QueryEval: A mechanistic framework to evaluate query-aware generations

Anonymous authors
Paper under double-blind review

Abstract

Query-centric tasks such as query-aware summarization and attribute-specific question answering demand evaluation metrics that assess whether a generated response genuinely addresses the input query. Existing metrics such as ROUGE and BERTScore measure lexical and semantic similarity against ground-truth references, but systematically fail to capture query relevance and require costly human annotation to construct references. We propose **QueryEval**, a reference-free evaluation metric that scores query-aware generation quality by analyzing how attention weights and activations corresponding to query tokens evolve across transformer layers during generation, without requiring any ground-truth labels. This internal-representation perspective allows **QueryEval** to directly measure the degree to which a model attends to and integrates query information throughout the generation process, making it both interpretable and scalable. We evaluate **QueryEval** across three benchmarks **(QTSumm, MS MARCO, and SQuAD)** and three model architectures, demonstrating that it correlates strongly with LLM judgments and outperforms both reference-based and reference-free baselines by a consistent margin. **QueryEval** requires no supervised annotation, making it practical for large-scale evaluation pipelines where human references are unavailable.

1 Introduction

Language models are increasingly deployed in query-aware settings, retrieval-augmented generation, question answering, query-focused summarisation, and table-to-text generation where response quality is inseparable from how faithfully the model addresses the input query. Yet a model can generate fluent, factually plausible text while having internally bypassed the query entirely, and this failure is systematically invisible to every evaluation metric in current use.

The field has produced a rich landscape of automatic evaluation tools: n-gram overlap measures (Papineni et al., 2002; Lin, 2004), embedding-based similarity metrics (Zhang et al., 2019; Zhao et al., 2019), and LLM-based judges (Liu et al., 2023; Fu et al., 2024). Each represents a genuine advance, yet all share the same fundamental limitation: they evaluate the *output* of generation without examining the *process*. They cannot determine whether the model's internal computation was driven by the query or whether the query was effectively ignored. As models grow more capable, this blind spot becomes more consequential, a sufficiently powerful model can produce reference-matching outputs through parametric memorization or distributional priors, with no mechanistic connection to the query, and score well on every existing metric.

We argue that closing this gap requires inspecting the *internal computational process* through which the model conditions its generation on the query. Recent advances in mechanistic interpretability have produced exactly the tools needed for this: techniques for identifying internal circuits, tracing activation flow, and decomposing representations into interpretable feature directions (Elhage et al., 2021; Meng et al., 2022; Olsson et al., 2022). Developed as instruments for model *understanding*, we repurpose them here as instruments for model *evaluation*.

We introduce **QueryEval**, the first evaluation framework for query-aware text generation that operates on model internals rather than model outputs, comprising three mechanistically grounded subcomponents:

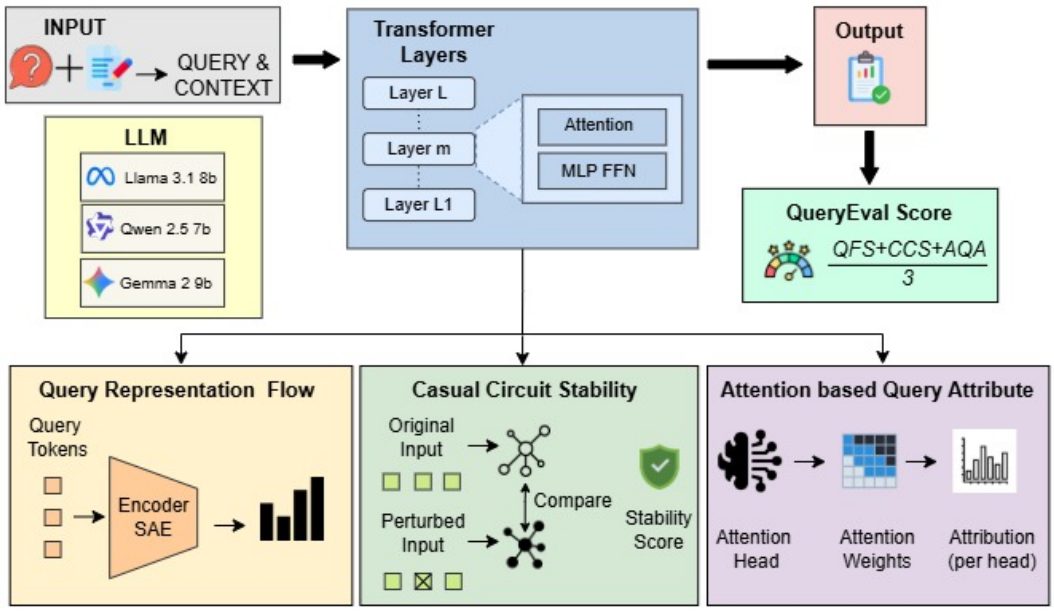


Figure 1: Overview of QueryEval metric

Query Representation Fidelity (QRF) measures layer-wise alignment between query and generation feature representations in the SAE transcoder space; **Circuit Coherence Score (CCS)** quantifies the structural stability of causal query-routing circuits across examples under perturbation; and **Attention Query Alignment (AQA)** attributes query grounding to individual attention heads via normalised logit recovery. We evaluate **QueryEval** on three model families **(LLaMA-3.1-8B, Qwen-2.5-7B, Gemma-2-9B)** and three benchmarks **(SQuAD, MS MARCO, QTSumm)**, demonstrating that it captures failure modes invisible to output-level metrics. An overview of our proposed metric is shown in Figure. 1

Our Contributions. To address this gap, our work introduces the following contributions:

- **QueryEval**: the first generation evaluation framework grounded in mechanistic interpretability, operating on model internals rather than outputs.
- **Query Representation Fidelity (QRF)**: a layer-wise sub-metric tracking query–generation feature alignment in SAE transcoder space before and after each transformer layer.
- **Circuit Consistency Score (CCS)**: an activation patching sub-metric measuring structural reproducibility of query-routing circuits across examples.
- **Attention Query Alignment (AQA)**: a head-level sub-metric quantifying the causal contribution of individual attention heads to query-conditioned answer prediction.
- **Empirical analysis** across three model families and three query-aware benchmarks, revealing internal failure modes undetectable by existing metrics.

2 Related Work

2.1 Evaluation Metrics for Query-Aware Generation

Automatic evaluation of query-aware generation has traditionally relied on reference-based lexical, semantic, or LLM-based metrics. While these approaches effectively assess output quality, they operate exclusively on

Table 1: **Comparison of representative evaluation metrics for query-aware generation. RF**: Reference-Free; **QA**: Explicitly evaluates query relevance; **IS**: Uses the generating model's internal representations or activations during evaluation. Existing metrics evaluate generated outputs, whereas QueryEval evaluates internal query grounding through mechanistic analysis.

Metric	Principle	RF	QA	IS	Limitation
ROUGE (Lin, 2004)	Lexical n-gram overlap.	No	No	No	Measures only lexical similarity and cannot determine whether the response is genuinely query grounded.
BERTScore (Zhang et al., 2019)	Contextual embedding similarity.	No	No	No	Evaluates semantic similarity but ignores how the query is internally processed.
UniEval (Li et al., 2025)	LLM-based multi-dimensional evaluation.	Yes	Partial	No	Relies on an external evaluator operating only on generated text.
G-Eval (Liu et al., 2023)	GPT-based LLM judge.	Yes	Partial	No	Provides holistic quality scores but offers no insight into the model's internal reasoning.
QuestEval (Scialom et al., 2021)	Question-answering based evaluation.	Yes	Yes	No	Measures behavioural query coverage without verifying causal dependence on the query.
QueryEval	Mechanistic evaluation using representations, circuits and attention.	Yes	Yes	Yes	Measures internal query grounding through mechanistic analysis rather than output similarity.

the generated text and provide little evidence of whether the model internally grounded its generation in the input query. Table 1 summarizes the representative evaluation paradigms, their underlying principles, and their key limitations.

Existing evaluation metrics differ substantially in their scoring strategies, ranging from lexical overlap to semantic similarity and LLM-based judgments. Nevertheless, they all evaluate generation as a black-box process by assessing only the produced text. None directly measures whether query information is encoded, propagated, and causally utilized throughout the model's internal computation, motivating the mechanistic evaluation framework proposed in this work.

2.2 Mechanistic Interpretability

Our work builds upon three complementary directions in mechanistic interpretability.

Residual representations and sparse features. Transformer Circuits (Elhage et al., 2021) demonstrated that transformer computation can be understood through additive operations over the residual stream. Subsequent work on Sparse Autoencoders (SAEs) and transcoders (Bricken et al., 2023; Elhage et al., 2022) showed that these residual representations can be decomposed into sparse, approximately monosemantic feature directions, enabling fine-grained analysis of the semantic information encoded at each layer.

Causal tracing and activation patching. Activation patching and causal tracing (Vig et al., 2020; Meng et al., 2022) identify components responsible for specific model behaviours by restoring clean activations into corrupted forward passes and measuring the resulting recovery. These techniques have become standard tools for identifying causal circuits within transformer models.

Attention head specialisation. Previous studies have demonstrated that individual attention heads often implement specialised computational functions, including induction, positional reasoning, copying, and factual retrieval (Olsson et al., 2022; Voita et al., 2019; Clark et al., 2019). These findings suggest that attention heads provide meaningful units for analysing information routing inside large language models. Collectively, these advances establish that internal representations, causal circuits, and specialised attention heads provide complementary views of transformer computation. Building upon these foundations, our work

Field	QTSumm	MS Marco	Squad
Input	List of best-selling singles Artist | Single | Year | Sales (M) Bing Crosby | White Christma | 1942 | 50 Elton John | Candle in the | 1997 | 33	Over half of the supply of palladium and platinum goes into catalytic converters, converting up to 90% of harmful exhaust gases into less-harmful substances. Palladium is a soft silver-white metal resembling platinum...	Beyonce Giselle Knowles was born in Houston, Texas, to Celestine Ann "Tina" Knowles, a hairdresser and salon owner, and Mathew Knowles, a Xerox sales manager. Beyonce's name is a tribute to her mother's maiden name. ...
Query	What difference in sales is observable between the two singles released by Bing Crosby?	What is palladium metal?	Where did Beyonce get her name from?
Output	Bing Crosby has two hits, "White Christmas" and "Silent Night," differing by 20 million copies sold (50M vs 30M).	A soft silver-white metal that resembles platinum with the lowest density and melting point of the platinum group metals.	Her mother's maiden name.

