# WHEN VERBOSE RESPONSES MISLEAD EVALUATORS: ANALYTIC DEBIASING FOR GENERATIVE KGQA

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Knowledge Graph Question Answering (KGQA) aims to answer natural-language questions over a knowledge graph. Classical KGQA systems typically return a single canonical answer entity or an executable logical form, so exact-match evaluation is usually well aligned with their outputs. Recent systems, however, increasingly generate free-form natural-language responses. This makes string matching less reliable: it can reject correct answers with mismatched surface forms and accept incorrect responses that merely mention answer-like entities. We formalize this issue as evaluator bias in generative KGQA. In our framework, observed scores reflect not only latent semantic correctness, but also false-negative mismatch and false-positive contamination. We then propose the Analytic Debiasing Evaluator (ADE), a simple, interpretable, and judge-free framework. ADE starts from matches accepted by a string-based evaluator and assigns each accepted match a reliability score based on its risk of lexical contamination. ADE estimates lexical-contamination risk using matched negative controls and discounts accepted matches that are likely to be contamination-prone. Experiments on human-annotated meta-evaluation sets show that ADE aligns more closely with human judgments than string-matching evaluators. ADE also achieves competitive agreement with LLM-as-Judge evaluators while being deterministic, faster, and free of API cost. In leaderboard analyses, ADE produces contamination-aware rankings that better match human semantic judgments, suggesting that it can help diagnose when biased evaluators distort model rankings.

## 1 INTRODUCTION

Knowledge Graph Question Answering (KGQA) studies how to answer natural-language questions over a structured knowledge graph (Yih et al., 2014; Berant & Liang, 2014). In classical KGQA, evaluation was relatively straightforward because systems usually returned standardized outputs, such as canonical answer entities or executable logical forms (Yih et al., 2015; Liang et al., 2017; Chen et al., 2019b). Exact-match scoring therefore served as a practical proxy for correctness. Recent KGQA systems, however, have increasingly shifted toward an LLM-based generative paradigm (Achiam et al., 2023; Liu et al., 2024), producing free-form responses that may include explanations, reasoning traces, and redundant entity mentions (Sun et al., 2024; Wan et al., 2025; Wang et al., 2026; Jiang et al., 2025). This shift makes string matching less reliable, since a correct response may use a different surface form, while an incorrect one may merely mention the gold answer.

To accommodate free-form outputs, recent generative KGQA papers use nominally similar metrics such as Exact Match, Accuracy, and Hits@1. However, their implementations differ in alias alignment, normalization, and matching rules (Chen et al., 2024b; Lin et al., 2025; Luo et al., 2025; Zhang et al., 2025). This inconsistency is consequential because fully reproducing KGQA systems can be expensive due to inference, retrieval, prompting, and API costs. Consequently, many comparisons rely on scores reported in prior papers rather than re-evaluating all systems under a shared script. If these scores are computed by different evaluators, cross-paper comparisons become unreliable and model rankings may reflect evaluator choices rather than true semantic performance. We summarize this fragmented evaluator landscape in Appendix B.

Beyond comparability, these string-based evaluators share a deeper limitation. They are vulnerable to false-negative mismatch, where a semantically correct response is rejected because its surface



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

form is absent from the gold answer string or alias inventory, and false-positive contamination, where an incorrect response is accepted simply because it mentions the gold answer or one of its aliases. Alias-aware evaluation can reduce false negatives (Xiong et al., 2024), but it also expands the lexical surface over which accidental matches may occur. This trade-off is particularly acute for verbose generative outputs, which often contain multiple answer-like entity mentions. Under permissive matching rules, a response may be counted as correct simply because it mentions the gold answer, even if it does not actually answer the question. This can inflate the measured performance of systems that produce more contamination-prone responses (Saito et al., 2023).

In this work, we formulate this issue as evaluator bias in generative KGQA. Our key observation is that automatic evaluator scores are biased proxies for semantic correctness, because they reflect both true correctness and evaluator errors. Surface-form mismatch can cause correct responses to be rejected, while lexical contamination can cause incorrect responses to be accepted. These errors can bias benchmark scores and even reverse model rankings, making a more contamination-prone model appear stronger than a semantically better one. Since true semantic capability is not identifiable from evaluator outputs alone, we treat generative KGQA evaluation as an estimation problem rather than merely a choice of metric name or matching script.

To make this estimation perspective operational, we propose the Analytic Debiasing Evaluator (ADE), a simple, interpretable, and judge-free framework for contamination-aware generative KGQA evaluation. ADE first applies an alias-aware string matcher and then estimates whether each accepted match could be explained by accidental lexical contamination. It does so using matched negative controls: answer-like but incorrect aliases chosen to resemble the gold answer in lexical properties that affect accidental matching. If the same response also matches these controls, its gold-answer match is treated as less reliable and is discounted accordingly. This design targets contamination-prone positives without simply penalizing verbose natural-language responses, and unlike LLM-as-Judge approaches (Liu et al., 2023; Zheng et al., 2023), it requires no additional judge model at evaluation time.

We evaluate ADE through a multi-faceted empirical study of generative KGQA evaluators. On human-annotated meta-evaluation sets, ADE aligns more closely with human semantic judgments than string-matching evaluators in our evaluated settings. It also achieves competitive agreement with LLM-as-Judge evaluators while being deterministic, faster, and free of API cost. In leaderboard analyses, ADE produces contamination-aware rankings that better match human semantic judgments, helping diagnose when biased evaluators distort model rankings.

Our contributions are threefold. (1) We formalize evaluator bias in generative KGQA by decomposing observed scores into semantic correctness, false-negative mismatch, and false-positive contamination. (2) We propose ADE, a judge-free contamination-aware evaluator that uses matched negative controls to estimate lexical-contamination risk and conservatively discount accepted string matches. (3) We show that evaluator implementation choices can alter leaderboard conclusions, and that ADE provides a conservative reliability signal better aligned with human judgments than string matching.

## 2 Related Work

### 2.1 KGQA Methods

Classical knowledge graph question answering (KGQA) has mainly been studied under two paradigms: semantic parsing, which maps a question to an executable query over the knowledge graph, and answer entity retrieval, which directly selects the most likely answer entity using embedding-based or graph-based models (Yih et al., 2015; Shu et al., 2022; Ye et al., 2022; Luo et al., 2024a). Because classical KGQA systems usually output standardized answer entities or executable logical forms, exact-match evaluation is typically sufficient. Recent KGQA research, however, has increasingly shifted toward LLM-based retrieval-augmented methods, differing in graph retrieval strategies but uniformly generating free-form text (Luo et al., 2024b; Ma et al., 2025; Xu et al., 2024; Liu et al., 2025; Liang & Gu, 2025). In these methods, retrieved triples or reasoning paths are provided to the LLM together with the question, and the model generates the final answer in natural language. This shift to free-form generation makes the evaluation problem substantially more challenging.

### 2.2 Semantic Evaluation and Judge Bias in Generative KGQA

Prior work in question answering has shown that lexical-overlap metrics such as Exact Match and token-level F1 are imperfect proxies for semantic correctness (Chen et al., 2019a; Bulian et al., 2022). This issue becomes more pronounced in open-domain and LLM-based QA, where generated answers are often longer and may be semantically equivalent to gold answers without sharing their surface forms (Kamalloo et al., 2023). To address such false-negative errors, prior work has explored learned or semantic answer-equivalence metrics, including BERT-based answer matching and semantic answer similarity models (Bulian et al., 2022; Risch et al., 2021), as well as general neural generation metrics such as BERTScore and BLEURT (Zhang* et al., 2020; Sellam et al., 2020). However, these approaches primarily aim to recover semantically correct answers missed by lexical matching. They do not explicitly model the complementary false-positive problem in generative KGQA: an incorrect response may incidentally mention a gold entity and receive credit under permissive substring or alias-based matching.

Recent generative KGQA work still largely relies on string-based variants of Exact Match, Accuracy, or Hits@1, differing mainly in alias alignment, normalization, and matching strictness. As we summarize in Appendix B, this evaluation landscape remains fragmented and fundamentally lexical. Another alternative is to use LLM-as-Judge evaluators (Liu et al., 2023; Zheng et al., 2023; Li et al., 2025) or judge models trained for fine-grained evaluation (Wang et al., 2024; Kim et al., 2024; Zhu et al., 2025). While these methods can capture semantic equivalence beyond surface overlap, recent meta-evaluation studies show that LLM judges can be sensitive to prompt design, model versioning, reference availability, position and format effects, and other judgment biases (Zeng et al., 2024; Chen et al., 2024a; Shi et al., 2025; Han et al., 2026). They also introduce additional cost, latency, and reproducibility concerns at evaluation time. In contrast, our method is a task-specific, judge-free correction framework that targets a different failure mode. It explicitly estimates and discounts false-positive lexical contamination in permissive generative KGQA evaluation.

## 3 Theoretical Framework for String-based Evaluator Bias

In this section, we formalize why string-based evaluation can be biased in generative KGQA. We focus on a fixed model and a fixed evaluator, and view the model response as a random variable induced by the model's output distribution. The key difficulty is that modern LLM-based KGQA systems often produce free-form natural-language answers. As a result, a string-based evaluator may reject a correct response whose surface form differs from the gold answer, or accept an incorrect response that merely mentions the gold answer. We refer to these two failure modes as false-negative mismatch and false-positive lexical contamination, respectively. Under these errors, the observed evaluator score is not a direct measurement of semantic correctness, but a biased proxy for it.

### 3.1 Variables and Problem Formulation

For the $i$-th query $q_i$, let $Y_i \in \{0, 1\}$ denote whether the model response is semantically correct, and let

$$l_i := \Pr(Y_i = 1 \mid q_i), \qquad L := \frac{1}{n} \sum_{i=1}^{n} l_i \tag{1}$$

denote the instance-level semantic correctness propensity and the benchmark-level latent capability, respectively.

Let $x_i \in \{0, 1\}$ be the observed binary score assigned by the string-based evaluator. We decompose evaluator errors into false negatives and false positives. The false-negative rate is

$$\alpha_i := \Pr(x_i = 0 \mid Y_i = 1, q_i), \tag{2}$$

namely the probability that a semantically correct response is judged incorrect. The false-positive rate is

$$\beta_i := \Pr(x_i = 1 \mid Y_i = 0, q_i), \tag{3}$$

namely the probability that a semantically incorrect response is accepted by the evaluator. For string-based evaluators in generative KGQA, such false positives often arise from lexical contamination. In

this case, an incorrect response is accepted because it contains the gold answer string or one of its aliases, even though the response does not actually identify it as the answer.

In classical KGQA, $\beta_i$ is often small because outputs are typically normalized logical forms or canonical answer entities, and $\alpha_i$ is easier to control when predictions and gold answers share the same canonical representation. In generative KGQA, model outputs are free-form text rather than canonical entities. Evaluators therefore often use aliases and normalization rules to avoid missing correct surface variants. However, these rules also increase the chance that an incorrect response accidentally matches a gold alias. This creates a trade-off: reducing false-negative mismatch can increase false-positive lexical contamination, making $\beta_i$ a central source of evaluator bias.

### 3.2 Bias Decomposition of String-based Scores

**Proposition 1** (Expectation decomposition)**.** *For any query $q_i$, the conditional expectation of the observed evaluator score is*

$$\mathbb{E}[x_i \mid q_i] = l_i(1-\alpha_i) + (1-l_i)\beta_i. \tag{4}$$

Thus, the observed evaluator score is a mixture of latent semantic correctness and evaluator-specific noise. Averaging over the benchmark,

$$\mathbb{E}[\bar{x}] = \frac{1}{n}\sum_{i=1}^{n}\left[l_i(1-\alpha_i) + (1-l_i)\beta_i\right], \qquad \bar{x} := \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{5}$$

The bias of the observed benchmark score relative to the latent capability $L$ is therefore

$$\mathbb{E}[\bar{x}] - L = \frac{1}{n}\sum_{i=1}^{n}\left[(1-l_i)\beta_i - l_i\alpha_i\right]. \tag{6}$$

The two terms correspond to score inflation from false-positive contamination and score deflation from false-negative mismatch, respectively.

### 3.3 Ranking Inversion Under False-Positive Contamination

False-positive contamination affects not only absolute scores but also relative model ordering. For a model $M$ with $L_M < 1$, define its effective benchmark-level false-positive contamination rate as

$$\bar{\beta}_M := \frac{\sum_{i=1}^{n}(1-l_{i,M})\beta_{i,M}}{\sum_{i=1}^{n}(1-l_{i,M})} = \frac{1}{n(1-L_M)}\sum_{i=1}^{n}(1-l_{i,M})\beta_{i,M}. \tag{7}$$

This is the average false-positive rate conditional on semantic incorrectness. For a clean analytic boundary, we consider the case where the false-negative rate is shared across models and instances.

