模型派生分数在 Spider 测试集划分上的独立 AUROC。

	生成器（评判器为 Gemma-4-31B）				评判器（生成器为 GPT-OSS-120B）			
分数	GPT-OSS	Olmo	Qwen2.5	Llama	Gemma	Qwen3.8	DeepSeek	Nemotron
c_{gen} (生成置信度)	0.739	0.697	0.658	0.704	0.715	0.715	0.715	0.715
v_{sql} (评估判定)	0.537	0.607	0.576	0.605	0.563	0.567	0.588	0.596
c_{val} (评估置信度)	0.567	0.569	0.554	0.525	0.580	0.771	0.700	0.572
a (对齐)	0.537	0.608	0.579	0.606	0.564	0.753	0.651	0.611
s_{sim} (问题相似度)	0.652	0.721	0.665	0.704	0.651	0.737	0.710	0.595

G.7 误差分析

特征分布（附录 C）揭示了各个信号在哪些情况下会失效。正确查询与错误查询的 logit 特征高度重叠，因为错误查询通常在逐个词元层面仍然流畅。错误出在语义上，而非解码时的犹豫。对于大多数正确查询，评判器的判定和对齐分数都达到其“看起来正确”的峰值，而大多数错误查询也处于这一峰值。

我们考察了主配置中的 144 个错误测试查询，其中 GPT-OSS-120B 负责生成，Gemma-4-31B 负责评判。若评判器的判定 v_{sql} 和对齐分数 a 均标记出某个错误（$v_{\text{sql}} = 0$ 和 $a < 50$；20 个查询），则称该错误已识别；若两者均未标记（124 个查询），则称该错误未识别。因此，评判器误判了大多数错误。已识别错误的森林后验均值为 $p = 0.34$，且在运行水平不高于 α=0.5的任何情况下，最多只会有一个错误被接受。未识别错误的后验均值更高（$p = 0.47$），但在 α=0.1时，124 个错误中仅有 3 个被接受，因为生成器侧特征仍能将它们与大多数正确查询区分开来。随着 α 升至 0.5，最多有 14 个（11%）错误被接受。因此，当 α较大时，被评判器误判的错误是已接受错误的主要来源。

G.8 向 BIRD 的零样本迁移

表19：向 BIRD 开发集进行零样本迁移（1,534 个查询，生成器 GPT-OSS-120B，评判器 Gemma-4-31B，采用在 Spider 上训练的森林和校准，未进行重新训练或重新校准）。错误检测（$\hat{c} < 0.9$）的精确率、召回率和 F1分数。对于此基础准确率，AUARC 的范围为 0.343（随机）至 0.709（完美排序）。

指标	**值**
基础准确率	0.342
ECE	0.091
ACE	0.090
布里尔分数	0.202
AUROC	0.724
AUARC	0.515
错误检测精确率	0.662
错误检测召回率	0.998
错误检测F1分数	0.796
条件准确率（α=0.1）	0.833
Spider 认证的风险 $\hat{\lambda}$ (ε=0.15)	0.301

为什么 BIRD 上的期望校准误差低于 Spider。 较低的期望校准误差并不意味着域外校准效果更好。这是基率效应（表 20）。在 Spider 上，基础准确率较高，因此即使较低的-

置信度预测通常是正确的，并且在每个分箱中，经验准确率都高于置信度。在 BIRD 上，大多数预测的置信度较低，而在该区域内，准确率接近置信度均值，因此对 ECE 的贡献很小。保形校准无法解释这两种结果。它关注的是阈值，而不是 ECE 所衡量的分箱一致性；此外，其保证也不适用于 BIRD，因为 BIRD 的查询与 Spider 校准集并不具备可交换性。

表 20：ECE 差异背后的基准率比较（生成器 GPT-OSS-120B，评判器 Gemma-4-31B）。

	Spider（域内）	BIRD（零样本）
不弃答时的错误率	21.5%	65.8%
含有 $p \leq 0.5$ 的预测占比	29%	72%
此处的准确率与平均置信度	53% 对比 36%	24% 对比 35%
ECE	0.133	0.091

G.9 延迟分解

表21：主配置的每次查询实际耗时（秒）。生成器、评判器和本地阶段在 40 个测试查询上逐个查询计时。第3阶段可与阶段2并行运行。

阶段	**操作**	**平均时间（秒）**
1	SQL 生成（GPT-OSS-120B） +执行	1.9 - 2.6
2	验证 + 对齐（Gemma-4-31B 评论器）	0.5 - 1.7
3–4	特征、森林、保形分数（本地）	0.03
总计	每次查询	**2.43 - 4.33**

由于查询执行，生成器占据了主要成本。两次大语言模型调用均通过启用了推理功能的共享推理网关运行。特征提取、森林训练与推理，以及保形校准均在没有 GPU 的 MacBook M4 Pro 上运行。

G.10 现有方法已发表的结果 DS

表 22 列出了原始论文中报告的 Spider 结果。这些结果采用了不同的骨干模型、数据划分和评估协议，因此不能直接与我们的方法进行比较。四项研究中有三项报告了 AUROC，但没有报告 AUARC，因此 AUROC 是最接近的共同指标。表 4 给出了受控比较。

表 22：现有置信度估计方法在 Spider 上报告的结果，各方法均采用各自的骨干模型和实验协议。

方法	**骨干模型**	**方法**	**报告的结果**
SAC-Product（Entezari Maleki 等，2026）	GPT-4o-mini	基于Logit	AUROC 79.9，期望校准误差 0.14
几何聚合（Xiao 等人，2025）	CodeLlama-34B	黑盒	AUROC 76.0，AUARC 35.0
多变量Platt缩放（Liu等人，2025年a）	Llama-3.1-8B	对数几率 + 采样	AUROC 74.8，布里尔分数 0.163，期望校准误差 0.024，自适应校准误差 0.026
汇聚式令牌级（Ramachandran & Sarawagi, 2024）	Codestral-22B	基于Logit	AUROC 78.8
Ours	Olmo-3-32B-Think + Gemma-4-31B 评论器	组合	AUARC 89.5，AUROC 81.6，ECE 0.126，自适应校准误差 0.125，布里尔分数 0.158

H 文本到 SQL 之外的适用性

流水线的某些部分并不依赖 SQL，包括生成器–批评器分离、批评器的验证步骤、令牌级特征、森林融合和保形校准。它们适用于任何能够生成带有词元概率的输出，并且拥有可据以计算正确性的标注数据的任务。阶段2的对齐步骤和结构特征则特定于 SQL。对齐步骤需要一个具体结果、一种生成该结果所回答问题的方法，以及对该结果的检查。代码生成（以单元测试作为执行类比）和数学推理（借助求解器）都有这样的类比。开放域问答则没有；阶段2需要采用替代方法，例如进行归因检查，以确定检索到的证据是否支持答案。我们尚未评估这些扩展。