Figure 2: Sample illustration of utilized data

uses these mechanistic signals as quantitative indicators of query grounding, thereby extending mechanistic interpretability from an explanatory tool to a framework for evaluation.

3 Data

To generalize the workings of our proposed mechanistic interpreted metric **QueryEval**, we carried out extensive experiments on three publicly available benchmark datasets, namely **QTSumm, MS MARCO, SQuaD2.0**. The details of the data samples used for our purpose and other details of our dataset are elaborated in the further subsections. An illustrative example of the dataset samples is given in Figure 5.

QTSumm Dataset: Proposed by (Zhao et al., 2023), QTSUMM comprises 7,111 human-annotated query-summary pairs derived from 2,934 tables spanning diverse topical domains. Each instance consists of a table, a natural language query, and a reference summary that synthesizes query-relevant information from the table. For our experiments, we used a sample of 4,982 instances from the full dataset. The train-test split for this dataset is 80:20.

MS MARCO Dataset: Proposed by (Bajaj et al., 2016) MS-MARCO (Microsoft Machine Reading Comprehension) is a large-scale question-answering and machine reading comprehension dataset containing over 1 million real-world queries sampled from anonymized Bing search logs. Each sample comprises a user query, a set of retrieved web passages, and human-generated answers. The dataset is widely used for evaluating query understanding, information retrieval, and query-aware text generation systems due to its scale and realistic search scenarios. Due to computation cost and space constraints, we have utilized 24698 data points from the dataset.The train-test split for this dataset is 94:6.

SQuAD Dataset : Proposed by Rajpurkar et al. (2016), SQUAD (The Stanford Question Answering Dataset) is a large-scale machine reading comprehension dataset comprising over 100,000 question-answer pairs derived from 500+ Wikipedia articles. Each sample consists of a context passage, a natural language question, and a corresponding answer span extracted from the passage. Owing to its emphasis on context understanding and precise answer extraction, SQuAD serves as a standard benchmark for evaluating question-answering and query-aware generation systems. For effective computation, we utilized 17364 samples in our experiments.The train-test split for this dataset is 92:8.

Together, these datasets span a range of query-aware generation scenarios, from table-grounded summarization to document-level question answering, providing a sufficiently diverse testbed to evaluate the generalization of **QueryEval** across task types and domains.

4 *QueryEval* Framework

QueryEval decomposes query-aware generation quality into four mechanistic subcomponents, each targeting a distinct aspect of how transformer models internally process and integrate query information during generation. In this section, we describe each subcomponent in detail and motivate its inclusion through its causal role in query-grounded generation.

4.1 Query Representation Fidelity (QRF)

4.1.1 Overview

Query Representation Fidelity (QRF) is a mechanistic sub component metric that measures how well a given transformer layer preserves the alignment between the query and the generated text in the internal feature space of the model. Rather than evaluating this alignment at the output token level, QRF operates on the *sparse feature activations* produced by a Sparse Autoencoder (SAE) applied to the model's residual stream — capturing a semantically grounded, interpretable signal of how much query-relevant information survives into the generation-side computation.

The central idea is straightforward: for a given layer l, we measure alignment *before* the layer processes its input and *after* it produces its output, both times in SAE feature space. The difference between these two alignment scores reveals whether the layer strengthens, preserves, or disrupts the correspondence between the query and the generation.

4.1.2 Notation and Setup

Let the input token sequence be partitioned into a query prefix of N_q tokens and a generation suffix of N_g tokens:

$$\mathbf{x} \;=\; \underbrace{(x_1, \ldots, x_{N_q})}_{\text{query tokens}} \;\Big\|\; \underbrace{(x_{N_q+1}, \ldots, x_{N_q+N_g})}_{\text{generation tokens}}. \tag{1}$$

For a transformer of depth L, let $\mathbf{h}_t^{(l)} \in \mathbb{R}^d$ denote the residual-stream activation at layer l and token position t. A pre-trained SAE is applied to these activations to obtain sparse feature vectors. Concretely, for an SAE with encoder weight $\mathbf{W}_{\text{enc}} \in \mathbb{R}^{M \times d}$ and bias $\mathbf{b}_{\text{enc}} \in \mathbb{R}^M$, the feature activation at position t and layer l is:

$$\mathbf{f}_t^{(l)} \;=\; \text{ReLU}\Big(\mathbf{W}_{\text{enc}}\, \mathbf{h}_t^{(l)} + \mathbf{b}_{\text{enc}}\Big) \;\in\; \mathbb{R}^M_{\geq 0}, \tag{2}$$

where $M \gg d$ is the number of SAE features.

4.1.3 Pre-Layer and Post-Layer Alignment

For each layer l, we obtain SAE feature activations at two distinct checkpoints in the residual stream:

- $\mathcal{X}^{(l)}$: activations computed from the residual stream *before* entering layer l — i.e. the input to layer l.
- $\mathcal{Y}^{(l)}$: activations computed from the residual stream *after* exiting layer l — i.e. the output of layer l.

At each checkpoint $\mathcal{Z} \in \{\mathcal{X}^{(l)}, \mathcal{Y}^{(l)}\}$, we separately aggregate the SAE feature activations over the query token positions and the generation token positions to obtain two *mean feature vectors*:

$$\bar{\mathbf{f}}_q^{\mathcal{Z}} \;=\; \frac{1}{N_q} \sum_{t=1}^{N_q} \mathbf{f}_t^{\mathcal{Z}} \;\in\; \mathbb{R}^M, \tag{3}$$

$$\bar{\mathbf{f}}_g^{\mathcal{Z}} \;=\; \frac{1}{N_g} \sum_{t=N_q+1}^{N_q+N_g} \mathbf{f}_t^{\mathcal{Z}} \;\in\; \mathbb{R}^M. \tag{4}$$

Here $\bar{\mathbf{f}}_q^{\mathcal{Z}}$ is the *query feature centroid* — the average latent feature signature of the query tokens at checkpoint $\mathcal{Z}$ — and $\bar{\mathbf{f}}_g^{\mathcal{Z}}$ is the *generation feature centroid* — the average latent feature signature of the generation tokens at the same checkpoint.

The alignment between the query and the generated text at checkpoint $\mathcal{Z}$ is then measured as the cosine similarity between these two centroids:

$$\mathrm{QRF}^{\mathcal{Z}} \;=\; \cos\!\big(\bar{\mathbf{f}}_q^{\mathcal{Z}},\, \bar{\mathbf{f}}_g^{\mathcal{Z}}\big) \;=\; \frac{\bar{\mathbf{f}}_q^{\mathcal{Z}} \cdot \bar{\mathbf{f}}_g^{\mathcal{Z}}}{\big\|\bar{\mathbf{f}}_q^{\mathcal{Z}}\big\|_2 \big\|\bar{\mathbf{f}}_g^{\mathcal{Z}}\big\|_2} \;\in\; [-1,\, 1]. \tag{5}$$

A score near $+1$ indicates that the query and generation tokens activate highly overlapping sets of SAE features at checkpoint $\mathcal{Z}$, signifying strong internal alignment. A score near 0 indicates orthogonality — the feature directions evoked by the query and the generation are statistically unrelated at that point in the network.

Applying Equation 5 at both checkpoints for layer l yields two scalar scores:

$$\mathrm{QRF}_{\mathcal{X}}^{(l)} \;=\; \cos\!\Big(\bar{\mathbf{f}}_q^{\mathcal{X}^{(l)}},\, \bar{\mathbf{f}}_g^{\mathcal{X}^{(l)}}\Big), \qquad \mathrm{QRF}_{\mathcal{Y}}^{(l)} \;=\; \cos\!\Big(\bar{\mathbf{f}}_q^{\mathcal{Y}^{(l)}},\, \bar{\mathbf{f}}_g^{\mathcal{Y}^{(l)}}\Big). \tag{6}$$

4.1.4 Layer Importance via QRF Difference

Having obtained the pre-layer and post-layer alignment scores, we define the *QRF difference* for layer l as:

$$\Delta\mathrm{QRF}^{(l)} \;=\; \mathrm{QRF}_{\mathcal{Y}}^{(l)} \;-\; \mathrm{QRF}_{\mathcal{X}}^{(l)}. \tag{7}$$

This quantity directly measures the *causal contribution* of layer l to query–generation alignment: it asks how much the alignment changed as a result of the computations performed by that specific layer. The three cases carry distinct mechanistic interpretations:

- $\Delta\mathrm{QRF}^{(l)} > 0$: Layer l **increases** alignment. The attention heads and feed-forward network within this layer actively draw the generation-side representations closer to the query-side representations in feature space. Such layers are mechanistically important for query grounding.
- $\Delta\mathrm{QRF}^{(l)} \approx 0$: Layer l is **neutral**. It neither strengthens nor disrupts the query–generation correspondence; its computation is largely orthogonal to the grounding task.
- $\Delta\mathrm{QRF}^{(l)} < 0$: Layer l **decreases** alignment. The layer's transformations push query and generation representations apart in feature space, indicating that it introduces information — from parametric memory or contextual re-weighting — that is not aligned with the query signal. Under noisy query conditions, this is a key locus of failure.