**Corollary 3.1** (Ranking inversion boundary)**.** *Consider two non-perfect models $A$ and $B$ with latent capabilities $1 > L_A > L_B$. Assume they share the same constant false-negative rate $\alpha$, and let $\bar{\beta}_A$ and $\bar{\beta}_B$ denote their effective false-positive contamination rates. If*

$$\bar{\beta}_B - \bar{\beta}_A > \frac{(L_A - L_B)(1-\alpha-\bar{\beta}_A)}{1-L_B}, \tag{8}$$

*then their expected observed evaluator scores satisfy*

$$\mathbb{E}[\bar{x}_B] > \mathbb{E}[\bar{x}_A]. \tag{9}$$

Corollary 3.1 shows that a less capable but more verbose model may obtain a higher expected evaluator score simply because its outputs are more likely to collide with the gold answer string. On sufficiently large benchmarks, such expectation-level inversion can manifest as a practical leaderboard inversion.

### 3.4 UNIDENTIFIABILITY OF LATENT CAPABILITY

Finally, latent semantic capability cannot in general be recovered from observed evaluator scores alone.

**Corollary 3.2** (Unidentifiability)**.** *Without additional information or modeling assumptions on $\alpha_i$ and $\beta_i$, the latent capability $L$ is not identifiable from the observed scores $\{x_i\}_{i=1}^n$ alone. Indeed, even if the exact evaluator expectations $\{\mathbb{E}[x_i \mid q_i]\}_{i=1}^n$ were known, $L$ would still not be identifiable.*

Corollary 3.2 highlights that observed evaluator scores cannot distinguish genuine semantic correctness from spurious lexical agreement without explicitly modeling evaluator bias. This motivates the contamination-aware evaluator developed in the next section.

## 4 ANALYTIC DEBIASING EVALUATOR

Section 3 shows that string-based scores in generative KGQA can mix semantic correctness with false-negative mismatch and false-positive lexical contamination. We propose the *Analytic Debiasing Evaluator* (ADE), a judge-free evaluator that assigns a contamination-aware reliability score to string-based matches. ADE first uses an alias-aware matcher to recover answers expressed with different surface forms. It then discounts accepted matches that are likely to be accidental lexical collisions. For each instance, ADE takes as input the question $q_i$, generated response $r_i$, gold answer alias set $\mathcal{A}_i$, and outputs a soft reliability score $S_{\mathrm{ADE}}^{(i)} \in [0, 1]$.

The key idea is to estimate contamination through *matched negative controls* (Lipsitch et al., 2010; Shi et al., 2020). For each question, ADE samples plausible but incorrect answers that are matched to the gold answer by properties such as answer type, alias length, and frequency. These sampled answers serve as negative controls: if the same response also matches their aliases, then its match to the gold answer is less reliable evidence of semantic correctness. ADE does not recover semantic correctness from evaluator outputs alone. Section 3 shows that this is not identifiable without additional assumptions. Instead, ADE introduces matched negative-control aliases as extra evidence for lexical contamination. Under the exchangeability assumption stated below, these controls estimate how likely an accepted match is to be accidental.

### 4.1 ALIAS-AWARE BASE MATCHING

ADE first applies a high-recall base evaluator,

$$x_i^{\mathrm{AA}} = \mathrm{AliasMatch}(r_i, \mathcal{A}_i) \in \{0, 1\}, \tag{10}$$

which matches the response $r_i$ against the gold alias set $\mathcal{A}_i$ using answer aliases, string normalization, and answer-type canonicalization for entities, dates, numbers, and set-valued answers. This stage reduces false-negative mismatch caused by surface-form variation. ADE then applies a positive-only correction: instances with $x_i^{\mathrm{AA}} = 0$ remain zero, while instances with $x_i^{\mathrm{AA}} = 1$ are further assessed for accidental lexical collision. After alias-aware matching, ADE focuses on accepted matches, since false-positive contamination can only affect responses that the base matcher has already marked as correct. We analyze the effect of alias expansion in Appendix I.1.

### 4.2 CONTAMINATION ESTIMATION WITH MATCHED NEGATIVE CONTROLS

For each instance $i$, ADE constructs $B$ negative-control alias sets

$$\mathcal{N}_i = \{\mathcal{A}_{i,1}^-, \ldots, \mathcal{A}_{i,B}^-\}.$$

Each $\mathcal{A}_{i,b}^-$ corresponds to an entity or answer value that is not a correct answer to $q_i$. To make it a useful negative control, ADE selects it to match the gold alias set $\mathcal{A}_i$ on properties that affect accidental lexical matching. In our implementation, these matching properties include answer type, alias-set size, alias length, entity frequency and neighbor entity. The negative controls are sampled before model evaluation and are fixed across all evaluated models for the same question.

Given the response $r_i$, ADE applies the same alias-aware matcher to each negative-control alias set:

$$z_{i,b}^- = \mathrm{AliasMatch}(r_i, \mathcal{A}_{i,b}^-) \in \{0, 1\}, \qquad b = 1, \ldots, B. \tag{11}$$

The counterfactual contamination estimate is the empirical negative-control collision rate:

$$\hat{\beta}_i = \frac{1}{B} \sum_{b=1}^{B} z^-_{i,b}. \tag{12}$$

Intuitively, $\hat{\beta}_i$ measures how often the response would have received lexical credit against plausible but incorrect answer aliases. A high value indicates that the response is lexically promiscuous and may obtain spurious credit under permissive matching. A low value indicates that the gold-answer match is less likely to be explained by accidental collision.

This estimator differs from a raw length penalty. A long response is not discounted merely because it contains many tokens. It is discounted only when the response also triggers matches to the counterfactual answer aliases under the same evaluator. Thus, ADE targets answer-like lexical contamination rather than verbosity itself.

Let

$$\beta_i = \Pr(\text{AliasMatch}(r_i, \mathcal{A}_i) = 1 \mid Y_i = 0, q_i) \tag{13}$$

denote the probability that a semantically incorrect response still matches the gold alias set. This is the gold-answer lexical-collision rate that ADE aims to estimate with negative controls.

**Proposition 2** (Negative-control collision consistency)**.** *Assume that, conditional on $(Y_i = 0, q_i)$ and the matched answer-side covariates, the gold alias set $\mathcal{A}_i$ and the negative-control alias sets $\mathcal{A}^-_{i,1}, \ldots, \mathcal{A}^-_{i,B}$ are exchangeable with respect to lexical matching against $r_i$. If the negative controls are sampled independently from the matched control pool, then*

$$\mathbb{E}[\hat{\beta}_i \mid Y_i = 0, q_i] = \beta_i, \tag{14}$$

*and for any $t > 0$,*

$$\Pr\Big(|\hat{\beta}_i - \beta_i| \geq t \mid Y_i = 0, q_i\Big) \leq 2\exp(-2Bt^2). \tag{15}$$

Proposition 2 does not identify semantic correctness from evaluator outputs alone. Rather, it shows that the lexical-contamination component becomes estimable once matched negative controls are introduced. The proof is provided in Appendix A.4, and implementation details for negative-control construction are provided in Appendix C.

### 4.3 Contamination-aware Reliability Scoring

ADE converts the alias-aware binary decision into a soft reliability score (Guo et al., 2017). We view an evaluator-positive match as produced by one of two channels: a semantic channel where $Y_i = 1$, or a contamination channel where $Y_i = 0$ but the response still matches an answer alias. For an evaluator-positive instance, Bayes' rule gives

$$\Pr(Y_i = 1 \mid x_i^{\text{AA}} = 1, q_i) = \frac{\pi_i(1 - \alpha_i)}{\pi_i(1 - \alpha_i) + (1 - \pi_i)\beta_i}, \tag{16}$$

where $\pi_i = \Pr(Y_i = 1 \mid q_i)$ is the prior probability of semantic correctness, $\alpha_i$ is the residual false-negative rate, and $\beta_i$ is the gold-answer lexical-collision rate under incorrect responses.

However, the posterior in Eq. equation 16 is not identifiable from string-based evaluator outputs alone, since $\pi_i$ and $\alpha_i$ are unknown. ADE therefore does not attempt to recover this exact posterior. Instead, Eq. equation 16 motivates the principle that an evaluator-positive match should receive lower reliability when its estimated contamination risk is higher. Using the negative-control estimate $\hat{\beta}_i$, ADE constructs the following monotonic reliability mapping:

$$S^{(i)}_{\text{ADE}} = x_i^{\text{AA}} \cdot \frac{1}{1 + \tau \frac{\hat{\beta}_i}{1 - \hat{\beta}_i + \epsilon}}, \tag{17}$$

where $\epsilon > 0$ is a small numerical constant and $\tau \geq 0$ controls the discount strength. The parameter $\tau$ is fixed globally rather than tuned for individual datasets. The odds-based form assigns mild correction when the estimated contamination probability is close to zero and increasingly conservative correction as it approaches one.

Equation equation 17 preserves the positive-only design. If $x_i^{\mathrm{AA}} = 0$, then $S_{\mathrm{ADE}}^{(i)} = 0$. If $x_i^{\mathrm{AA}} = 1$, ADE assigns a reliability score that decreases monotonically with the estimated counterfactual contamination risk.

The benchmark-level ADE score is

$$\hat{L}_{\mathrm{ADE}} = \frac{1}{n} \sum_{i=1}^{n} S_{\mathrm{ADE}}^{(i)}. \tag{18}$$

For binary agreement analysis, we threshold the soft score:

$$\hat{y}_i^{\mathrm{ADE}} = \mathbb{I}\left[ S_{\mathrm{ADE}}^{(i)} \geq \delta \right], \tag{19}$$

where $\delta$ is selected on a calibration split or fixed to $0.5$ in the parameter-free setting. We report both soft benchmark scores and thresholded instance-level agreement to distinguish score calibration from binary evaluator decisions.

# 5 Experiments

In this section, we design experiments to evaluate the effectiveness, robustness, and efficiency of the Analytic Debiasing Evaluator (ADE). Specifically, our empirical study aims to answer the following four research questions:

- **RQ1 (Human Alignment):** Does ADE better align with human judgments of semantic correctness than existing string-based evaluator variants?
- **RQ2 (Leaderboard Sensitivity and Debiasing):** How sensitive are generative KGQA rankings to string-matching implementation choices, and what ranking does ADE produce?
- **RQ3 (Exchangeability Diagnostics):** Do matched negative controls provide calibrated estimates of accidental gold-alias collision risk, and how important is the control-matching strategy?
- **RQ4 (Efficiency):** How does ADE compare with LLM-as-Judge evaluators in human agreement, latency, and API cost?

## 5.1 Experimental Setup

**Datasets.** We use two standard KGQA benchmarks, WebQuestionsSP (WebQSP) (Yih et al., 2016) and Complex WebQuestions (CWQ) (Talmor & Berant, 2018), both of which contain natural-language questions paired with gold answer entities. For human meta-evaluation, we evaluate generated responses from four recent generative KGQA systems: PoG (Chen et al., 2024b), LMP (Wan et al., 2025), GCR (Luo et al., 2025), and RJE (Lin et al., 2025). PoG and LMP provide publicly released prediction results on WebQSP and CWQ. For GCR and RJE, which do not release per-instance predictions, we reproduce their outputs using the official implementations.

**Evaluation metrics.** For human-alignment experiments, we use human annotations as the oracle of semantic correctness and report evaluator accuracy (**Acc.**), the signed accuracy gap from the human oracle ($\Delta$ **Acc.**), and instance-level disagreement with human judgments ($\Delta$ **Eval.**). The human annotation protocol, including annotator blinding, majority-vote aggregation, and inter-annotator agreement, is described in Appendix D. For efficiency analysis, we report human agreement, runtime, and evaluation cost.

**Implementation details.** For each question, we construct matched negative-control alias sets once before model evaluation and reuse them across all evaluated systems. We sample $B = 50$ controls from the benchmark answer inventory and match them to the gold alias set by answer type, alias statistics, entity frequency, and neighbor entity. The contamination probability is estimated using the empirical collision rate in Eq. (12). Unless otherwise stated, we set $\tau = 1$, $\epsilon = 10^{-8}$, and use $\delta = 0.5$ only for binarizing ADE scores. Additional implementation details are provided in Appendix C and Appendix F.

Table 1: Human meta-evaluation results on WebQSP and CWQ. We report instance-level disagreement with human judgments (Δ Eval.; lower is better).

| Evaluator | WebQSP | | CWQ | |
|---|---|---|---|---|
| | PoG | LMP | PoG | LMP |
| Best String-Matching Evaluator | 4.84% | 8.01% | 7.20% | 9.47% |
| LLM-as-Judge (GPT-4o-mini) | 3.20% | **6.70%** | 4.50% | 7.50% |
| ADE (Ours) | **2.94%** | 7.92% | **4.40%** | **6.90%** |

### 5.2 Human Meta-Evaluation of Evaluators

Table 1 summarizes human meta-evaluation results across WebQSP and CWQ, while the full evaluator breakdown is provided in Appendix H. We focus on instance-level disagreement with human semantic judgments, since aggregate accuracy alone can hide compensating false positives and false negatives. Across all four model–dataset settings, ADE obtains lower disagreement than the best post-hoc string-matching evaluator. This shows that the gain is not merely due to selecting a different string-matching configuration, but comes from explicitly discounting contamination-prone positive matches. Compared with GPT-4o-mini Judge, ADE achieves lower disagreement on three of the four settings and remains competitive on LMP-WebQSP, while requiring no external judge model at evaluation time. These results support RQ1: ADE provides a substantially more reliable judge-free evaluator than standard string matching and offers a strong lightweight alternative to LLM-as-Judge evaluation.

### 5.3 Leaderboard Sensitivity under Evaluator Implementation Choices

To answer RQ2, we examine whether evaluator implementation choices can change the relative ordering of recent generative KGQA systems. We compare three representative methods, LMP (Wan et al., 2025), GCR (Luo et al., 2025), and RJE (Lin et al., 2025). Figure 1 shows that the resulting leaderboard is highly sensitive to evaluator configuration. Although the originally reported scores are nearly tied, with GCR at 72.7 and both LMP and RJE at 72.6, re-evaluating all systems under shared string-matching configurations changes both absolute scores and rankings. For example, GCR remains first under *w/o Align + w/ Norm + NS* and *w/o Align + w/ Norm + S*, whereas LMP becomes first under *w/ Align + w/o Norm + NS*. This indicates that seemingly minor choices in alias alignment, normalization, and matching strictness can interact with model generation style and alter leaderboard conclusions. ADE provides a more conservative contamination-aware view by discounting evaluator-positive cases with high estimated lexical-collision risk. In this setting, ADE ranks GCR first, followed by LMP and RJE, matching the human semantic ranking in Figure 1. These results show that ADE is not merely another string-matching variant: by explicitly discounting contamination-prone positives, it produces a ranking that is more stable under evaluator implementation changes and agrees with the human semantic ranking in this setting.

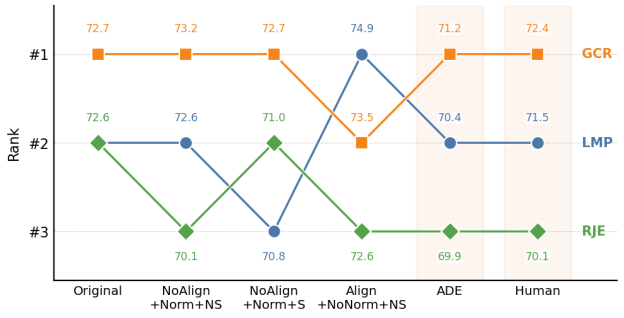
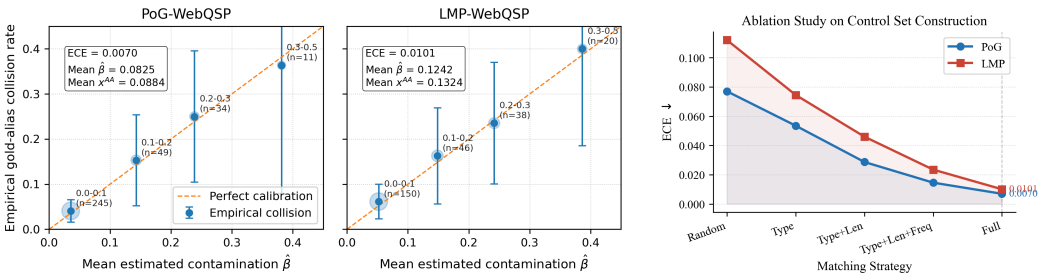



Figure 1: Leaderboard reshaping under evaluator bias. Markers denote method ranks, and the final columns compare ADE-based and human semantic rankings.

### 5.4 Exchangeability Diagnostics for Matched Negative Controls

ADE relies on the assumption that, under semantic incorrectness and matched covariates, the gold alias set and matched negative-control alias sets exhibit similar lexical-collision behavior. We diagnose this assumption on human-incorrect WebQSP instances, where gold-alias matches correspond to semantic-null collisions. As shown in Figure 2(a), the matched estimator is well calibrated, achieving




(a) Semantic-null calibration

(b) Control-construction ablation (ECE)

Figure 2: Exchangeability diagnostics for matched negative controls on WebQSP. (a) Calibration of contamination estimates $\hat{\beta}_i$. (b) Control-set construction ablation using semantic-null ECE.

ECE $= 0.0070$ on PoG-WebQSP and ECE $= 0.0101$ on LMP-WebQSP. The estimated contamination probabilities closely track empirical gold-alias collision rates across bins.

We further evaluate different control-construction strategies, from random controls to progressively richer matching. Figure 2(b) shows that richer matching consistently improves semantic-null calibration, reducing ECE from $0.0770$ to $0.0070$ on PoG-WebQSP and from $0.1120$ to $0.0101$ on LMP-WebQSP. These results do not prove exact exchangeability, but provide empirical evidence that matched negative controls better approximate semantic-null collision behavior than random controls. Full diagnostics and additional analyses are provided in Appendix E and Appendix I.2.

### 5.5 Judge Comparison and Efficiency

Finally, we compare ADE with LLM-as-Judge evaluators to examine whether it can provide a cost-efficient alternative to judge-based semantic evaluation. We evaluate GPT-3.5-turbo, GPT-4o-mini, DeepSeek-V3, and ADE on the human-annotated PoG-WebQSP and LMP-WebQSP sets,the exact judge prompt, visible input fields, decoding parameters, output parsing rule, and cost/latency accounting procedure are provided in Appendix G.1. Table 2 reports the quality–efficiency trade-off: GPT-4o-mini is the strongest LLM judge overall, GPT-3.5-turbo substantially underestimates semantic correctness, and DeepSeek-V3 shows higher instance-level disagreement. ADE achieves the lowest disagreement on PoG-WebQSP and remains competitive on LMP-WebQSP, while requiring no prompt tokens, completion tokens, API calls, or external model availability. Thus, ADE is not a universal replacement for strong LLM judges, but provides a deterministic and substantially cheaper judge-free complement with competitive human alignment in our evaluated settings. Pairwise agreement matrices and full Acc. / Δ Acc. results are provided in Appendix G.2.

Table 2: Quality–efficiency comparison.

| Evaluator | Δ Eval. | | Cost | Lat. |
|---|---|---|---|---|
| | PoG | LMP | | |
| GPT-3.5 | 19.95 | 32.82 | $0.1966 | 0.731s |
| GPT-4o-mini | 3.20 | **6.70** | $0.0554 | 0.842s |
| DeepSeek-V3 | 9.41 | 10.08 | $0.0713 | 2.437s |
| **ADE** | **2.94** | <u>7.92</u> | $0 | **0.007s** |

## 6 Conclusion

We studied evaluator bias in generative KGQA, where free-form responses make string-based metrics vulnerable to lexical mismatch and contamination. We proposed the Analytic Debiasing Evaluator (ADE), a judge-free framework that uses matched negative controls to estimate contamination risk and discount unreliable matches. Experiments show that ADE improves alignment with human judgments over string-matching baselines in evaluated KGQA settings while providing a deterministic and cost-efficient complement to LLM-as-Judge evaluation.

### AI USE STATEMENT

In this work, we used generative AI tools for editing and improving the readability of the manuscript, brainstorming and refining research presentation, assisting with the design of figures, and summarizing or analyzing existing literature. We also used generative AI tools to assist with minor software implementation and debugging tasks. All AI-assisted outputs were reviewed, verified, and revised by the authors before being incorporated into the final manuscript.

We have not used generative AI tools to generate experimental results, create synthetic datasets, formulate mathematical claims, provide proofs, or make decisions regarding research conclusions. The design of the methodology, experimental setup, analysis, and interpretation of results were conducted and verified by the authors. Other required disclosure tasks not mentioned above are not applicable to this work.

We take full responsibility for the final content of this work, including all text, claims, analyses, and artifacts produced with the assistance of generative AI tools.

### ETHICS STATEMENT

This work studies evaluator bias in generative KGQA systems and does not involve the collection or release of personal or sensitive information. Our human meta-evaluation relies on human judgments of model-generated responses. Annotators were informed about the purpose of the evaluation and were only asked to assess the semantic correctness of generated answers. No personally identifiable information was collected during annotation.

This work focuses on analyzing and reducing biases introduced by automatic evaluation mechanisms. The proposed ADE framework is intended to provide a more reliable evaluation signal and should not be interpreted as a replacement for human judgment in high-stakes decision-making scenarios. All experiments and analyses were conducted for research purposes, and we take responsibility for the ethical use and interpretation of the proposed method.

### REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed ADE framework, including the contamination estimation procedure, reliability scoring formulation, and theoretical assumptions in Sections 4.2 and 4.3, with complete proofs and additional analysis provided in the appendix. The datasets, preprocessing procedures, human annotation protocols, and evaluation settings are described in Sections 5 and the appendix. We also provide implementation details, hyperparameter settings, and ablation studies to facilitate reproduction of all reported results.

## REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In Kristina Toutanova and Hua Wu (eds.), *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1415–1425, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1133. URL https://aclanthology.org/P14-1133/.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 291–305, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.20. URL https://aclanthology.org/2022.emnlp-main.20/.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Evaluating question answering evaluation. In Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi

Chen (eds.), *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pp. 119–124, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-5817. URL https://aclanthology.org/D19-5817/.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or LLMs as the judge? a study on judgement bias. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8301–8327, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.474. URL https://aclanthology.org/2024.emnlp-main.474/.

Liyi Chen, Panrong Tong, Zhongming Jin, Ying Sun, Jieping Ye, and Hui Xiong. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=CwCUEr6wO5.

Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku. UHop: An unrestricted-hop relation extraction framework for knowledge-based question answering. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *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)*, pp. 345–356, Minneapolis, Minnesota, June 2019b. Association for Computational Linguistics. doi: 10.18653/v1/N19-1031. URL https://aclanthology.org/N19-1031/.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Songhee Han, Jueun Shin, Jiyoon Han, Bung-Woo Jun, and Hilal Ayan Karabatman. How trustworthy are llm-as-judge ratings for interpretive responses? implications for qualitative research workflows, 2026. URL https://arxiv.org/abs/2604.00008.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=MPJ3oXtTZl.

Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, Yang Song, Chen Zhu, Hengshu Zhu, and Ji-Rong Wen. KG-agent: An efficient autonomous agent framework for complex reasoning over knowledge graph. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9505–9523, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.468. URL https://aclanthology.org/2025.acl-long.468/.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. Evaluating open-domain question answering in the era of large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5591–5606, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.307. URL https://aclanthology.org/2023.acl-long.307/.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8euJaTveKw.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2757–2791,

Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.138. URL https://aclanthology.org/2025.emnlp-main.138/.

Chen Liang, Jonathan Berant, Quoc Le, Kenneth Forbus, and Ni Lao. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23–33, 2017.

Xujian Liang and Zhaoquan Gu. Fast think-on-graph: Wider, deeper and faster reasoning of large language model on knowledge graph, 2025. URL https://arxiv.org/abs/2501.14300.

Can Lin, Zhengwang Jiang, Ling Zheng, Qi Zhao, Yuhang Zhang, Qi Song, and Wangqiu Zhou. RJE: A retrieval-judgment-exploration framework for efficient knowledge graph question answering with LLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17277–17294, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.873. URL https://aclanthology.org/2025.emnlp-main.873/.

Marc Lipsitch, Eric Tchetgen Tchetgen, and Ted Cohen. Negative controls: a tool for detecting confounding and bias in observational studies. *Epidemiology*, 21(3):383–388, 2010.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Runxuan Liu, Bei Luo, Jiaqi Li, Baoxin Wang, Ming Liu, Dayong Wu, Shijin Wang, and Bing Qin. Ontology-guided reverse thinking makes large language models stronger on knowledge graph question answering. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15269–15284, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.741. URL https://aclanthology.org/2025.acl-long.741/.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Haoran Luo, Haihong E, Zichen Tang, Shiyao Peng, Yikai Guo, Wentai Zhang, Chenghao Ma, Guanting Dong, Meina Song, Wei Lin, Yifan Zhu, and Anh Tuan Luu. ChatKBQA: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2039–2056, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.122. URL https://aclanthology.org/2024.findings-acl.122/.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=ZGNWW7xZ6Q.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Fr7kH2SFq7.

Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=oFBu7qaZpS.

Julian Risch, Timo Möller, Julian Gutsch, and Malte Pietsch. Semantic answer similarity for evaluating question answering models. In Adam Fisch, Alon Talmor, Danqi Chen, Eunsol Choi, Minjoon Seo, Patrick Lewis, Robin Jia, and Sewon Min (eds.), *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pp. 149–157, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.mrqa-1.15. URL https://aclanthology.org/2021.mrqa-1.15/.

Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. Verbosity bias in preference labeling by large language models, 2023. URL https://arxiv.org/abs/2310.10076.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7881–7892, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.704. URL https://aclanthology.org/2020.acl-main.704/.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in LLM-as-a-judge. In Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh (eds.), *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 292–314, Mumbai, India, December 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. ISBN 979-8-89176-298-5. doi: 10.18653/v1/2025.ijcnlp-long.18. URL https://aclanthology.org/2025.ijcnlp-long.18/.