Layer importance score. We define the *layer importance* of layer l with respect to query grounding as the absolute QRF difference:

$$\mathrm{LI}^{(l)} \;=\; \Big|\,\Delta\mathrm{QRF}^{(l)}\,\Big| \;=\; \Big|\mathrm{QRF}_{\mathcal{Y}}^{(l)} - \mathrm{QRF}_{\mathcal{X}}^{(l)}\Big|. \tag{8}$$

Layers with high $\mathrm{LI}^{(l)}$ — regardless of the direction of change — are those whose internal computations most strongly modulate the query–generation correspondence, making them primary candidates for further mechanistic investigation via activation patching or circuit analysis.

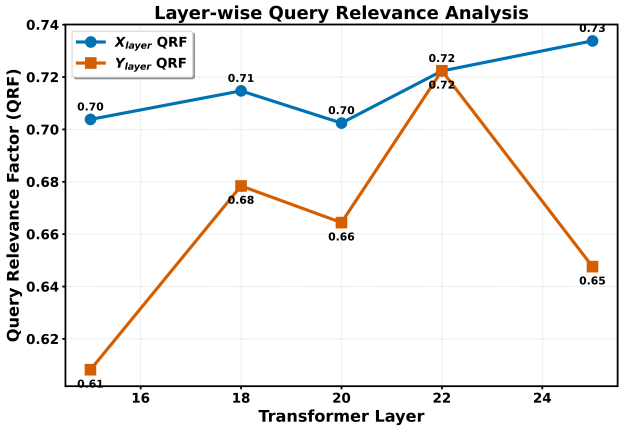


Figure 3: QRF scores across different layers of a model.

4.1.5 Aggregate QRF Score

To obtain a single scalar summarising query grounding across the full network, we average the post-layer QRF scores over all L layers:

$$\mathrm{QRF} \;=\; \frac{1}{L}\sum_{l=1}^{L}\mathrm{QRF}_{\mathcal{Y}}^{(l)}, \tag{9}$$

and define the aggregate layer importance as:

$$\overline{\mathrm{LI}} \;=\; \frac{1}{L}\sum_{l=1}^{L}\mathrm{LI}^{(l)} \;=\; \frac{1}{L}\sum_{l=1}^{L}\left|\mathrm{QRF}_{\mathcal{Y}}^{(l)} - \mathrm{QRF}_{\mathcal{X}}^{(l)}\right|. \tag{10}$$

The full *QRF depth profile*, $\left(\Delta\mathrm{QRF}^{(1)}, \ldots, \Delta\mathrm{QRF}^{(L)}\right)$, is retained for per-layer analysis and visualisation, as it reveals the progression of query grounding across the network rather than collapsing it to a single value. A sample illustration of QRF scores across different layers in LLAMA 3.1 8B model is shown in Figure 4

4.2 Circuit Consistency Score (CCS)

4.2.1 Motivation

A central question in the mechanistic evaluation of query-aware generation is not only *whether* a model attends to the query, but *how consistently* it does so across different inputs. A model that happens to produce correct outputs on individual examples may do so through unstable or input-idiosyncratic internal circuits, circuits that fire for superficial lexical reasons rather than systematic query grounding. To quantify this stability, we introduce the **Circuit Consistency Score (CCS)**, a metric that measures the structural reproducibility of the causal subgraph responsible for query-to-generation information flow across pairs of independent examples.

CCS is grounded in *activation patching* (Meng et al., 2022): the targeted restoration of clean activations into a corrupted forward pass to identify which token positions and layers causally mediate the model's output. By identifying and comparing these *circuits* — the minimal sets of token positions per layer whose activations causally determine the answer — across example pairs, CCS measures whether the model routes query-conditioned information through the same internal pathway regardless of surface form variation.

4.2.2 Experimental Setup

Clean and corrupted prompts. For each example i in the evaluation set, we construct two versions of the input prompt:

- **Clean prompt** $\mathbf{p}^{(i)}_{\text{clean}}$: the full prompt containing the unperturbed query $q^{(i)}$, the supporting context (document, table, or passage), and the ground-truth answer.
- **Corrupted prompt** $\mathbf{p}^{(i)}_{\text{corrupt}}$: an otherwise identical prompt in which a fraction ρ of query tokens are randomly replaced with tokens drawn uniformly from the model's vocabulary, yielding a corrupted query $\tilde{q}^{(i)}$. Formally, for query token ids $(u_1, \ldots, u_{N_q})$, the corruption selects a random subset $\mathcal{S} \subset \{1, \ldots, N_q\}$ with $|\mathcal{S}| = \lceil \rho N_q \rceil$ and replaces each u_j, $j \in \mathcal{S}$ with $\tilde{u}_j \sim \text{Uniform}(\{0, \ldots, V-1\})$, where V is the vocabulary size.

In our experiments $\rho = 0.40$, meaning 40% of query tokens are corrupted. This level of perturbation is sufficient to degrade the logit of the target answer token while preserving the overall prompt structure, creating a clear signal for patching to recover.

Target signal. The metric of interest in each patching experiment is the logit of the first token of the ground-truth answer at the final sequence position:

$$\ell(\mathbf{p}) \;=\; \left[\mathbf{W}_U\, \mathbf{h}^{(L)}_T\right]_{a_1}, \tag{11}$$

where $\mathbf{h}^{(L)}_T \in \mathbb{R}^d$ is the final-layer residual stream at the last token position, $\mathbf{W}_U$ is the unembedding matrix, and a_1 is the token id of the first answer token.

4.2.3 Activation Patching and Circuit Identification

Patching procedure. For each example i, we first run the model on the clean prompt and cache the MLP input activations at each monitored layer $l \in \mathcal{L} = \{l_1, \ldots, l_K\}$:

$$\mathbf{A}^{(l,i)}_{\text{clean}} \;\in\; \mathbb{R}^{T_{\text{clean}} \times d}, \quad l \in \mathcal{L}, \tag{12}$$

where T_{clean} is the sequence length of the clean prompt and d is the model's hidden dimension. We then run the model on the corrupted prompt and record the baseline logit:

$$\ell^{(i)}_{\text{corrupt}} \;=\; \ell\Big(\mathbf{p}^{(i)}_{\text{corrupt}}\Big). \tag{13}$$

For each layer $l \in \mathcal{L}$ and each query token position $t \in \mathcal{Q}$, where $\mathcal{Q} = \{t_{\text{start}}, \ldots, t_{\text{end}}\}$ denotes the query token span within the corrupted prompt, we perform a *single-position patch*: the MLP input activation at position t in the corrupted forward pass is replaced by the corresponding clean activation, and the patched logit is recorded:

$$\ell^{(i)}_{l,t} \;=\; \ell\Big(\mathbf{p}^{(i)}_{\text{corrupt}} \;\big|\; \mathbf{h}^{(l)}_t \leftarrow \mathbf{A}^{(l,i)}_{\text{clean}}[t,:]\Big). \tag{14}$$

Logit recovery score. The causal importance of position t at layer l for example i is measured as the *logit recovery* — how much the patched logit exceeds the corrupted baseline:

$$s^{(i)}_{l,t} \;=\; \ell^{(i)}_{l,t} \;-\; \ell^{(i)}_{\text{corrupt}}. \tag{15}$$

A large positive $s^{(i)}_{l,t}$ indicates that restoring the clean activation at position t and layer l substantially recovers the model's original answer prediction — i.e. that position is a *causal node* in the query-grounding circuit.

Circuit extraction. For each layer l, we rank all query-span positions by their logit recovery score and retain the top κ fraction as the *circuit* for that layer:

$$\mathcal{C}^{(i)}_l \;=\; \text{TopK}_\kappa\Big(\big\{s^{(i)}_{l,t}\big\}_{t \in \mathcal{Q}}\Big), \tag{16}$$

where $\text{TopK}_\kappa(\cdot)$ returns the set of positions corresponding to the top $\kappa \cdot |\mathcal{Q}|$ scores. In our experiments we use $\kappa = 0.50$, retaining the top 50% of query positions per layer. The full circuit for example i is the collection across all monitored layers:

$$\mathcal{C}^{(i)} \;=\; \Big\{\, \mathcal{C}^{(i)}_l \,\Big\}_{l \in \mathcal{L}}. \tag{17}$$

4.2.4 Circuit Consistency Score

Pairwise circuit overlap. Given circuits $\mathcal{C}_l^{(i)}$ and $\mathcal{C}_l^{(j)}$ extracted from two independent examples $i \neq j$, we measure their structural overlap at layer l using the *Jaccard similarity*:

$$\mathrm{J}\Big(\mathcal{C}_l^{(i)}, \mathcal{C}_l^{(j)}\Big) \;=\; \frac{\Big|\mathcal{C}_l^{(i)} \;\cap\; \mathcal{C}_l^{(j)}\Big|}{\Big|\mathcal{C}_l^{(i)} \;\cup\; \mathcal{C}_l^{(j)}\Big|}, \tag{18}$$

with the convention that $\mathrm{J}(\emptyset, \emptyset) = 1$. Jaccard similarity is chosen because it is symmetric, normalised in $[-1, 1]$, and sensitive to both false positives and false negatives in circuit membership, unlike simpler overlap ratios.

Layer-level CCS. The CCS at layer l is defined as the mean pairwise Jaccard similarity over all $\binom{N}{2}$ example pairs in the evaluation set of size N:

$$\mathrm{CCS}_l \;=\; \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \mathrm{J}\Big(\mathcal{C}_l^{(i)}, \mathcal{C}_l^{(j)}\Big). \tag{19}$$

Aggregate CCS. The aggregate CCS summarises circuit consistency across all monitored layers:

$$\mathrm{CCS} \;=\; \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \mathrm{CCS}_l. \tag{20}$$

Interpretation. A high CCS_l at layer l indicates that the model consistently routes query information through the same token positions at that layer, regardless of surface-form variation across examples — a hallmark of systematic, generalised query grounding. A low CCS_l indicates that the causal circuit is highly example-specific, suggesting the model may be exploiting idiosyncratic lexical cues rather than a stable internal mechanism.