Xu Shi, Wang Miao, and Eric Tchetgen Tchetgen. A selective review of negative control methods in epidemiology. *Current epidemiology reports*, 7(4):190–202, 2020.

Yiheng Shu, Zhiwei Yu, Yuhan Li, Börje F. Karlsson, Tingting Ma, Yuzhong Qu, and Chin-Yew Lin. TIARA: Multi-grained retrieval for robust question answering over large knowledge base. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8108–8121, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.555. URL https://aclanthology.org/2022.emnlp-main.555/.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nnVO1PvbTv.

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 641–651, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1059. URL https://aclanthology.org/N18-1059/.

Junhong Wan, Tao Yu, Kunyu Jiang, Yao Fu, Weihao Jiang, and Jiang Zhu. Digest the knowledge: Large language models empowered message passing for knowledge graph question answering. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15426–15442, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.750. URL https://aclanthology.org/2025.acl-long.750/.

Yidong Wang, Zhuohao Yu, Wenjin Yao, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. PandaLM: An automatic evaluation benchmark for LLM instruction tuning optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5Nn2BLV7SB.

Yingxu Wang, Shiqi Fan, Mengzhu Wang, Siyang Gao, Chao Wang, and Nan Yin. DAMR: Efficient and adaptive context-aware knowledge graph question answering with LLM-guided MCTS. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=mUx7WLC8q6.

Guanming Xiong, Junwei Bao, and Wen Zhao. Interactive-KBQA: Multi-turn interactions for knowledge base question answering with large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10561–10582, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.569. URL https://aclanthology.org/2024.acl-long.569/.

Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Jun Zhao, and Kang Liu. Generate-on-graph: Treat LLM as both agent and KG for incomplete knowledge graph question answering. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18410–18430, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1023. URL https://aclanthology.org/2024.emnlp-main.1023/.

Shiqi Yan, Yubo Chen, Ruiqi Zhou, Zhengxi Yao, Shuai Chen, Tianyi Zhang, Shijie Zhang, Wei-Qiang Zhang, Yongfeng Huang, Haixin Duan, and Yunqi Zhang. Explore-on-graph: Incentivizing autonomous exploration of large language models on knowledge graphs with path-refined reward modeling. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=NfuBj8jleE.

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6032–6043, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.417. URL https://aclanthology.org/2022.acl-long.417/.

Wen-tau Yih, Xiaodong He, and Christopher Meek. Semantic parsing for single-relation question answering. In Kristina Toutanova and Hua Wu (eds.), *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 643–648, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2105. URL https://aclanthology.org/P14-2105/.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In Chengqing Zong and Michael Strube (eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1321–1331, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1128. URL https://aclanthology.org/P15-1128/.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 201–206, 2016.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating large language models at evaluating instruction following. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tr0KidwPLc.

Liangliang Zhang, Zhuorui Jiang, Hongliang Chi, Haoyang Chen, Mohammed ElKoumy, Fali Wang, Qiong Wu, Zhengyi Zhou, Shirui Pan, Suhang Wang, and Yao Ma. Diagnosing and addressing pitfalls in KG-RAG datasets: Toward more reliable benchmarking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=Vd5JXiX073.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=uccHPGDlao.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned large language models are scalable judges. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xsELpEPn4A.

## APPENDIX CONTENTS


**A Proofs** **17**

A.1 Proof of Proposition 1 . . . . . 17

A.2 Proof of Corollary 3.1 . . . . . 17

A.3 Proof of Corollary 3.2 . . . . . 18

A.4 Proof of Proposition 2 . . . . . 18

**B KGQA Evaluator Landscape** **19**

B.1 Naming Inconsistency: Similar Metrics under Different Names . . . . . 19

B.2 Implementation Inconsistency: Similar Metrics under Different Rules . . . . . 19

**C ADE Implementation Details** **20**

C.1 Alias-Aware Matching . . . . . 20

C.2 Matched Negative-Control Construction . . . . . 20

C.3 Contamination Estimation and Information Boundary . . . . . 21

**D Annotation Protocol for Human Meta-Evaluation** **22**

**E Empirical Diagnostics for the Exchangeability Assumption** **23**

E.1 Covariate Balance . . . . . 23

E.2 Semantic-null Calibration . . . . . 23

E.3 Control-Construction Ablation . . . . . 24

**F Ablation and Sensitivity Analysis** **24**

F.1 Component Ablation . . . . . 24

F.2 Hyperparameter Sensitivity . . . . . 25

**G LLM-as-Judge Configuration and Agreement** **26**

G.1 LLM-as-Judge Configuration . . . . . 26

G.2 Pairwise Agreement among Evaluators . . . . . 27

**H Full Human Meta-Evaluation Results** **28**

**I Additional Mechanism Analysis** **28**

I.1 Alias Expansion: False Negatives and False Positives . . . . . 29

I.2 Verbosity-Associated False Positives . . . . . 30

**J Limitations** **32**

**K Broader Impacts** **32**

**L Licenses and Terms of Use for Existing Assets** **32**

# A PROOFS

## A.1 PROOF OF PROPOSITION 1

Since $x_i \in \{0, 1\}$, we have

$$\mathbb{E}[x_i \mid q_i] = \Pr(x_i = 1 \mid q_i). \tag{20}$$

Applying the law of total probability over $Y_i$ gives

$$\begin{aligned}\Pr(x_i = 1 \mid q_i) = \Pr(x_i = 1 \mid Y_i = 1, q_i) \Pr(Y_i = 1 \mid q_i) \\ + \Pr(x_i = 1 \mid Y_i = 0, q_i) \Pr(Y_i = 0 \mid q_i).\end{aligned} \tag{21}$$

By definition,

$$\Pr(x_i = 1 \mid Y_i = 1, q_i) = 1 - \alpha_i, \qquad \Pr(x_i = 1 \mid Y_i = 0, q_i) = \beta_i, \tag{22}$$

and

$$\Pr(Y_i = 1 \mid q_i) = l_i, \qquad \Pr(Y_i = 0 \mid q_i) = 1 - l_i. \tag{23}$$

Substituting these expressions yields

$$\mathbb{E}[x_i \mid q_i] = l_i(1 - \alpha_i) + (1 - l_i)\beta_i. \tag{24}$$

Averaging over instances gives

$$\mathbb{E}[\bar{x}] = \frac{1}{n} \sum_{i=1}^{n} \left[ l_i(1 - \alpha_i) + (1 - l_i)\beta_i \right]. \tag{25}$$

Subtracting $L = \frac{1}{n} \sum_{i=1}^{n} l_i$ further gives

$$\mathbb{E}[\bar{x}] - L = \frac{1}{n} \sum_{i=1}^{n} \left[ (1 - l_i)\beta_i - l_i\alpha_i \right]. \tag{26}$$

This completes the proof.

## A.2 PROOF OF COROLLARY 3.1

For a model $M$, Proposition 1 gives

$$\begin{aligned}\mathbb{E}[\bar{x}_M] &= \frac{1}{n} \sum_{i=1}^{n} \left[ l_{i,M}(1 - \alpha) + (1 - l_{i,M})\beta_{i,M} \right] \\ &= L_M(1 - \alpha) + \frac{1}{n} \sum_{i=1}^{n} (1 - l_{i,M})\beta_{i,M}.\end{aligned} \tag{27}$$

By Eq. equation 7,

$$\frac{1}{n} \sum_{i=1}^{n} (1 - l_{i,M})\beta_{i,M} = (1 - L_M)\bar{\beta}_M. \tag{28}$$

Therefore,

$$\mathbb{E}[\bar{x}_M] = L_M(1 - \alpha) + (1 - L_M)\bar{\beta}_M. \tag{29}$$

A ranking inversion in expectation occurs when

$$\mathbb{E}[\bar{x}_B] > \mathbb{E}[\bar{x}_A]. \tag{30}$$

Substituting the expressions above gives

$$L_B(1 - \alpha) + (1 - L_B)\bar{\beta}_B > L_A(1 - \alpha) + (1 - L_A)\bar{\beta}_A. \tag{31}$$

Rearranging terms,

$$(1 - L_B)(\bar{\beta}_B - \bar{\beta}_A) > (L_A - L_B)(1 - \alpha - \bar{\beta}_A). \tag{32}$$

Since $L_B < 1$, dividing both sides by $1 - L_B$ gives

$$\bar{\beta}_B - \bar{\beta}_A > \frac{(L_A - L_B)(1 - \alpha - \bar{\beta}_A)}{1 - L_B}. \tag{33}$$

Thus, under this condition, $\mathbb{E}[\bar{x}_B] > \mathbb{E}[\bar{x}_A]$ even though $L_A > L_B$. This completes the proof.

### A.3 Proof of Corollary 3.2

Let

$$m_i := \mathbb{E}[x_i \mid q_i]. \tag{34}$$

From Proposition 1,

$$m_i = l_i(1-\alpha_i) + (1-l_i)\beta_i. \tag{35}$$

The scalar quantity $m_i$ depends jointly on three unknown quantities: $l_i$, $\alpha_i$, and $\beta_i$. Therefore, without additional assumptions on the evaluator error rates, the latent semantic correctness propensity $l_i$ is not identified.

To see this directly, fix any $m_i \in [0,1]$. The following construction sets

$$l_i = 0, \qquad \alpha_i = 0, \qquad \beta_i = m_i, \tag{36}$$

which yields

$$l_i(1-\alpha_i) + (1-l_i)\beta_i = m_i. \tag{37}$$

Thus the same evaluator expectation is compatible with zero latent semantic correctness.

On the other hand, the alternative construction

$$l_i = 1, \qquad \alpha_i = 1 - m_i, \qquad \beta_i = 0 \tag{38}$$

also yields

$$l_i(1-\alpha_i) + (1-l_i)\beta_i = m_i. \tag{39}$$

This construction is compatible with unit latent semantic correctness.

Applying these two constructions independently to all instances shows that the same set of exact evaluator expectations $\{m_i\}_{i=1}^n$ can be induced by parameterizations with benchmark-level latent capabilities $L = 0$ and $L = 1$. Therefore, $L$ is not identifiable from observed evaluator expectations, and hence cannot be identified from the observed evaluator scores alone. This completes the proof.

### A.4 Proof of Proposition 2

For each instance $i$, define the gold-alias collision indicator under the semantic-null condition as

$$Z_i^{\text{gold}} = \text{AliasMatch}(r_i, A_i). \tag{40}$$

By definition, the semantic-null lexical contamination probability is

$$\beta_i = \Pr\Big(Z_i^{\text{gold}} = 1 \mid Y_i = 0, q_i\Big). \tag{41}$$

This is the gold-answer lexical-collision rate defined in Eq. equation 13.

For the $b$-th negative-control alias set, define

$$Z_{i,b}^{-} = \text{AliasMatch}(r_i, A_{i,b}^{-}). \tag{42}$$

Under the exchangeability assumption in Proposition 2, conditional on $(Y_i = 0, q_i)$ and the matched answer-side covariates used to construct the control pool, the gold alias set $A_i$ and the negative-control alias sets $A_{i,1}^{-}, \ldots, A_{i,B}^{-}$ have the same marginal lexical-collision probability against $r_i$. Therefore, for every $b$,

$$\mathbb{E}\left[Z_{i,b}^{-} \mid Y_i = 0, q_i\right] = \mathbb{E}\left[Z_i^{\text{gold}} \mid Y_i = 0, q_i\right] = \beta_i. \tag{43}$$

Recall that the negative-control estimator is

$$\hat{\beta}_i = \frac{1}{B}\sum_{b=1}^{B} Z_{i,b}^{-}. \tag{44}$$

Taking conditional expectation gives

$$\begin{aligned}\mathbb{E}\left[\hat{\beta}_i \mid Y_i = 0, q_i\right] &= \frac{1}{B}\sum_{b=1}^{B} \mathbb{E}\left[Z_{i,b}^{-} \mid Y_i = 0, q_i\right] \\ &= \frac{1}{B}\sum_{b=1}^{B} \beta_i = \beta_i.\end{aligned} \tag{45}$$

This proves the unbiasedness claim.

It remains to prove the concentration bound. Since each $Z^-_{i,b} \in \{0,1\}$ and the negative controls are sampled independently from the matched control pool, Hoeffding's inequality gives, for any $t > 0$,

$$\Pr\left(\left|\frac{1}{B}\sum_{b=1}^{B} Z^-_{i,b} - \mathbb{E}\left[\frac{1}{B}\sum_{b=1}^{B} Z^-_{i,b} \mid Y_i = 0, q_i\right]\right| \geq t \mid Y_i = 0, q_i\right) \leq 2\exp(-2Bt^2). \tag{46}$$

Using the unbiasedness result above, this becomes

$$\Pr\Big(|\hat{\beta}_i - \beta_i| \geq t \mid Y_i = 0, q_i\Big) \leq 2\exp(-2Bt^2). \tag{47}$$

This completes the proof.

# B KGQA Evaluator Landscape

In classical semantic-parsing-based KGQA, model outputs are typically structured queries or canonical entities, making string-based evaluation relatively straightforward and standardized. In contrast, under the generative paradigm, modern LLM-based KGQA systems often produce verbose natural-language responses, which makes string-based evaluation more vulnerable to false-positive lexical collisions. Despite this shift, the community has not reached a consensus on how to evaluate free-form KGQA outputs. Our empirical survey of recent KGQA literature from 2024–2026 reveals a fragmented evaluator landscape, where nominally similar metrics differ in alias alignment, normalization, and matching strictness.

## B.1 Naming Inconsistency: Similar Metrics under Different Names

A profound symptom of this lack of consensus is the terminological inconsistency across the literature. While the community universally relies on string-matching heuristics to approximate semantic correctness, papers report identical or fundamentally similar underlying procedures under wildly different names.

For instance, essentially the same evaluation scripts are reported as *Hits@1* or *Hit* (e.g., EoG Yan et al. (2026), RoG Luo et al. (2024b), G-Retriever He et al. (2024), GCR Luo et al. (2025), GoG Xu et al. (2024)), *Exact Match (EM)* (e.g., ToG-2 Ma et al. (2025), Interactive-KBQA Xiong et al. (2024)), or a hybrid terminology like *Exact Match Accuracy / Hits@1* (e.g., LMP Wan et al. (2025), ToG Sun et al. (2024), PoG Chen et al. (2024b), RJE Lin et al. (2025)). This fragmentation creates an illusion of diverse evaluation methodologies while obscuring the fundamental vulnerability—accidental lexical collision—shared by all these works.

## B.2 Implementation Inconsistency: Similar Metrics under Different Rules

Through a comprehensive code review of recent open-source KGQA frameworks, we categorize the existing evaluator implementations along three primary dimensions:

- **Alignment (Align):**
  - *w/ Align:* Utilizes an alias dictionary to align the predicted answer with a comprehensive set of ground-truth aliases.
  - *w/o Align:* Evaluates strictly against a single ground-truth string.
- **Normalization (Norm):**
  - *w/ Norm:* Full string normalization (e.g., converting to lowercase, removing punctuation, filtering articles like "a/an/the", and fixing whitespaces).
  - *w/o Norm:* Simple normalization (e.g., only lowering cases and stripping trailing spaces).
- **Measurement (Measure):** The logical condition used to determine a match.
  - *Non-Strict (NS):* Bidirectional inclusion or exact equality (`pred == ans || ans in pred || pred in ans`). This is highly vulnerable to accidental collisions.

- *Strict (S):* Unidirectional inclusion or exact equality (`pred == ans || ans in pred`).
- *More Strict (MS):* Strict equality only (`pred == ans`).

Based on this taxonomy, we observe vastly different hurdle rates for what qualifies as a "correct" answer. Rather than a standardized metric, we find five distinct implementation clusters in recent literature, summarized in Table 3.

Table 3: Implementation variance of generative KGQA evaluation scripts across recent works.

| Align | Norm | Measure | Representative Works |
|---|---|---|---|
| ✓ | ✓ | MS | Interactive-KBQA *(ACL'24)* |
| ✓ | × | NS | RJE *(EMNLP'25)*, PoG *(NeurIPS'24)* |
| × | ✓ | NS | LMP *(ACL'25)* |
| × | ✓ | S | GCR *(ICML'25)*, RoG *(ICLR'24)*, G-Retriever *(NeurIPS'24)* |
| × | × | NS | EoG *(ICLR'26)*, ToG-2 *(ICLR'25)*, GoG *(EMNLP'24)*, ToG *(ICLR'24)* |

# C ADE Implementation Details

This appendix provides the implementation details of ADE that are omitted from the main text for space. These details are not additional modeling assumptions beyond Section 4; they specify how the matched negative controls and alias-aware matcher are instantiated in our experiments.

## C.1 Alias-Aware Matching

ADE uses the same normalization function for gold aliases, negative-control aliases, and model responses. We normalize strings by lowercasing, stripping leading and trailing spaces, removing punctuation, normalizing whitespace, and applying answer-type-specific canonicalization for dates, numbers, and entity names when available. After normalization, duplicate aliases are removed within each alias set.

Given a response $r_i$ and an alias set $A$, the alias-aware matcher returns

$$\mathrm{AliasMatch}(r_i, A) = \mathbb{I}\left[\exists a \in A : \ \mathrm{match}(\mathrm{norm}(r_i), \mathrm{norm}(a)) = 1\right],$$

where $\mathrm{match}(\cdot,\cdot)$ follows the inclusion/equality rule of the base string-matching evaluator being corrected. For set-valued questions, an output is treated as evaluator-positive if it matches at least one valid gold answer alias, following the common KGQA evaluation convention used in our human annotation protocol.

## C.2 Matched Negative-Control Construction

For each question $q_i$, ADE constructs a candidate pool $\mathcal{C}_i$ from the benchmark/KG answer inventory. An alias set is eligible as a negative control only if it is answer-like but semantically invalid for $q_i$. We therefore remove: (i) the gold answer entity and all of its aliases; (ii) all alternative gold answers for the same question; (iii) any candidate whose normalized alias overlaps with a normalized gold alias; (iv) candidates with the same canonical entity identifier as a gold answer, when such identifiers are available; and (v) candidates whose coarse answer type is incompatible with the gold answer type. For multi-answer questions, a candidate is retained only if it does not correspond to any annotated gold answer. This filtering ensures that negative controls are not trivially correct answers or surface variants of the gold answer. The remaining candidates are used only to estimate lexical collision risk; human semantic labels are never used to construct $\mathcal{C}_i$.

For each gold alias set $A_i$ and each candidate control alias set $A_j \in \mathcal{C}_i$, ADE computes answer-side covariates that affect the probability of accidental lexical matching. The full ADE configuration uses

answer type, alias-set size, alias length, entity frequency, and neighbor-entity matching, corresponding to the matching strategy described in the main text.

Answer type is represented by a coarse category such as entity, date, number, string, or set-valued answer. When KG type information is available, we additionally use the entity type provided by the KG or benchmark metadata. Alias-set size is defined as the number of unique normalized aliases:

$$s(A) = |\{\mathrm{norm}(a) : a \in A\}|.$$

Alias length is computed as the average number of normalized tokens across aliases:

$$\ell(A) = \frac{1}{|A|} \sum_{a \in A} |\mathrm{tok}(\mathrm{norm}(a))|.$$

Entity frequency measures how often the aliases of an answer candidate appear in the benchmark/KG textual corpus:

$$f(A) = \log \left( 1 + \sum_{a \in A} \mathrm{count}_{\mathcal{D}}(\mathrm{norm}(a)) \right),$$

where $\mathcal{D}$ denotes the benchmark/KG textual corpus used for computing corpus-level entity frequency.

Neighbor-entity matching measures whether a candidate answer occupies a similar local KG neighborhood to the gold answer. For an alias set $A$ associated with a canonical entity, let $\mathcal{N}(A)$ denote the set of one-hop neighboring entity identifiers of that entity in the KG. For two alias sets $A_i$ and $A_j$, we define the neighbor-entity distance as

$$d_{\mathrm{nbr}}(A_i, A_j) = 1 - \frac{|\mathcal{N}(A_i) \cap \mathcal{N}(A_j)|}{|\mathcal{N}(A_i) \cup \mathcal{N}(A_j)| + \epsilon}.$$

When KG neighborhood information is unavailable for either candidate, we set $d_{\mathrm{nbr}}(A_i, A_j) = 1$. This covariate encourages negative controls to be drawn from candidates that are locally similar to the gold answer in the KG, rather than from arbitrary entities with unrelated neighborhoods.

ADE first restricts $\mathcal{C}_i$ to candidates with the same coarse answer type as the gold answer. Among the remaining candidates, we rank controls by a standardized covariate distance:

$$\begin{aligned} d(A_i, A_j) = \lambda_s &| \log(s(A_i) + 1) - \log(s(A_j) + 1)| \\ &+ \lambda_\ell |\ell(A_i) - \ell(A_j)| \\ &+ \lambda_f |f(A_i) - f(A_j)| \\ &+ \lambda_n d_{\mathrm{nbr}}(A_i, A_j). \end{aligned}$$

In our experiments, scalar covariates are standardized before distance computation, and we use equal weights unless otherwise stated. We then form a matched pool from the nearest candidates and sample $B = 50$ negative-control alias sets independently with replacement for each question.

The sampled negative controls are constructed once per question and fixed across all evaluated systems. This design prevents ADE from selecting different controls for different models and ensures that score differences are caused by model responses rather than by evaluator resampling.

### C.3 Contamination Estimation and Information Boundary

For each sampled negative-control alias set $A^-_{i,b}$, ADE applies the same alias-aware matcher used for the gold answer:

$$z^-_{i,b} = \mathrm{AliasMatch}(r_i, A^-_{i,b}).$$

The empirical negative-control collision rate is

$$\hat{\beta}_i = \frac{1}{B} \sum_{b=1}^{B} z^-_{i,b}.$$

If the gold alias-aware match is negative, ADE assigns score zero. If the gold alias-aware match is positive, ADE computes the reliability score

$$S^{(i)}_{\mathrm{ADE}} = \frac{1}{1 + \tau \frac{\hat{\beta}_i}{1 - \hat{\beta}_i + \epsilon}}.$$

Table 4: Summary of human annotation statistics and inter-annotator agreement across evaluated KGQA systems.

| Set | Instances | Unanimous | Pairwise Agr. | Fleiss' $\kappa$ |
|---|---|---|---|---|
| PoG-WebQSP | 1,639 | 84.7% | 89.8% | 0.80 |
| LMP-WebQSP | 1,639 | 86.7% | 91.1% | 0.82 |
| PoG-CWQ | 3,531 | 80.3% | 86.9% | 0.74 |
| LMP-CWQ | 3,531 | 82.9% | 88.6% | 0.77 |
| GCR-CWQ | 3,531 | 83.3% | 85.5% | 0.74 |
| RJE-CWQ | 3,531 | 79.9% | 81.2% | 0.76 |

In all main experiments, we use $\tau = 1$ and $\epsilon = 10^{-8}$. The threshold $\delta = 0.5$ is used only when converting soft ADE scores into binary decisions for instance-level agreement analysis; benchmark-level ADE scores are computed by averaging the soft reliability scores.

ADE uses only information available to an automatic KGQA evaluator: the question, the model response, the benchmark gold answer aliases, the answer/entity inventory, and KG metadata used to construct answer types, entity frequencies, and neighbor-entity features. It does not use human semantic labels, LLM judge predictions, model identities, or leaderboard outcomes when constructing negative controls or computing ADE scores. Human labels are used only for meta-evaluating evaluators in Section 5.

## D Annotation Protocol for Human Meta-Evaluation

We provide additional details on the semantic correctness labels used in our human meta-evaluation. We evaluate four representative generative KGQA systems, including PoG (Chen et al., 2024b), LMP (Wan et al., 2025), GCR (Luo et al., 2025), and RJE (Lin et al., 2025). PoG and LMP are selected because their prediction results are publicly released, allowing us to directly annotate real outputs from recent generative KGQA systems. GCR and RJE are included because their original papers employ evaluation scripts different from those used by PoG and LMP, enabling a broader comparison across evaluator implementations. Since GCR and RJE do not release per-instance predictions, we reproduce their results using the official implementations and annotate the generated CWQ outputs under the same semantic correctness annotation protocol.

**Annotation interface.** Each instance was annotated by three NLP graduate students. Annotators were shown the question, the gold answer list, and anonymized model responses. They were not shown model names, automatic evaluator scores, ADE scores, LLM-judge predictions, dataset identifiers, or leaderboard results. For each question, the order of model responses was randomized independently to reduce position bias.

**Annotation criteria.** Annotators were instructed to judge whether each model response semantically answers the question, rather than whether it exactly matches the gold answer string. An output was labeled correct if it expressed a valid answer to the question. Surface-form differences, aliases, paraphrases, and minor formatting differences were ignored as long as the answer was semantically equivalent to a gold answer. For questions with multiple gold answers, we followed the common KGQA evaluation convention and labeled an output as correct if it correctly provided at least one gold answer. An output was labeled incorrect if it failed to answer the question, gave a wrong entity, merely mentioned the gold answer without selecting it as the answer, or listed conflicting candidates without making a correct final commitment.

**Aggregation and disagreement handling.** Each model response received three independent binary labels. The final semantic correctness label was obtained by majority vote. We did not use automatic evaluator outputs, ADE scores, or LLM-judge predictions to resolve annotator disagreements. Instances without unanimous agreement were retained rather than discarded, and the majority-vote labels were used only for evaluator meta-evaluation and mechanism analysis. They were not used by ADE during test-time evaluation. We report inter-annotator agreement statistics in Table 4.