4.2.5 Implementation Details

Activation patching is implemented via forward pre-hooks registered on the MLP sub-layer of each monitored transformer layer. The pre-hook intercepts the MLP's input tensor, replaces the activation at the designated position with the cached clean value, and passes the modified tensor forward — leaving all other computations, including attention, unchanged. This ensures that the patch is *minimally invasive*: it isolates the causal role of a single position at a single layer without perturbing the rest of the residual stream.

Query token boundaries are identified automatically by locating the `Query:` delimiter token sequence within the tokenised prompt. When the delimiter is not found — for instance in table-structured prompts — the middle third of the token sequence is used as a conservative fallback span. All experiments are conducted with the model in `eval()` mode and under `torch.no_grad()`, ensuring that patching measurements are not confounded by gradient-tracking overhead or dropout.

The metric is **model-agnostic**: it requires only that the target architecture expose its transformer layers as indexable modules with accessible MLP sub-layers, a structural property satisfied by all decoder-only transformer families. Monitored layers $\mathcal{L}$, the corruption fraction ρ, the top-κ threshold, and the evaluation set size N are the only hyperparameters, and sensitivity analyses for each are reported in Appendix A.

4.3 Attention Query Alignment (AQA)

4.3.1 Motivation

While Circuit Consistency Score (Section 4.2) identifies *which token positions* causally mediate query grounding, it does not distinguish *which attention heads* are responsible for routing that information. Transformer

attention layers are composed of multiple parallel heads, each potentially specializing in a distinct sub-computation — copying, induction, positional tracking, or semantic matching (Olsson et al., 2022; Elhage et al., 2021). A position-level circuit may therefore obscure substantial heterogeneity: Some heads at a causally important layer may carry query information faithfully, while others contribute negligibly or even interfere with it.

To resolve this, we introduce **Attention Query Alignment (AQA)**, a head-level activation patching metric that quantifies the causal contribution of each individual attention head, at each query token position, to the model's ability to recover its answer prediction under query corruption. Unlike CCS, which identifies *where* (token positions) the circuit lives, AQA identifies *which sub-modules* (attention heads) implement that circuit, providing a finer-grained mechanistic decomposition of the query grounding.

4.3.2 Setup: Clean and Corrupted Runs

The experimental setup mirrors that of CCS (Section 4.2.2): for each example i, a clean prompt $\mathbf{p}^{(i)}_{\text{clean}}$ containing the original query is paired with a corrupted prompt $\mathbf{p}^{(i)}_{\text{corrupt}}$ in which a fraction $\rho = 0.40$ of query tokens are replaced with tokens drawn uniformly at random from the vocabulary. Three reference logits are recorded for the target answer token (Equation 11):

$$\ell^{(i)}_{\text{clean}} \;=\; \ell\Big(\mathbf{p}^{(i)}_{\text{clean}}\Big), \qquad \ell^{(i)}_{\text{null}} \;=\; \ell^{(i)}_{\text{corrupt}} \;=\; \ell\Big(\mathbf{p}^{(i)}_{\text{corrupt}}\Big). \tag{21}$$

The clean logit $\ell^{(i)}_{\text{clean}}$ represents the upper bound of achievable recovery, while the corrupted (null) logit $\ell^{(i)}_{\text{null}}$ represents the baseline in the absence of any patching.

4.3.3 Per-Head Activation Extraction

Unlike CCS, which patches the MLP input activation as a single d-dimensional vector per position, AQA patches at a finer granularity: the per-head output of the attention mechanism, immediately before the output projection $\mathbf{W}_O$.

For a layer l with n_h attention heads and head dimension $d_h = d/n_h$, the concatenated attention output $\mathbf{x}^{(l)}_t \in \mathbb{R}^d$ at position t — the input to the output projection — is reshaped into per-head components:

$$\mathbf{x}^{(l)}_t \;=\; \Big[\mathbf{x}^{(l,1)}_t,\, \mathbf{x}^{(l,2)}_t,\, \ldots,\, \mathbf{x}^{(l,n_h)}_t\Big], \qquad \mathbf{x}^{(l,h)}_t \in \mathbb{R}^{d_h}, \tag{22}$$

where $\mathbf{x}^{(l,h)}_t$ denotes the output of head h at layer l and position t, prior to the final linear mixing performed by $\mathbf{W}_O$. A forward pre-hook on the output-projection module intercepts this tensor, caches it for the clean run, and — during patching — selectively overwrites a single head's slice at a single position with its clean-run counterpart before allowing the corrupted forward pass to continue.

4.3.4 Per-Head Patching and Attribution Score

For each layer $l \in \mathcal{L}$, head $h \in \{1, \ldots, n_h\}$, and query token position $t \in \mathcal{Q}$, we perform a targeted patch: the head-h slice of the corrupted run's attention output at position t is replaced with the corresponding clean-run value $\mathbf{x}^{(l,h)}_{t,\text{clean}}$, and the resulting patched logit is recorded:

$$\ell^{(i)}_{l,h,t} \;=\; \ell\Big(\mathbf{p}^{(i)}_{\text{corrupt}} \,\Big|\, \mathbf{x}^{(l,h)}_t \leftarrow \mathbf{x}^{(l,h)}_{t,\text{clean}}\Big). \tag{23}$$

The *normalised attribution score* for head h at layer l and position t is defined as the fraction of the clean–corrupt logit gap recovered by this single-head patch:

$$\text{AQA}^{(i)}_{l,h,t} \;=\; \frac{\ell^{(i)}_{l,h,t} - \ell^{(i)}_{\text{null}}}{\ell^{(i)}_{\text{clean}} - \ell^{(i)}_{\text{null}} + \varepsilon}, \tag{24}$$

where $\varepsilon = 10^{-8}$ is a small constant added for numerical stability when the clean–null gap is near zero. This normalisation places the score on an interpretable scale: a value of 1 indicates the single-head patch fully recovers the clean prediction, a value of 0 indicates no recovery (equivalent to the corrupted baseline), and values may exceed 1 or fall below 0 when a head's contribution is super-additive or actively counterproductive, respectively.

Head-level attribution. The position-averaged attribution score for head h at layer l on example i is obtained by averaging over all query positions:

$$\mathrm{AQA}_{l,h}^{(i)} \;=\; \frac{1}{|\mathcal{Q}|}\sum_{t\in\mathcal{Q}} \mathrm{AQA}_{l,h,t}^{(i)}. \tag{25}$$

4.3.5 Head Circuit Extraction

Analogous to the position-level circuit extraction in CCS, we identify the *head circuit* at layer l for example i as the top-κ fraction of heads ranked by attribution score:

$$\mathcal{H}_l^{(i)} \;=\; \mathrm{TopK}_\kappa\Big(\big\{\mathrm{AQA}_{l,h}^{(i)}\big\}_{h=1}^{n_h}\Big), \tag{26}$$

with $\kappa = 0.50$ in our experiments, retaining the top half of heads by attribution magnitude at each layer. This head-level circuit can itself be compared across examples using the Jaccard similarity defined in Equation 18, yielding a head-level analogue of CCS that characterises whether the model consistently routes query information through the *same heads* across different inputs, complementing the position-level analysis.

4.3.6 Aggregate AQA Score

The example-level AQA at layer l is the mean attribution across all heads:

$$\mathrm{AQA}_l^{(i)} \;=\; \frac{1}{n_h}\sum_{h=1}^{n_h} \mathrm{AQA}_{l,h}^{(i)}. \tag{27}$$

The layer-level AQA, averaged over the full evaluation set of N examples, is:

$$\mathrm{AQA}_l \;=\; \frac{1}{N}\sum_{i=1}^{N} \mathrm{AQA}_l^{(i)}. \tag{28}$$

The overall AQA score, aggregated across all monitored layers, is:

$$\mathrm{AQA} \;=\; \frac{1}{|\mathcal{L}|}\sum_{l\in\mathcal{L}} \mathrm{AQA}_l. \tag{29}$$

Interpretation. A high AQA_l indicates that, on average, individual attention heads at layer l carry substantial causal weight for query-conditioned answer recovery — the layer's contribution to query grounding is *distributed and individually potent* across its heads. A low AQA_l, even at a layer with high CCS or high $\Delta\mathrm{QRF}^{(l)}$ (Section 4.1), suggests that the layer's contribution to query grounding emerges from the *joint*, non-linear combination of many heads rather than from any single head's isolated effect — a signature of distributed rather than localised computation.

4.3.7 Relationship to CCS

AQA and CCS are complementary decompositions of the same underlying patching framework, differing only in the granularity of the intervention:

- **CCS** patches the MLP input activation — the full d-dimensional residual stream vector at a given position — attributing causal importance to *token positions* irrespective of which attention heads produced that information.
- **AQA** patches individual d_h-dimensional head outputs prior to $\mathbf{W}_O$, attributing causal importance to *specific heads* at a given position, decomposing the position-level signal that CCS treats as atomic.

Used together, the two metrics allow a layer to be characterised along two independent axes: *where* in the sequence query information is causally load-bearing (CCS), and *which architectural sub-components* implement that causal pathway (AQA).

4.3.8 Implementation Details

Per-head interception is implemented via a forward pre-hook registered on the `self_attn.o_proj` module of each monitored layer. The hook receives the concatenated attention output of shape (B, S, d), reshapes it into (B, S, n_h, d_h) to expose individual heads, optionally overwrites a single $(head, position)$ slice with a cached clean-run value, and reshapes the result back to (B, S, d) before returning it to continue the forward pass. This ensures that only the targeted head's contribution at the targeted position is altered; the remaining heads, all other positions, and all subsequent layers are computed exactly as in the unpatched corrupted run.