Table 5: Balance of scalar matching covariates between gold alias sets and negative-control alias sets. Lower SMD indicates better balance. The table reports scalar lexical and frequency covariates; the neighbor-entity component in the full matcher is evaluated through downstream semantic-null calibration in Tables 8 and 9.

| Match level | Alias-set size | Mean alias length | Entity frequency |
|---|---|---|---|
| Random | 2.4266 | 0.9364 | 15.5538 |
| Type only | 1.6130 | 1.1959 | 27.7948 |
| Type + length | 0.2953 | 0.4618 | 1.5860 |
| Type + length + frequency | 0.4619 | 0.6186 | 0.2778 |
| Full | 0.4649 | 0.6166 | 0.2893 |

# E Empirical Diagnostics for the Exchangeability Assumption

This appendix provides the full diagnostic results for the matched negative-control assumption used by ADE. All diagnostics in this appendix use $B = 50$ negative controls. For analyses that require ADE reliability scoring or binarization, we use $\tau = 1.0$ and $\delta = 0.5$. Since the diagnostics below evaluate the negative-control contamination estimate $\hat{\beta}_i$, the reliability-scoring parameters $\tau$ and $\delta$ do not affect the covariate-balance or semantic-null calibration results directly.

ADE relies on the assumption that, under semantic incorrectness and after matching on collision-relevant covariates, the gold alias set and the negative-control alias sets exhibit similar lexical-collision behavior. The following experiments are intended as empirical diagnostics of this assumption. They do not prove exact exchangeability; rather, they test whether matched controls approximate semantic-null gold-alias collision behavior better than random or weakly matched controls.

## E.1 Covariate Balance

We first measure covariate balance between gold alias sets and sampled negative-control alias sets using standardized mean difference (SMD). Lower SMD indicates better balance for the corresponding covariate. We compare five control-construction strategies: random controls, type-only matching, type-and-length matching, type-length-frequency matching, and the full matching strategy used by ADE.

The SMD results show that different matching criteria improve different scalar covariates. For example, type-and-length matching yields the strongest balance on alias-set size and alias length, while frequency-aware matching substantially improves entity-frequency balance. The full matching strategy additionally incorporates neighbor-entity matching, which is not summarized as a scalar SMD in this table but is evaluated through the downstream calibration results below. This is expected because ADE aims to match the lexical-collision behavior of the gold alias set, rather than independently minimizing every covariate imbalance. The downstream calibration results below show that the full matching strategy best approximates empirical gold-alias collision rates.

## E.2 Semantic-null Calibration

We next evaluate whether the negative-control collision estimate $\hat{\beta}_i$ is calibrated to the empirical gold-alias collision rate under semantic incorrectness. For this diagnostic, we restrict to human-incorrect instances. On these instances, a gold-alias match corresponds to a semantic-null lexical collision. We bin instances by $\hat{\beta}_i$ and compare the mean estimated contamination probability with the empirical gold-alias collision rate in each bin.

Tables 6 and 7 show that $\hat{\beta}_i$ closely tracks the empirical gold-alias collision rate on human-incorrect instances. For PoG-WebQSP, the mean contamination estimate under full matching is close to the empirical collision rate across bins, yielding ECE $= 0.0070$. For LMP-WebQSP, the same diagnostic yields ECE $= 0.0101$. Higher-risk bins contain fewer instances, so their estimates are naturally less stable, but the overall calibration remains close.

Table 6: Semantic-null calibration on human-incorrect PoG-WebQSP instances using full matching. The full matched estimator achieves ECE $= 0.0070$.

| Bin low | Bin high | Mean $\hat{\beta}$ | Collision rate | $n$ |
|---|---|---|---|---|
| 0.0 | 0.1 | 0.0354 | 0.0407 | 245 |
| 0.1 | 0.2 | 0.1428 | 0.1531 | 49 |
| 0.2 | 0.3 | 0.2386 | 0.2500 | 34 |
| 0.3 | 0.5 | 0.3815 | 0.3636 | 11 |

Table 7: Semantic-null calibration on human-incorrect LMP-WebQSP instances using full matching. The full matched estimator achieves ECE $= 0.0101$.

| Bin low | Bin high | Mean $\hat{\beta}$ | Collision rate | $n$ |
|---|---|---|---|---|
| 0.0 | 0.1 | 0.0521 | 0.0614 | 150 |
| 0.1 | 0.2 | 0.1485 | 0.1628 | 46 |
| 0.2 | 0.3 | 0.2417 | 0.2353 | 38 |
| 0.3 | 0.5 | 0.3862 | 0.4000 | 20 |

### E.3 Control-Construction Ablation

Finally, we evaluate whether richer control matching improves both calibration and discrimination. For each control-construction strategy, we report ECE, Spearman rank correlation between $\hat{\beta}_i$ and gold-alias collision, AUROC, AUPRC, the mean estimated contamination probability, and the empirical gold-alias collision rate under the alias-aware matcher.

The ablation results show a consistent trend across both PoG-WebQSP and LMP-WebQSP. Random controls substantially underestimate semantic-null gold-alias collision rates, as reflected by low mean $\hat{\beta}$ and high ECE. Adding answer type, alias-set size, alias length, entity frequency, and neighbor-entity matching progressively improves calibration. Under full matching, the mean estimated contamination probability approaches the empirical gold-alias collision rate: $0.0825$ versus $0.0884$ for PoG-WebQSP, and $0.1242$ versus $0.1324$ for LMP-WebQSP.

Richer matching also improves discrimination. On PoG-WebQSP, AUROC increases from $0.512$ under random controls to $0.738$ under full matching. On LMP-WebQSP, AUROC increases from $0.519$ to $0.754$. This indicates that the full matched estimator is not only better calibrated in aggregate, but also better at ranking human-incorrect instances by their risk of gold-alias lexical collision.

## F Ablation and Sensitivity Analysis

This appendix provides additional analyses of ADE to examine whether its empirical behavior is driven by a single design choice or by a fragile hyperparameter setting. We focus on two questions: (i) whether the matching covariates used to construct negative-control alias sets contribute to human alignment, and (ii) whether the default reliability-scoring configuration is stable under reasonable hyperparameter variations. All analyses use the same human-annotated PoG-WebQSP and LMP-WebQSP sets as in Section 5.2. Unless otherwise stated, ADE uses the same estimator defined in Section 4: the contamination estimate $\hat{\beta}_i$ is computed by the empirical negative-control collision rate in Eq. equation 12, and the final reliability score is computed by the odds-based discount in Eq. equation 17. We set $\tau = 1$, $\epsilon = 10^{-8}$, and use a fixed threshold $\delta = 0.5$ when binarizing ADE scores for instance-level agreement analysis.

### F.1 Component Ablation

We first ablate the main components of ADE. The AliasMatch only variant corresponds to the high-recall alias-aware base evaluator before contamination correction. ADE w/o frequency matching removes entity-frequency matching from the negative-control construction. ADE w/o neighbor-entity matching removes the neighbor-entity matching component from the negative-control construction. Both variants still estimate contamination using the same empirical negative-

Table 8: Control-construction ablation on human-incorrect PoG-WebQSP instances. Richer matching monotonically improves semantic-null calibration and discrimination. The full matching strategy reduces ECE from $0.0770$ under random controls to $0.0070$, while increasing AUROC from $0.512$ to $0.738$.

| Match level | ECE | Spearman $\rho$ | AUROC | AUPRC | Mean $\hat{\beta}$ | Gold collision rate |
|---|---|---|---|---|---|---|
| Random | 0.0770 | 0.031 | 0.512 | 0.088 | 0.0062 | 0.0884 |
| Type only | 0.0535 | 0.126 | 0.563 | 0.115 | 0.0305 | 0.0884 |
| Type + length | 0.0287 | 0.241 | 0.641 | 0.168 | 0.0558 | 0.0884 |
| Type + length + frequency | 0.0146 | 0.318 | 0.704 | 0.223 | 0.0702 | 0.0884 |
| Full | 0.0070 | 0.361 | 0.738 | 0.265 | 0.0825 | 0.0884 |

Table 9: Control-construction ablation on human-incorrect LMP-WebQSP instances. Richer matching again improves both calibration and discrimination. The full matching strategy reduces ECE from $0.1120$ under random controls to $0.0101$, while increasing AUROC from $0.519$ to $0.754$.

| Match level | ECE | Spearman $\rho$ | AUROC | AUPRC | Mean $\hat{\beta}$ | Gold collision rate |
|---|---|---|---|---|---|---|
| Random | 0.1120 | 0.044 | 0.519 | 0.135 | 0.0185 | 0.1324 |
| Type only | 0.0745 | 0.139 | 0.576 | 0.175 | 0.0580 | 0.1324 |
| Type + length | 0.0460 | 0.257 | 0.653 | 0.240 | 0.0870 | 0.1324 |
| Type + length + frequency | 0.0235 | 0.342 | 0.721 | 0.320 | 0.1105 | 0.1324 |
| Full | 0.0101 | 0.389 | 0.754 | 0.385 | 0.1242 | 0.1324 |

control collision rate in Eq. equation 12; they differ only in how the negative-control alias sets are matched to the gold alias set. ADE W/O ODDS CORRECTION keeps the same negative-control estimate $\hat{\beta}_i$ but replaces the odds-based discount in Eq. equation 17 with a direct probability discount:

$$S^{(i)} = x_i^{AA}(1 - \hat{\beta}_i).$$

The full ADE model uses all matching covariates when constructing negative controls and applies the odds-based reliability correction.

Table 10 shows that the high-recall alias-aware evaluator substantially overestimates human correctness, especially on LMP-WebQSP. Adding ADE-style contamination correction reduces instance-level disagreement. Removing frequency matching or neighbor-entity matching weakens performance, suggesting that both covariates help construct negative-control alias sets whose collision behavior better reflects the false-positive contamination risk of the gold alias set. The full ADE variant achieves the lowest disagreement on both model-output sets, supporting the design choice of using both frequency and neighbor-entity information when estimating $\hat{\beta}_i$ through negative controls.

### F.2 HYPERPARAMETER SENSITIVITY

We next evaluate whether ADE is sensitive to its reliability-scoring hyperparameters. We vary one hyperparameter at a time while keeping the others fixed at their default values. Specifically, we consider $\tau \in \{0.25, 0.5, 1, 2, 4\}$ and $\delta \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$. Here, $\tau$ controls the strength of the odds-based reliability discount in Eq. equation 17, while $\delta$ is used only to binarize ADE scores for instance-level agreement analysis. The contamination estimate $\hat{\beta}_i$ itself is always computed by the empirical negative-control collision rate in Eq. equation 12.

Table 11 shows that ADE is not driven by a narrow hyperparameter choice. Across a broad range of $\tau$, the disagreement rate remains close to the default configuration, indicating that the odds-based correction is not overly sensitive to the discount strength. The binarization threshold $\delta$ has a larger effect because it directly changes instance-level decisions, but the default value $\delta = 0.5$ remains competitive on both sets. These results suggest that ADE's gains primarily come from the negative-control contamination estimate rather than fragile threshold tuning.

Table 10: Component ablation of ADE on human-annotated generative KGQA outputs. ALIASMATCH ONLY denotes the high-recall base evaluator before contamination correction. All ADE variants use the same empirical negative-control collision estimator in Eq. equation 12. Removing either frequency matching or neighbor-entity matching weakens human alignment, and the full ADE variant achieves the lowest instance-level disagreement on both sets.

| Evaluator | PoG-WebQSP | | | LMP-WebQSP | | |
|---|---|---|---|---|---|---|
| | Acc. | Δ Acc. | Δ Eval. | Acc. | Δ Acc. | Δ Eval. |
| ALIASMATCH ONLY | 0.8255 | +0.0323 | 5.31% | 0.9250 | +0.0799 | 8.31% |
| ADE W/O FREQUENCY MATCHING | 0.7968 | +0.0036 | 3.65% | 0.9079 | +0.0629 | 8.36% |
| ADE W/O NEIGHBOR-ENTITY MATCHING | 0.7858 | -0.0074 | 3.30% | 0.9018 | +0.0568 | 8.05% |
| ADE W/O ODDS CORRECTION | 0.7827 | -0.0105 | 3.18% | 0.9006 | +0.0556 | 8.08% |
| ADE FULL | 0.7791 | -0.0140 | **2.94%** | 0.8981 | +0.0531 | **7.92%** |

Table 11: Hyperparameter sensitivity of ADE. We report instance-level disagreement with human judgments under one-at-a-time hyperparameter variation. The default setting $(\tau, \delta) = (1, 0.5)$ remains near the best configuration across both human-annotated sets.

| Hyperparameter | Values | PoG-WebQSP Δ Eval. | LMP-WebQSP Δ Eval. |
|---|---|---|---|
| $\tau$ | 0.25 / 0.5 / 1 / 2 / 4 | 3.10 / 3.00 / **2.94** / 3.02 / 3.25 | 8.08 / 7.99 / **7.92** / 8.06 / 8.28 |
| $\delta$ | 0.3 / 0.4 / 0.5 / 0.6 / 0.7 | 3.60 / 3.20 / **2.94** / 3.05 / 3.70 | 8.85 / 8.22 / **7.92** / 8.10 / 8.71 |

# G LLM-AS-JUDGE CONFIGURATION AND AGREEMENT

This appendix provides the implementation details of the LLM-as-Judge baselines used in Section 5.5, together with additional pairwise agreement results among judge-based evaluators and ADE.