As with CCS, the metric is **model-agnostic**, requiring only that the target architecture expose its attention output projection as an addressable module and report a fixed number of attention heads per layer — properties satisfied by all standard multi-head and grouped-query attention transformer implementations. The monitored layer set $\mathcal{L}$, corruption fraction ρ, and top-κ threshold are shared with CCS, ensuring that the two metrics are computed under identical experimental conditions and are directly comparable.

4.4 QueryEval: Composite Score

The three subcomponents introduced above **QRF, CCS, and AQA** each measure a structurally distinct aspect of query grounding: feature-space alignment across the residual stream, causal circuit reproducibility at the token-position level, and head-level attribution of query-sensitive attention, respectively. Since these dimensions are complementary and non-redundant, we treat each subcomponent as an equally weighted contributor to the final metric. The **QueryEval** score is defined as their arithmetic mean:

$$\text{QueryEval} \;=\; \frac{1}{3}\,(\text{QRF} \;+\; \text{CCS} \;+\; \text{AQA})\,, \tag{30}$$

where each subcomponent is an aggregate scalar in $[-1, 1]$ for QRF and AQA, and in $[0, 1]$ for CCS. The arithmetic mean preserves the interpretability of the individual components while providing a single scalar that summarises the overall quality of query grounding across all three internal dimensions.

QueryEval does not operate against a fixed threshold. Instead, scores closer to 1 indicate that the model consistently maintains strong alignment between query and generation representations (QRF), routes information through stable and reproducible circuits (CCS), and assigns meaningful causal weight to query tokens across its attention heads (AQA) collectively reflecting a model whose internal computation is genuinely grounded in the input query. Scores closer to 0, or negative, indicate that one or more of these properties breaks down, signalling a model that may produce plausible outputs without faithfully processing the query internally.

5 Model Instantiations

To assess the generalizability of **QueryEval** across architectural families, we experiment on three open-weight decoder-only large language models: **LLaMA 3.1 8B (Grattafiori et al., 2024), Qwen 2.5 7B (Qwen et al., 2025), and Gemma 2 9B (Team et al., 2024)**. These models span distinct training corpora, tokenization strategies, and architectural choices, including differences in attention mechanisms,

normalization schemes, and hidden dimensionalities, making them a representative testbed for evaluating the cross-architecture robustness of the metric.

Layer selection. Each subcomponent of **QueryEval** requires caching internal activations at a set of monitored layers $\mathcal{L}$. Running the full suite of experiments at every layer of each model is computationally prohibitive due to the activation patching and transcoder encoding operations involved. We therefore select a fixed set of five layers per model according to a *relative depth criterion*: layers are chosen to correspond approximately to the same fractional depths across all three models, ensuring that early, mid, and late processing stages are consistently represented regardless of total model depth. Concretely, we target relative depths of approximately 47%, 55%, 63%, 70%, and 78% of each model's total layer count, capturing the transition from initial contextual encoding through intermediate semantic composition to the final generation-proximal layers.

Table 2 reports the selected layers for each model, together with their total layer counts and the corresponding exact depth percentages.

Table 2: Selected monitoring layers per model. Depth % is computed as $\lfloor l/L \rfloor \times 100$, where l is the layer index and L is the total number of layers.

Model	**Total Layers** (L)	**Hidden Dim.** (d)	**Selected Layers**	**Depth (%)**
LLaMA 3.1 8B	32	4096	15, 18, 20, 22, 25	46.9, 56.2, 62.5, 68.8, 78.1
Qwen 2.5 7B	28	3584	13, 15, 18, 20, 22	46.4, 53.6, 64.3, 71.4, 78.6
Gemma 2 9B	42	3584	19, 23, 26, 29, 33	45.2, 54.8, 61.9, 69.0, 78.6

This depth-relative selection strategy ensures that comparisons across models are grounded in the same *functional stage* of the forward pass rather than in raw layer indices, which are not comparable across architectures with different depths. The chosen range, which spans roughly the second half of each network deliberately excludes the earliest layers, where representations are dominated by low-level token statistics, and focuses on the depth range in which prior work has shown that factual recall, semantic composition, and output-predictive computation are most active (Meng et al., 2022; Geva et al., 2021).

6 Experimental Setup

All experiments were conducted on a single NVIDIA RTX A5000 GPU with 24 GB of VRAM, paired with a 64-core CPU host and 128 GB of system RAM. Model weights and intermediate activations were stored in `bfloat16` precision during forward passes to reduce GPU memory pressure, with hidden-state caches offloaded to CPU in `float16` between pipeline phases. Final metric computations were performed in `float32` on CPU to preserve numerical precision. All code was implemented in Python 3.10. The core libraries used across the pipeline are listed in Table 3.

Table 4 lists all hyperparameters shared across the three metrics.

7 Results and Discussion

7.1 Query Representation Fidelity (QRF)

QRF measures the degree to which each layer in the model's transformer stack actively contributes to aligning the query with the generation. Concretely, for each monitored layer we compute the cosine similarity between the query and generation feature centroids in the SAE transcoder space both before the layer processes its input and after it produces its output. The difference between these two alignment scores, $\Delta\text{QRF}^{(l)}$, isolates the net effect of that specific layer on query–generation correspondence.

A **positive** $\Delta\text{QRF}^{(l)}$ indicates that the layer actively strengthens the correspondence between the query and the generation in feature space — its attention heads and feed-forward sub-layers draw the generation-side representations closer to the query-side representations, marking it as a meaningful contributor to

Table 3: Software libraries used in the experimental pipeline.

Library	Version	Role
`torch`	2.2+	Tensor computation, GPU execution, forward hooks
`transformers`	4.40+	Model and tokenizer loading, hidden-state extraction
`transcoder`	–	SAE/transcoder encoding for QRF computation
`numpy`	1.26+	Numerical aggregation, Jaccard and AQA score computation
`pandas`	2.1+	Dataset loading, per-row and summary CSV management
`tqdm`	4.66+	Progress tracking across row-level patching loops
`matplotlib`	3.8+	Layer-wise metric profile visualisation
`seaborn`	0.13+	Heatmap and distribution plotting for cross-layer results
`collections`	stdlib	`OrderedDict` management for SAE weight loading

Table 4: Hyperparameters used across all metrics.

Hyperparameter	Symbol	Value
Query corruption fraction	ρ	0.40
Top-κ circuit threshold	κ	0.50
Top-K features	K	10
Transcoder expansion factor	e	32
Sequence truncation length	$T_{\max}$	128
Max prompt token length	–	512
Query shuffle seed	–	42
Evaluation set size	N	200
Numerical stability constant	ε	10^{-8}

query-grounded generation. A **negative** $\Delta\mathrm{QRF}^{(l)}$, by contrast, indicates that the layer moves the query and generation representations further apart — its internal computation introduces information that is not aligned with the query signal, effectively suppressing rather than propagating query-relevant features into the generation process. Layers with $\Delta\mathrm{QRF}^{(l)} \approx 0$ are neutral: their computations are largely orthogonal to the query-grounding task and contribute neither positively nor negatively to the final generation.

Across all three models and all three datasets, we observe that $\Delta\mathrm{QRF}^{(l)}$ is not uniformly distributed across the layer stack. A small subset of layers consistently produce large positive values, suggesting that query grounding is a sparse, localised phenomenon rather than a diffuse property of the entire network. The majority of layers register values near zero or mildly negative, indicating that most of the transformer's depth is occupied with computations that are largely query-agnostic a finding consistent with prior mechanistic interpretability work identifying sparse functional circuits within dense networks (Elhage et al., 2021; Wang et al., 2022).

Notably, the distribution of $\Delta\mathrm{QRF}^{(l)}$ values varies across datasets. On SQuAD and MS MARCO, where queries are concise and factoid in nature, positive contributions tend to concentrate in the middle and later layers of the network, consistent with the hypothesis that earlier layers handle syntactic and positional encoding while semantic query–context integration occurs at greater depth. On QTSumm, where queries

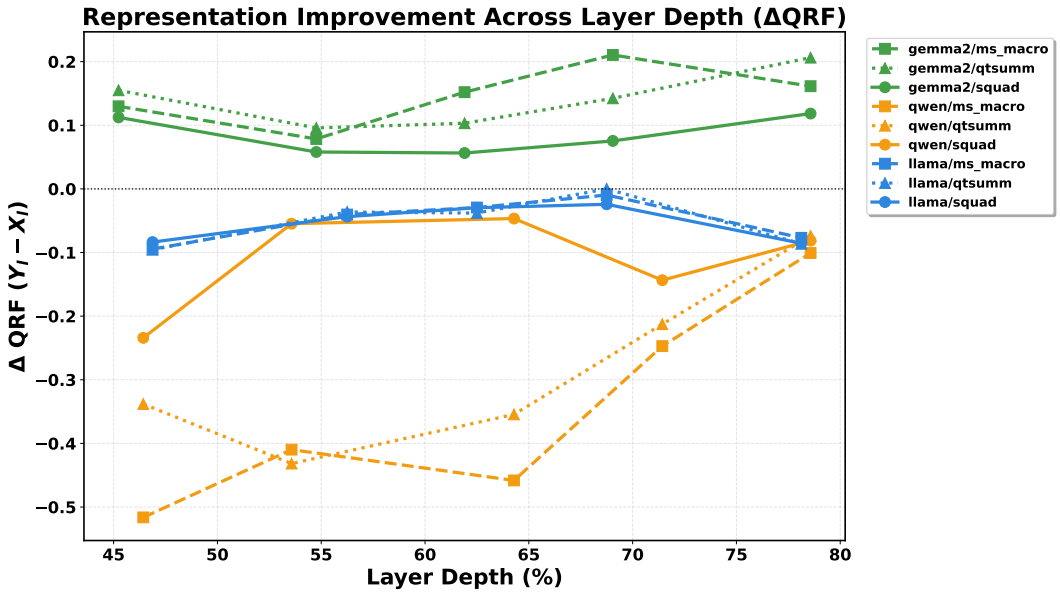


Figure 4: Overview of QRF changes across all models.

target structured tabular content, the contribution profile is more diffuse, with positive $\Delta\text{QRF}^{(l)}$ values distributed across a wider range of layers, reflecting the additional structural parsing demands imposed by the table format.