## G.1 LLM-AS-JUDGE CONFIGURATION

All judge models use the same prompt template and differ only in the model name. The judge is given the question, the gold answer list with aliases when available, and the model-generated response. It is not given retrieved evidence, reasoning paths, dataset identifiers, model identifiers, ADE features, or human labels. We follow the implementation description of LASM Zhang et al. (2025), a KGQA-oriented LLM-as-Judge method, when designing this judge baseline. Since LASM does not release its evaluation code, we are unable to perform a direct code-level comparison against its implementation.

**Prompt.** The system prompt is:

> You are an expert evaluator for Knowledge Graph Question Answering (KGQA). Your task is to judge whether a model's response correctly answers a question based on the provided gold answers.

For each instance, we use the following user prompt:

```
Question: {question}
Gold Answer(s): {gold_answers}
Model Response: {response}
Does the model response correctly answer the question? A
response is correct if it contains information that is
semantically equivalent to any one of the gold answers.
Accept paraphrases, alternative entity names, and partial
matches when they unambiguously refer to the correct answer.
Reject responses that merely mention the answer entity
incidentally without committing to it as the answer.
Reply with exactly one word: Yes or No.
Example: {example}
```

We include at most 20 gold answers or aliases in the prompt. When a system output contains multiple response fields, they are concatenated before judging. Responses longer than 3000 characters are truncated.

**Judge parameters.** We evaluate `gpt-3.5-turbo`, `gpt-4o-mini`, and `deepseek-v3`. All models are queried with temperature 0.0, maximum output length 5 tokens, and a single completion per instance. We do not set top-$p$, random seed, or stop sequences, and therefore use the provider defaults for these fields. Each request is retried up to 3 times with a retry delay of 10 seconds. Requests are issued sequentially without concurrency, with a request interval of 0.5 seconds.

**Output parsing.** The judge output is parsed with a deterministic rule. We first apply `strip()` and `lower()` to the returned text, and then map the output to correct if and only if it starts with `yes`:

$$\hat{y}_{\text{judge}} = \mathbb{I}\left[\text{lower}(\text{strip}(o)) \text{ starts with "yes"}\right],$$

where $o$ is the raw judge response. Thus, outputs such as `Yes`, `yes`, or `yes.` are mapped to correct, while `No`, empty outputs, malformed outputs, and failed requests after all retries are mapped to incorrect.

**Cost accounting.** For each LLM judge, API cost is estimated from the total number of input and output tokens:

$$\text{Cost} = \frac{T_{\text{in}}}{10^6} p_{\text{in}} + \frac{T_{\text{out}}}{10^6} p_{\text{out}},$$

where $T_{\text{in}}$ and $T_{\text{out}}$ denote the total input and output token counts, and $p_{\text{in}}$ and $p_{\text{out}}$ denote the corresponding prices per one million tokens. We use the following pricing configuration: `gpt-3.5-turbo` uses \$0.50 per 1M input tokens and \$1.50 per 1M output tokens; `gpt-4o-mini` uses \$0.15 per 1M input tokens and \$0.60 per 1M output tokens; and `deepseek-v3` uses \$0.27 per 1M input tokens and \$1.10 per 1M output tokens. The costs reported in Table 2 are normalized to cost per 1,000 evaluated instances. ADE requires no prompt tokens, completion tokens, or API calls, so its API cost is zero.

**Latency accounting.** Latency is measured as average wall-clock time per instance. For LLM judges, the timer covers API request time, response parsing, and retry time, but does not include prompt construction time. All LLM judge requests are executed sequentially, so the reported latency is not affected by parallelism. For ADE, latency is measured over the full `compute_score()` procedure, including alias-aware matching, negative-control matching, neighbor-entity feature computation, and reliability-score computation. We report per-instance latency to make the comparison independent of dataset size.

### G.2 Pairwise Agreement among Evaluators

To further examine the stability of judge-based evaluation, we report pairwise agreement among GPT-3.5-turbo, GPT-4o-mini, DeepSeek-V3, and ADE. For ADE, we binarize the soft reliability score using a threshold of 0.5. The results show that stronger judges such as GPT-4o-mini and DeepSeek-V3 are highly consistent with each other, while GPT-3.5-turbo exhibits substantially lower agreement, especially on LMP-WebQSP. ADE also shows high agreement with GPT-4o-mini and DeepSeek-V3, supporting its use as a deterministic judge-free alternative.

Table 12: Pairwise agreement among evaluators on PoG-WebQSP.

| | **GPT-3.5** | **GPT-4o-mini** | **DeepSeek-V3** | **ADE** |
|---|---|---|---|---|
| GPT-3.5-turbo | 100.00% | 82.49% | 85.85% | 81.15% |
| GPT-4o-mini | 82.49% | 100.00% | 90.42% | 89.38% |
| DeepSeek-V3 | 85.85% | 90.42% | 100.00% | 87.61% |
| **ADE (Ours)** | 81.15% | 89.38% | 87.61% | 100.00% |

Table 13: Pairwise agreement among evaluators on LMP-WebQSP.

| | **GPT-3.5** | **GPT-4o-mini** | **DeepSeek-V3** | **ADE** |
|---|---|---|---|---|
| GPT-3.5-turbo | 100.00% | 67.11% | 70.96% | 66.56% |
| GPT-4o-mini | 67.11% | 100.00% | 91.28% | 89.81% |
| DeepSeek-V3 | 70.96% | 91.28% | 100.00% | 88.41% |
| **ADE (Ours)** | 66.56% | 89.81% | 88.41% | 100.00% |

Table 14: Full human meta-evaluation results on WebQSP generative KGQA outputs.

| Evaluator Configurations | PoG-WebQSP | | | LMP-WebQSP | | |
|---|---|---|---|---|---|---|
| | Acc. | Δ Acc. | Δ Eval. | Acc. | Δ Acc. | Δ Eval. |
| *String Matching Evaluators* | | | | | | |
| w/ Align + w/ Norm + NS | 0.8255 | +0.0323 | 5.31% | 0.9250 | +0.0799 | 8.31% |
| w/ Align + w/ Norm + S | 0.7932 | +0.0000 | 5.98% | 0.8932 | +0.0482 | 9.14% |
| w/ Align + w/ Norm + MS | 0.6022 | -0.1910 | 20.20% | 0.7291 | -0.1159 | 21.48% |
| w/ Align + w/o Norm + NS | 0.8200 | +0.0268 | 5.37% | 0.9213 | +0.0763 | 8.19% |
| w/ Align + w/o Norm + S | 0.7877 | -0.0055 | 6.04% | 0.8902 | +0.0451 | 9.08% |
| w/ Align + w/o Norm + MS | 0.6004 | -0.1928 | 20.38% | 0.6931 | -0.1519 | 24.34% |
| w/o Align + w/ Norm + NS | 0.7916 | -0.0016 | 4.84% | 0.8964 | +0.0514 | 8.01% |
| w/o Align + w/ Norm + S | 0.7608 | -0.0323 | 5.55% | 0.8810 | +0.0360 | 9.02% |
| w/o Align + w/ Norm + MS | 0.5882 | -0.2050 | 20.74% | 0.7175 | -0.1275 | 22.39% |
| w/o Align + w/o Norm + NS | 0.7907 | -0.0024 | 5.90% | 0.9134 | +0.0683 | 8.01% |
| w/o Align + w/o Norm + S | 0.7596 | -0.0336 | 5.67% | 0.8816 | +0.0366 | 8.96% |
| w/o Align + w/o Norm + MS | 0.5857 | -0.2074 | 20.99% | 0.6815 | -0.1635 | 25.26% |
| *Advanced Evaluators* | | | | | | |
| LLM-as-Judge (GPT-4o-mini) | 0.8035 | +0.0103 | 3.20% | 0.8835 | +0.0384 | **6.70%** |
| ADE (Ours) | 0.7791 | -0.0140 | **2.94%** | 0.8981 | +0.0531 | 7.92% |
| Human (Oracle) | 0.7932 | – | – | 0.8450 | – | – |

## H Full Human Meta-Evaluation Results

This appendix reports the complete human meta-evaluation results omitted from Section 5.2 due to space constraints. For each model–dataset setting, we compare all string-matching evaluator configurations with GPT-4o-mini Judge and ADE. Acc. denotes the evaluator score, Δ Acc. is the signed gap from the human semantic oracle, and Δ Eval. denotes the instance-level disagreement rate with human judgments. Lower Δ Eval. indicates better alignment with human semantic correctness.

Overall, the complete results confirm the trend summarized in Table 1: string-matching evaluators exhibit substantial variation across alignment, normalization, and matching strictness, whereas ADE consistently reduces instance-level disagreement relative to the best post-hoc string-matching configuration. These results indicate that ADE's improvement is not simply due to choosing a different string-matching rule, but comes from explicitly correcting contamination-prone positive matches.

## I Additional Mechanism Analysis

This appendix provides additional mechanism analyses for the evaluator failures studied in the main text. We first examine how alias expansion changes the false-negative and false-positive profile of Non-Strict matching. We then analyze how verbose answer-like mentions are associated with false-positive lexical contamination and how ADE discounts such contamination-prone matches.

Table 15: Full human meta-evaluation results on CWQ generative KGQA outputs.

| Evaluator Configurations | PoG-CWQ | | | LMP-CWQ | | |
|---|---|---|---|---|---|---|
| | Acc. | Δ Acc. | Δ Eval. | Acc. | Δ Acc. | Δ Eval. |
| *String Matching Evaluators* | | | | | | |
| w/ Align + w/ Norm + NS | 0.6300 | +0.0280 | 7.58% | 0.7272 | +0.0672 | 10.05% |
| w/ Align + w/ Norm + S | 0.5759 | -0.0261 | 8.54% | 0.7094 | +0.0494 | 10.80% |
| w/ Align + w/ Norm + MS | 0.4429 | -0.1591 | 20.56% | 0.5741 | -0.0859 | 21.52% |
| w/ Align + w/o Norm + NS | 0.6327 | +0.0307 | 7.33% | 0.7272 | +0.0672 | 9.87% |
| w/ Align + w/o Norm + S | 0.5756 | -0.0264 | 8.43% | 0.7094 | +0.0494 | 10.84% |
| w/ Align + w/o Norm + MS | 0.4427 | -0.1593 | 20.87% | 0.5633 | -0.0967 | 23.53% |
| w/o Align + w/ Norm + NS | 0.6300 | +0.0280 | 7.20% | 0.7262 | +0.0662 | 9.47% |
| w/o Align + w/ Norm + S | 0.5759 | -0.0261 | 8.61% | 0.7094 | +0.0494 | 10.89% |
| w/o Align + w/ Norm + MS | 0.4429 | -0.1591 | 21.06% | 0.5741 | -0.0859 | 22.33% |
| w/o Align + w/o Norm + NS | 0.6297 | +0.0277 | 7.62% | 0.7272 | +0.0672 | 9.98% |
| w/o Align + w/o Norm + S | 0.5756 | -0.0264 | 8.51% | 0.7094 | +0.0494 | 10.71% |
| w/o Align + w/o Norm + MS | 0.4427 | -0.1593 | 21.25% | 0.5633 | -0.0967 | 24.09% |
| *Advanced Evaluators* | | | | | | |
| LLM-as-Judge (GPT-4o-mini) | 0.6123 | +0.0103 | 4.50% | 0.6984 | +0.0384 | 7.50% |
| ADE (Ours) | 0.5797 | -0.0223 | **4.40%** | 0.7272 | +0.0672 | **6.90%** |
| Human (Oracle) | 0.6020 | – | – | 0.6600 | – | – |

### I.1 Alias Expansion: False Negatives and False Positives

ADE applies a positive-only correction after alias-aware matching: evaluator-negative instances are assigned score zero, while evaluator-positive matches are discounted according to their estimated contamination risk. This design relies on a high-recall base matcher, but alias expansion itself may also increase the surface area for accidental lexical matches. We therefore analyze how alias expansion changes the confusion matrix of Non-Strict matching against human semantic judgments.

We compare Non-Strict matching with and without alias expansion on the WebQSP human meta-evaluation sets for PoG and LMP. In both cases, full normalization is used, and the human majority-vote label is treated as the semantic correctness reference. Table 16 reports the resulting true positives, false positives, true negatives, and false negatives.