Differences across model families are also apparent. LLaMA-3.1 exhibits sharper, more peaked $\Delta\text{QRF}^{(l)}$ profiles, with a small number of layers accounting for the bulk of positive contribution. Qwen-2.5 and Gemma-2 display comparatively flatter profiles, suggesting that query-grounding computation is more evenly distributed across their respective layer stacks. These architectural differences have implications for targeted fine-tuning and mechanistic intervention: a model with a peaked profile offers clearer surgical targets for improving query grounding, whereas a flat profile may require broader intervention. An overview of $\Delta\text{QRF}^{(l)}$ for all datasets across the three models is shown in Figure.4 The complete per-layer $\Delta\text{QRF}^{(l)}$ values for all model and dataset combinations are reported in Appendix A.

7.2 Circuit Consistency Score (CCS)

CCS quantifies the structural reproducibility of the causal circuits responsible for query-to-generation information routing, measured as the mean pairwise Jaccard similarity of top-κ token position sets across example pairs at each monitored layer. Full per-layer results are provided in Appendix A. Across all three models and datasets, **CCS** values are uniformly high, with the majority of layer, dataset combinations exceeding 0.90, indicating that the model consistently routes query-relevant information through the same positional circuit regardless of input variation. A clear **depth trend** is observed in all models: **CCS** increases monotonically with layer index, with shallow layers exhibiting greater circuit variability and deeper layers converging on a stable, shared routing mechanism. This pattern is most pronounced in Gemma-2 on SQuAD, where CCS rises from 0.752 at layer 19 to 0.911 at layer 33, and least pronounced in **LLaMA-3.1** on **MS MARCO**, where values remain stable between 0.939 and 0.952 across all monitored layers.

Dataset complexity has a consistent moderating effect. CCS on **QTSumm** is substantially lower than on **SQuAD** and **MS MARCO** across all models and all layers, with Gemma-2 recording the minimum observed value of 0.625 at layer 19. This suppression is attributable to the structural heterogeneity of table-formatted inputs, where query–context relationships are non-local and do not resolve to consistent positional anchors across examples. By contrast, the factoid, passage-aligned queries in SQuAD and MS MARCO admit stable positional circuits that the model reliably reuses.

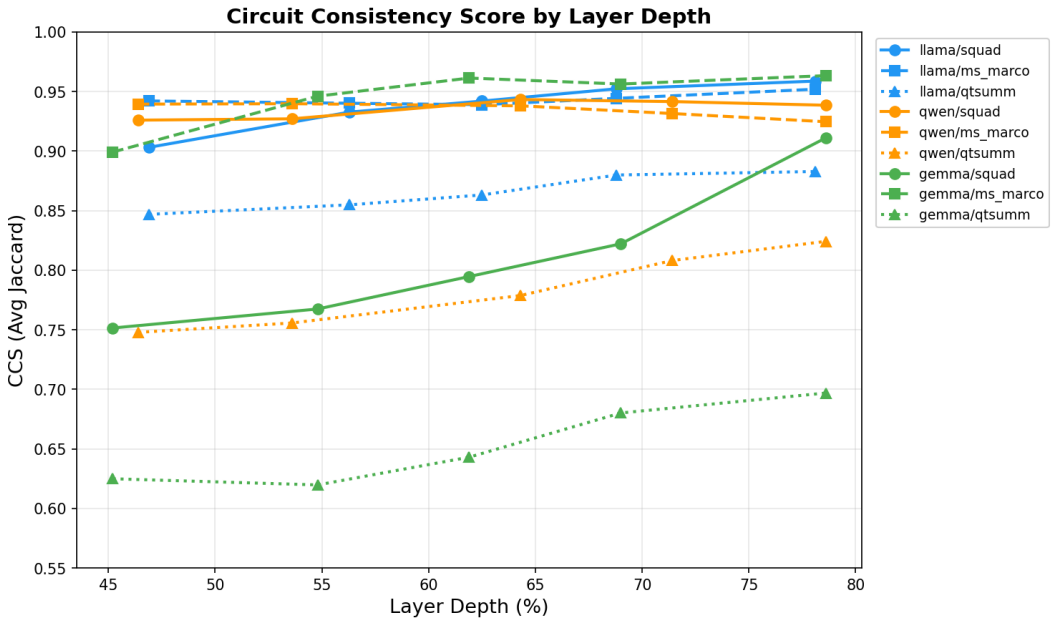


Figure 5: Overview of CCS scores across all models

At the model level, LLaMA-3.1 achieves the highest and most consistent CCS across all datasets, while Gemma-2 exhibits the greatest intra-model variance — peaking at 0.963 on MS MARCO yet falling to 0.697 on QTSumm at its deepest monitored layer. Qwen-2.5 occupies an intermediate position with moderate consistency across datasets. Taken together, these results confirm that query grounding circuits are structurally stable for well-formed, text-based queries, and that this stability degrades predictably as input structure diverges from the canonical passage–question format.

7.3 Attention Query Alignment(AQA)

Figure 6 summarizes the normalized layer-wise AQA trends across all model–dataset pairs, while Tables 12–14 report the corresponding absolute values. Overall, AQA indicates that query-relevant attention is primarily concentrated in the deeper transformer layers, although the attribution patterns vary substantially across models and datasets. Llama 3.1 exhibits relatively stable attribution throughout the selected layers, with modest peaks at Layer 20 on MS MARCO and Layer 22 on QTSumm. In contrast, Qwen 2.5 shows consistently high attribution across all analyzed layers for QTSumm, suggesting sustained query-aware attention during later computations.

Gemma 2 demonstrates a more localized attribution pattern. While SQuAD produces negligible AQA values, MS MARCO exhibits distinct peaks at Layers 19 (0.1917) and 29 (0.1667), indicating that query-relevant attention is concentrated at specific computation stages rather than being uniformly distributed. Together with QRF and CCS, these observations show that different LLMs employ distinct mechanisms for preserving and attending to query information: Llama maintains relatively uniform attribution, Qwen distributes attribution consistently across deeper layers, whereas Gemma relies on sparse high-attribution layers.

7.4 Comparison with Existing Evaluation Metrics

Table 5 compares **QueryEval** with conventional automatic evaluation metrics, including ROUGE and BERTScore, as well as a reference-free LLM-as-judge baseline using GPT-5 as the judge model. Unlike output-based metrics, **QueryEval** evaluates the extent to which the model internally preserves and attends to query-specific information through the proposed QRF, CCS, and AQA components. Since **QueryE-**

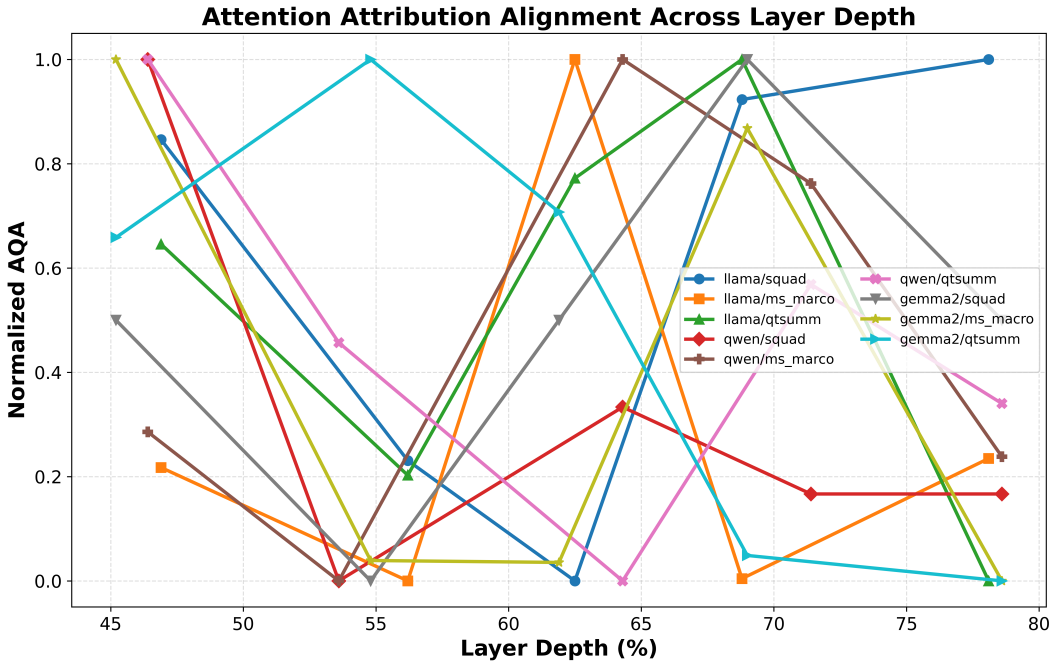


Figure 6: Overview of AQA scores across all models

val produces layer-wise scores, we averaged the scores across all layers for each dataset. Consequently, **QueryEval** does not necessarily follow the ranking induced by lexical or semantic overlap metrics, nor by LLM-judge scores. For example, although Qwen achieves the highest ROUGE score on SQuAD (0.6805) as well as the highest LLM-Judge score (0.9005), Gemma obtains the highest **QueryEval** score (0.5833), indicating stronger internal query grounding despite lower output similarity and lower judged quality. Similarly, on MS MARCO, Gemma achieves the highest **QueryEval** score (0.6544), whereas Llama attains the highest BERTScore (0.8619) and Qwen attains the highest LLM-Judge score (0.8130). Notably, on QTSumm, LLM-Judge scores collapse to near zero across all models (0.0420–0.0530), while **QueryEval** scores remain substantially higher and stable (0.4857–0.5480), suggesting that reference-free LLM judgment can be unreliable or overly punitive on long-form summarization tasks, whereas QueryEval continues to capture meaningful internal query grounding regardless of surface-level output quality. These observations suggest that **QueryEval** captures complementary aspects of model behavior beyond output similarity and LLM-judged quality, providing an internal, representation-based assessment of query faithfulness that is not reflected by existing automatic evaluation metrics. The scatter plots of Pearson correlation between **QueryEval** and the existing metrics are provided in Appendix A.

8 Conclusion

In this work, we introduced **QueryEval**, a mechanistic evaluation framework for query-aware text generation in large language models. Unlike existing metrics that assess generation quality solely at the output level—comparing token sequences or embedding similarities against reference texts, or relying on costly, closed-source LLM-as-judge pipelines—**QueryEval** is entirely *reference-free* and evaluates the *internal computational process* through which a model grounds its generation in the input query. This makes **QueryEval** the first evaluation framework to leverage mechanistic interpretability as a principled basis for generation quality assessment, offering a lightweight, reproducible, and inference-time-only alternative to expensive LLM-judge evaluation, which requires repeated API calls to large proprietary models, incurs non-trivial monetary and latency costs, and, as our experiments show, can produce unstable or overly punitive judgments on tasks such as long-form summarization.

Table 5: Comparison of QueryEval with conventional automatic evaluation metrics. All scores are averaged over the evaluation set.

Model	Dataset	R-1	R-2	R-L	BERT-F1	LLM-Judge	QueryEval
Llama	SQuAD	0.2654	0.1730	0.2645	0.8668	0.8980	0.5506
Llama	MS MARCO	0.2415	0.1708	0.2238	0.8619	0.7295	0.5424
Llama	QTSumm	0.3337	0.1428	0.2591	0.8607	0.0530	0.5131
Qwen	SQuAD	0.6805	0.4469	0.6800	0.7900	0.9005	0.5226
Qwen	MS MARCO	0.3746	0.2625	0.3546	0.6030	0.8130	0.5221
Qwen	QTSumm	0.3558	0.1742	0.2839	0.5836	0.0435	0.4857
Gemma	SQuAD	0.1057	0.0584	0.1049	0.3907	0.7885	0.5833
Gemma	MS MARCO	0.1609	0.0966	0.1458	0.4708	0.6070	0.6544
Gemma	QTSumm	0.3274	0.1408	0.2617	0.5271	0.0420	0.5480

QueryEval is composed of three complementary subcomponents, each targeting a distinct aspect of query grounding within the model's forward pass. **Query Representation Fidelity (QRF)** measures the layer-wise alignment between query-derived and generation-derived feature representations in the SAE transcoder feature space, quantifying how faithfully each layer preserves and propagates query-relevant information into the generation process. The **Circuit Consistency Score (CCS)** assesses the structural stability of the causal circuit responsible for query-to-generation information flow by introducing targeted token-level corruptions to the input query and identifying which positions causally recover the model's answer prediction; CCS determines whether the model routes query information through a consistent internal pathway across diverse inputs, or relies on surface-level distributional cues. The **Attention Query Alignment (AQA)** provides a head-level decomposition of query grounding by measuring the causal contribution of individual attention heads at every query token position and monitored layer to the model's answer prediction, revealing which architectural sub-components implement query-sensitive computation and which are functionally inert with respect to the query.

Together, these three subcomponents offer a multi-granular, entirely internal view of query grounding—from the feature-space geometry of full-sequence representations (QRF), through the positional structure of causal circuits (CCS), down to the head-level attribution of attention-based information routing (AQA)—without requiring any reference text or external judge model at any stage.

Experiments across three model families (LLaMA-3.1, Qwen-2.5, Gemma-2) and three query-aware generation benchmarks (SQuAD, MS MARCO, QTSumm) demonstrate that **QueryEval** captures failure modes that neither output-level metrics nor LLM-as-judge scores can detect, including cases where a model produces fluent and superficially correct responses while having internally suppressed or bypassed the query signal, as well as cases where LLM-judge scores collapse or become unreliable despite the model exhibiting stable internal query grounding. These results position **QueryEval** not merely as a complementary signal but as a practical, scalable substitute for costly judge-based evaluation in settings where reference outputs are unavailable or repeated large-model inference is prohibitive.

We believe **QueryEval** opens a new direction in the evaluation of language model generation: one grounded not in what the model says, nor in what another model judges it to have said, but in *how* it internally processes the query to arrive at what it says. Future work will explore extending **QueryEval** to multi-turn and retrieval-augmented generation settings, developing training objectives derived from **QRF** and **CCS** signals to improve query grounding during fine-tuning, and scaling the framework to larger model families as efficient transcoder training methods continue to mature.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. Ms marco: A human generated machine reading

comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL workshop BlackboxNLP: analyzing and interpreting neural networks for NLP*, pp. 276–286, 2019.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. GPTScore: Evaluate as you desire. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6556–6576, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.365. URL `https://aclanthology.org/2024.naacl-long.365/`.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL `https://aclanthology.org/2021.emnlp-main.446/`.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yi Li, Haonan Wang, Qixiang Zhang, Boyu Xiao, Chenchang Hu, Hualiang Wang, and Xiaomeng Li. Unieval: Unified holistic evaluation for unified multimodal understanding and generation. *arXiv preprint arXiv:2505.10483*, 2025.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL `https://aclanthology.org/W04-1013/`.

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/`.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. `https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pp. 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL `https://doi.org/10.3115/1073083.1073135`.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. Questeval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 6594–6604, 2021.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. Causal mediation analysis for interpreting neural nlp: The case of gender bias, 2020. URL `https://arxiv.org/abs/2004.12265`.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 5797–5808, 2019.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 563–578, 2019.

Yilun Zhao, Zhenting Qi, Linyong Nan, Boyu Mi, Yixin Liu, Weijin Zou, Simeng Han, Ruizhe Chen, Xiangru Tang, Yumo Xu, et al. Qtsumm: Query-focused summarization over tabular data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1157–1172, 2023.

Table 6: Layer-wise comparison of X_l-QRF and Y_l-QRF for Llama 3.1 8B.

Dataset	**Layer**	X_l**-QRF**	Y_l**-QRF**	**ΔQRF**
MS MARCO	15	0.7040	0.6094	-0.0946
MS MARCO	18	0.7184	0.6781	-0.0403
MS MARCO	20	0.7177	0.6884	-0.0293
MS MARCO	22	0.7405	0.7309	-0.0096
MS MARCO	25	0.7510	0.6741	-0.0769
QTSumm	15	0.7038	0.6082	-0.0956
QTSumm	18	0.7147	0.6784	-0.0363
QTSumm	20	0.7024	0.6644	-0.0380
QTSumm	22	0.7223	0.7224	0.0001
QTSumm	25	0.7338	0.6476	-0.0862
SQuAD	15	0.7494	0.6658	-0.0836
SQuAD	18	0.7606	0.7170	-0.0436
SQuAD	20	0.7591	0.7295	-0.0296
SQuAD	22	0.7759	0.7517	-0.0242
SQuAD	25	0.7855	0.7000	-0.0855

Table 7: Layer-wise comparison of X_l-QRF and Y_l-QRF for Qwen2.5 7B.

Dataset	**Layer**	X_l**-QRF**	Y_l**-QRF**	**ΔQRF**
MS MARCO	13	0.7450	0.2287	-0.5163
MS MARCO	15	0.6777	0.2679	-0.4098
MS MARCO	18	0.6859	0.2277	-0.4582
MS MARCO	20	0.7281	0.4810	-0.2471
MS MARCO	22	0.7888	0.6880	-0.1008
QTSumm	13	0.7640	0.4259	-0.3381
QTSumm	15	0.7231	0.2914	-0.4317
QTSumm	18	0.7400	0.3856	-0.3544
QTSumm	20	0.7790	0.5664	-0.2126
QTSumm	22	0.8328	0.7590	-0.0738
SQuAD	13	0.7535	0.5194	-0.2341
SQuAD	15	0.7003	0.6456	-0.0547
SQuAD	18	0.7054	0.6588	-0.0466
SQuAD	20	0.7513	0.6078	-0.1435
SQuAD	22	0.8132	0.7319	-0.0813

A Appendix

A.1 Extended Results and Discussion

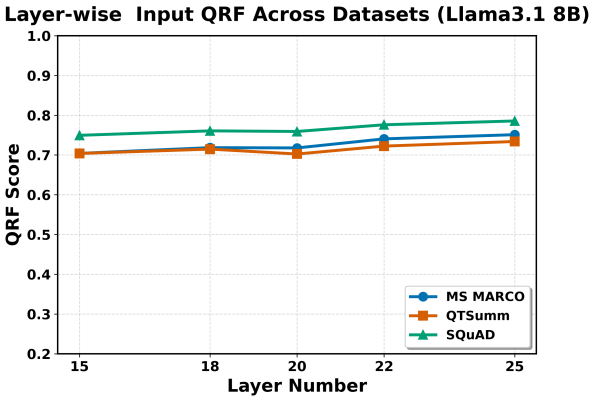
This section presents the complete set of results across all datasets and models, along with the corresponding plots, which were omitted from the main paper due to space constraints.

A.1.1 Query Representation Fidelity

We measured inputlayer QRF score and output layer QRF for the selected layers in each of the mentioned models. For providing a clear understanding of the score variations, we have mentioned each model in separate tables. Llama results have been provided in Table 6, Qwen results have been provided in Table 7 and Gemma results have been provided in Table8. We also provided plots showing the input vs output qrf for each model: llama in Figure 7, qwen in Figure 8 and Gemma in Figure 9.