Table 16: Effect of alias expansion on Non-Strict matching under human semantic judgments on WebQSP. Alias expansion reduces false negatives but also introduces additional false positives. NS Score denotes the evaluator positive rate, not instance-level agreement with human labels.

| Model | Group | w/o Alias | w/ Alias | Δ |
|---|---|---|---|---|
| PoG | TP | 1268 | 1283 | +15 |
| PoG | FP | 31 | 70 | +39 |
| PoG | TN | 308 | 269 | -39 |
| PoG | FN | 32 | 17 | -15 |
| PoG | NS Score | 0.7926 | 0.8255 | +0.0329 |
| LMP | TP | 1359 | 1366 | +7 |
| LMP | FP | 138 | 150 | +12 |
| LMP | TN | 116 | 104 | -12 |
| LMP | FN | 26 | 19 | -7 |
| LMP | NS Score | 0.9134 | 0.9250 | +0.0116 |

The results show that alias expansion indeed reduces false negatives. For PoG, the number of false negatives decreases from 32 to 17; for LMP, it decreases from 26 to 19. This confirms that alias-aware matching helps recover semantically correct responses that would otherwise be missed by surface-form mismatch.

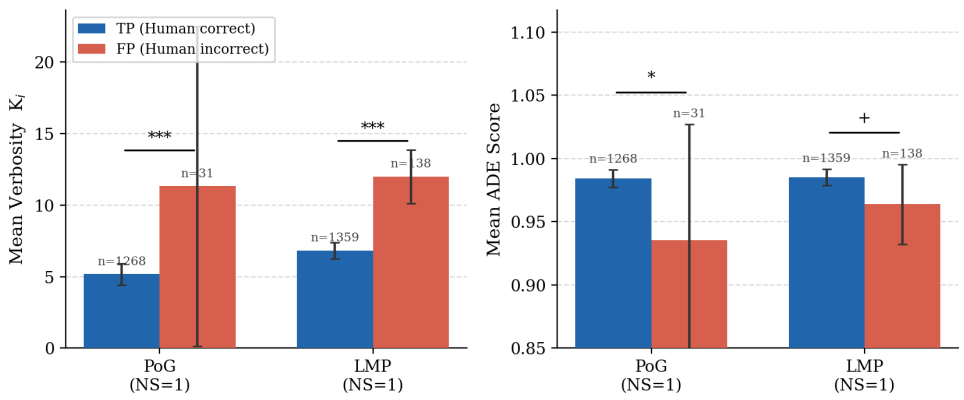



Figure 3: Mechanism analysis of verbosity-associated false positives under Non-Strict EM (NS). Left: NS false positives contain substantially more candidate answer-like mentions than NS true positives. Right: ADE assigns lower reliability scores to NS false positives than to NS true positives. Error bars denote 95% confidence intervals.

However, the same alias expansion also introduces additional false positives. For PoG, false positives increase from 31 to 70, which is larger than the false-negative reduction. For LMP, false positives increase from 138 to 150 while false negatives decrease by only 7. As a result, alias expansion raises the reported Non-Strict score from 0.7926 to 0.8255 for PoG and from 0.9134 to 0.9250 for LMP, even though the human semantic correctness rates remain fixed at 0.7932 and 0.8450, respectively.

This analysis illustrates the trade-off faced by high-recall string matching in generative KGQA. Alias expansion mitigates false-negative mismatch, but it also enlarges the lexical matching surface and makes verbose incorrect responses more likely to collide with a gold alias. Thus, alias-aware matching alone is insufficient as a semantic evaluator. These results motivate ADE's two-stage design: first use alias-aware matching to reduce missed correct answers, and then apply a positive-match contamination correction to discount alias-induced false positives.

### I.2 Verbosity-Associated False Positives

We provide additional diagnostics for the mechanism behind false-positive contamination in permissive string matching. Specifically, we analyze whether false positives of Non-Strict EM (NS) are associated with verbose answer-like mentions, and whether ADE assigns lower reliability to such contamination-prone matches. Using human annotations as ground truth, we partition all 1,639 WebQSP instances for each model into four groups: TP $(\mathrm{NS}=1,\mathrm{Human}=1)$, FP $(\mathrm{NS}=1,\mathrm{Human}=0)$, TN $(\mathrm{NS}=0,\mathrm{Human}=0)$, and FN $(\mathrm{NS}=0,\mathrm{Human}=1)$.

Figure 3 focuses on the NS-positive subset, where false-positive contamination can occur. The left panel shows that NS false positives are substantially more verbose than true positives. For PoG, the average number of candidate answer-like mentions increases from $K_i = 5.17$ for TP instances to $K_i = 11.32$ for FP instances, with a significant difference ($p = 7.9 \times 10^{-5}$). For LMP, the same pattern is stronger: FP instances have $K_i = 11.99$, compared with $K_i = 6.81$ for TP instances ($p = 1.0 \times 10^{-18}$). Thus, across both systems, NS false positives contain roughly twice as many answer-like mentions as true positives.

The right panel of Figure 3 shows that ADE assigns lower reliability to NS false positives than to NS true positives. For PoG, the mean ADE score decreases from $0.984$ on TP instances to $0.935$ on FP instances ($p = 0.038$). For LMP, the same direction holds, decreasing from $0.985$ to $0.964$ ($p = 0.060$). Although the per-instance correction is modest, it is consistent across both systems and accumulates into the aggregate correction effect observed in the main human-alignment results.

Figure 4 shows the full distribution of candidate answer-like mentions $K_i$ across TP, FP, TN, and FN groups. For PoG, TP instances are concentrated at relatively low $K_i$, whereas FP instances exhibit a heavier tail and a larger median. This indicates that accidental NS matches are often associated with responses containing unusually many answer-like mentions. For LMP, TN and FN instances

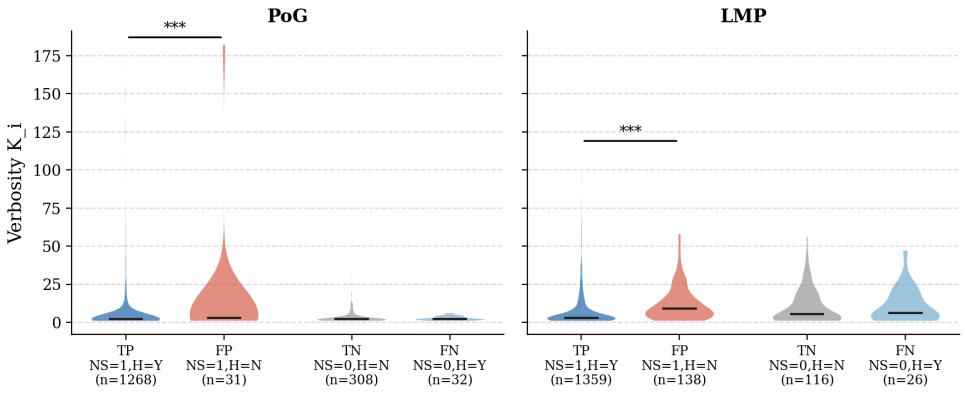



Figure 4: Distribution of candidate answer-like mentions $K_i$ across TP, FP, TN, and FN groups under Non-Strict EM. FP instances exhibit heavier-tailed mention distributions than TP instances, especially within the NS-positive subset.

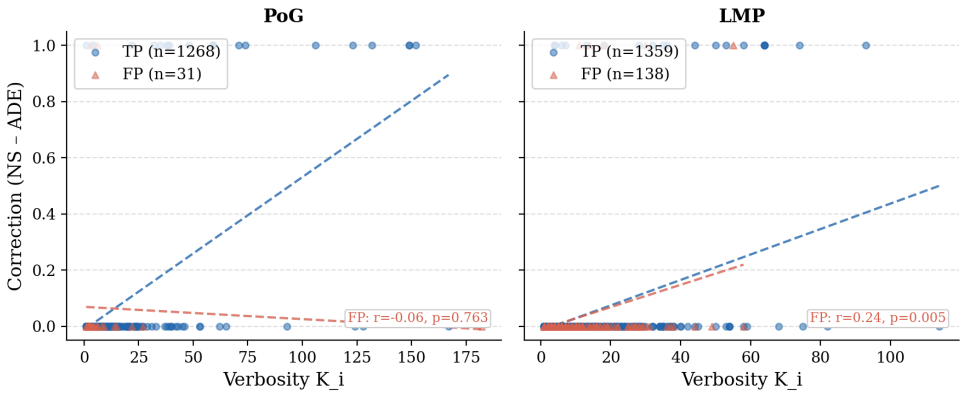



Figure 5: Relationship between candidate answer-like mentions $K_i$ and ADE correction magnitude, defined as $\mathrm{NS} - S_{\mathrm{ADE}}$, for NS-positive instances. For LMP false positives, correction magnitude is positively correlated with $K_i$, suggesting that ADE assigns larger discounts to some responses with many answer-like mentions.

are also relatively verbose, suggesting that LMP generally produces longer responses. Nevertheless, within the NS-positive subset, FP instances remain more verbose than TP instances, consistent with the main mechanism analysis.

Figure 5 further plots the ADE correction magnitude, defined as the difference between the binary NS decision and the soft ADE score, $\mathrm{NS} - S_{\mathrm{ADE}}$, against $K_i$ for NS-positive instances. For PoG, the FP correlation is not significant ($r = -0.06$, $p = 0.763$), likely due to the small number of FP instances. For LMP, the FP correction is positively correlated with $K_i$ ($r = 0.24$, $p = 0.005$), indicating that false-positive instances with more answer-like mentions tend to receive larger ADE discounts.

These results should not be interpreted as showing that ADE directly penalizes response length. Rather, they support a lexical-collision interpretation: responses with more answer-like mentions have more opportunities to collide with gold aliases or matched negative-control aliases under permissive string matching. ADE operationalizes this intuition through the empirical matched negative-control collision rate in Eq. equation 12, rather than through a parametric length penalty. Thus, the correction depends on whether the response also matches counterfactual negative-control alias sets, not on verbosity alone.

We also observe a small number of extreme TP cases with large correction values. These cases correspond to highly verbose NS-positive responses that collide with multiple matched negative-

Table 17: Existing assets used in this paper.

| Asset | Source | Version | License / terms |
|---|---|---|---|
| WebQuestionsSP (WebQSP) | RoG-WebQSP | rmanluo/RoG-webqsp | Not specified |
| Complex WebQuestions (CWQ) | RoG-CWQ | rmanluo/RoG-cwq | Not specified |
| Freebase | Google Freebase | Freebase Data Dump | CC BY |
| PoG outputs / code | official PoG repository | main branch | Apache License 2.0 |
| LMP outputs / code | official LMP repository | main branch | Not specified |
| GCR outputs / code | official GCR repository | main branch | MIT License |
| RJE outputs / code | official RJE repository | master branch | Not specified |
| OpenAI API | API data controls | `gpt-3.5-turbo`, `gpt-4o-mini` | OpenAI Services Agreement |
| DeepSeek API | Open Platform Terms | DeepSeek-V3 | DeepSeek Open Platform Terms |

*Note.* For assets whose official license could not be identified, we mark the license as "Not specified" and use the asset only for research evaluation without redistribution.

control aliases, causing ADE to assign low reliability despite the human label being correct. Such cases illustrate a limitation of conservative contamination correction: when a semantically correct response contains many answer-like entities, ADE may over-discount it. However, as shown in the main human-alignment results, these occasional over-discounts are outweighed by reductions in contamination-prone false positives in the evaluated settings.

## J Limitations

This work focuses on debiasing lexical evaluation rather than replacing semantic evaluation. ADE can discount contamination-prone matches after a response has already matched a gold alias, but it cannot recover correct answers that are missed by the alias inventory or by the base matcher. Its effectiveness also depends on the quality of the matched negative controls: when gold answers are systematically more likely to be mentioned than matched controls because of retrieval artifacts, question context, or model priors, the estimated contamination risk may be imperfect.

This paper focuses on generative KGQA and does not systematically study whether similar evaluation inconsistencies and systematic biases also arise in other tasks that rely on string-based matching, such as open-domain question answering. We also do not evaluate whether ADE is effective in these broader settings. Nevertheless, the theoretical framework developed in this paper for string-based evaluator bias is not specific to KGQA, and can potentially be used to analyze false-negative mismatch and false-positive lexical contamination in other reference-based evaluation tasks. Extending ADE to these broader settings is an important direction for future work.

## K Broader Impacts

This work aims to improve the reliability and reproducibility of generative KGQA evaluation. By identifying and correcting false-positive lexical contamination, ADE may help researchers make more trustworthy comparisons between KGQA systems and reduce dependence on costly or non-deterministic LLM-as-Judge evaluators. However, automatic evaluators can also create risks if their scores are over-interpreted as definitive measures of semantic correctness. In particular, ADE may still mis-score responses when the alias inventory is incomplete or when correct verbose answers collide with negative-control aliases. Therefore, ADE should be used as a diagnostic tool and a lightweight complement to human evaluation or semantic judges, rather than as the sole basis for high-stakes leaderboard decisions.

## L Licenses and Terms of Use for Existing Assets

We use several existing assets, including public KGQA benchmarks, released model prediction files, and external model APIs for judge-based evaluation. Table 17 summarizes the source, version or access date, license or terms of use, and how each asset is used in this paper. We use these assets only for research evaluation and do not redistribute third-party raw data or model outputs unless permitted by the corresponding license or terms.