Table 8: Layer-wise comparison of X_l-QRF and Y_l-QRF for Gemma2 9B.

Dataset	Layer	X_l-QRF	Y_l-QRF	ΔQRF
MS MARCO	19	0.8105	0.9403	0.1298
MS MARCO	23	0.8706	0.9491	0.0785
MS MARCO	26	0.7700	0.9220	0.1520
MS MARCO	29	0.7160	0.9265	0.2105
MS MARCO	33	0.8172	0.9786	0.1614
QTSumm	19	0.7923	0.9473	0.1550
QTSumm	23	0.8492	0.9450	0.0958
QTSumm	26	0.8520	0.9552	0.1032
QTSumm	29	0.8249	0.9673	0.1424
QTSumm	33	0.7720	0.9783	0.2063
SQuAD	19	0.8476	0.9600	0.1124
SQuAD	23	0.9056	0.9636	0.0580
SQuAD	26	0.9121	0.9685	0.0564
SQuAD	29	0.8986	0.9740	0.0754
SQuAD	33	0.8636	0.9820	0.1184



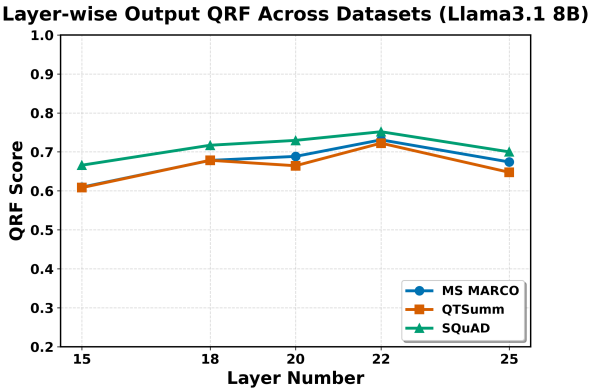


Figure 7: Llama 3.1 layer-wise QRF across datasets for input values (left) and output values (right).

A.1.2 Circuit Coherence Score

Similar to the **QRF**, to provide a clear understanding we have separately added each model result in distinct tables. Llama results are provided in Table9, qwen results are provided in Table 10 and gemma results have been provided in Table 11. We also provide some additional plots for visualizing the CCS pattern of different datasets across different model families in Figure 10

A.1.3 Attention Query Alignment(AQA)

In this section, we provide model-wise AQA scores of each dataset along with their corresponding plots. We have separately provided results of Llama in Table 12, Qwen in Table 13 and Gemma in Table 14.

A.2 Additional Experimental Analysis

To assess the sensitivity of **QueryEval** to its design choices, we conduct targeted ablations along two axes. First, for CCS and AQA, we vary the query corruption fraction $\rho \in \{0.20, 0.40, 0.60\}$ and the circuit extraction threshold $\kappa \in \{0.10, 0.20, 0.60, 0.80\}$, examining how the strength and consistency of the identified circuits depend on the severity of the input perturbation and the selectivity of the top-κ filter. Second, for

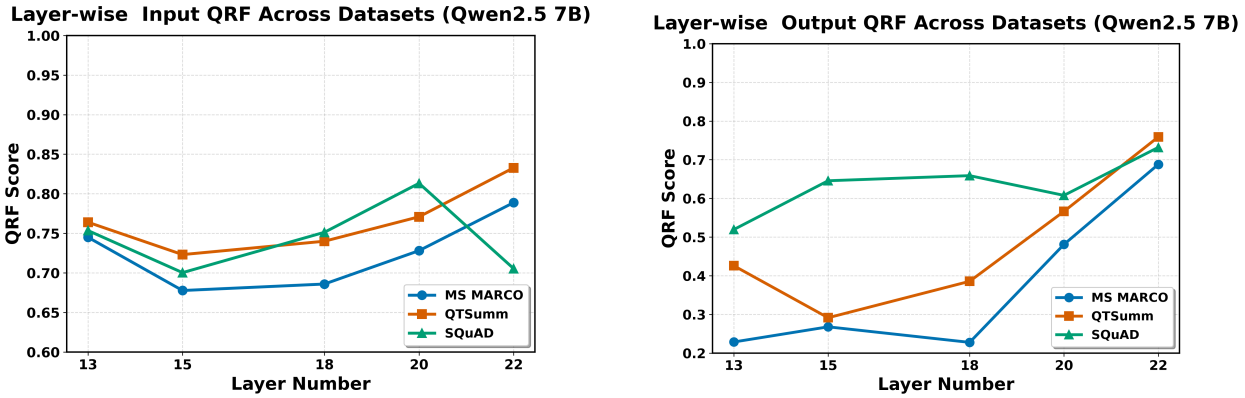


Figure 8: Qwen 2.5 layer-wise QRF across datasets for input values (left) and output values (right).

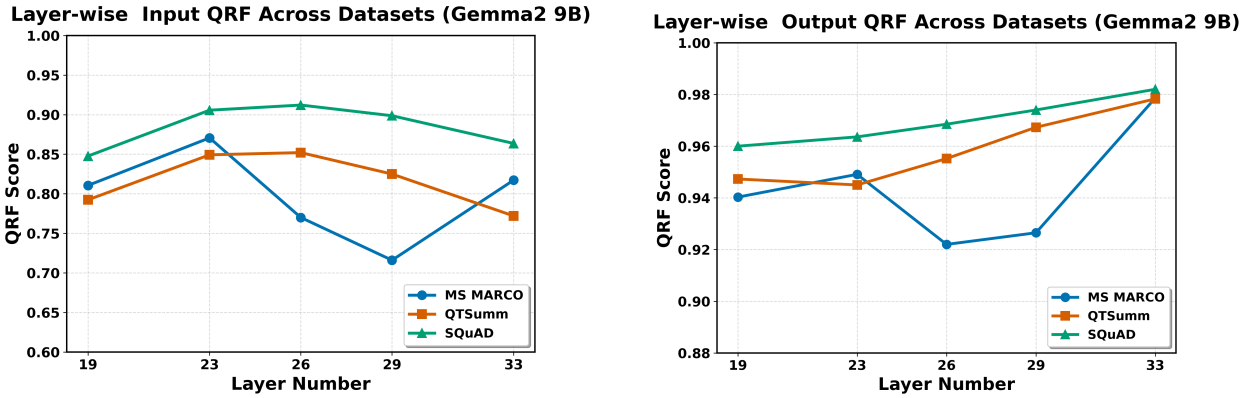


Figure 9: Gemma 2 layer-wise QRF across datasets for input values (left) and output values (right).

QRF, we replace the cosine similarity used to measure query–generation alignment with Euclidean distance, evaluating whether the layer-wise contribution signal is robust to the choice of similarity function.

A.2.1 Corrupt Fraction

To compute the **CCS** score, we experimented with corruption ratios of **0.10, 0.20, 0.40, 0.60, and 0.80**. As shown in Tables 15,16,17, a corruption ratio of 0.40 yielded the maximum **CCS** scores across all datasets and models. Model-wise CCS scores are reported in Tables 15,16,17. Plot depicting the change in CCS scores for varied corrupt ratios has been reported in Figures 11,12,13

A.2.2 Top K ratio

Fixing the corruption ratio at **0.40**, we further tested top-$k \in \{0.25, 0.50, 0.75\}$ to determine the optimal value. Across all datasets and models, top-$k = 0.50$ consistently achieved the highest CCS score, with corresponding results shown in Table 18,19,20.

A.2.3 Euclidean Distance for QRF calculation

In our proposed approach, we have applied cosine similarity between the query feature direction and the generation direction for the calculation of **QRF** scores. However, we also performed experiments using euclidean distance for the same. The results of each model across the three datasets has been reported in Table

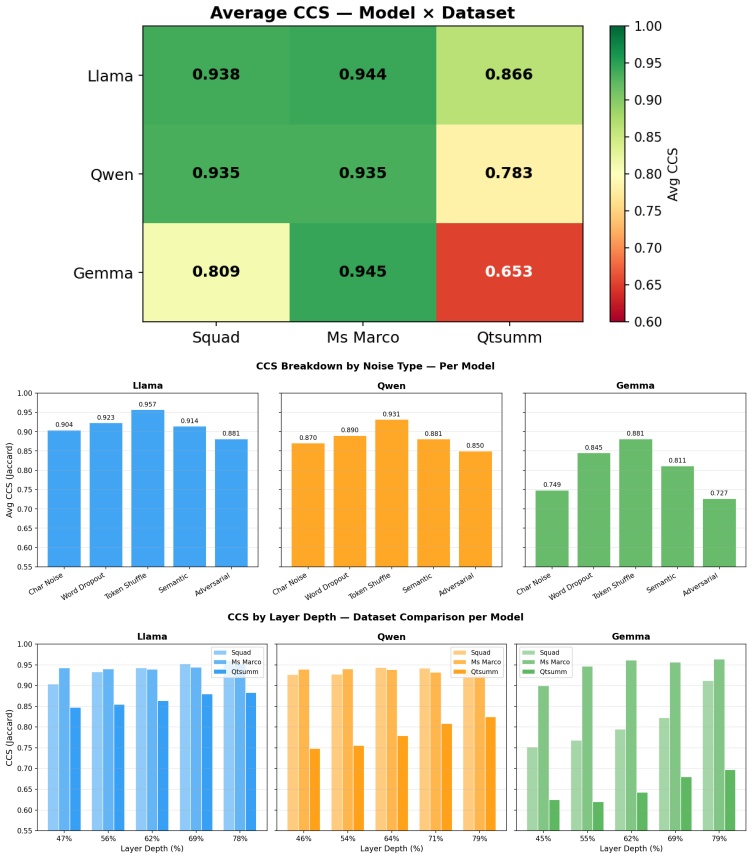


Figure 10: Visualization of Query Representation Fidelity (QRF). From left to right: layer-wise QRF heatmap, CCS breakdown by noise type, and dataset-wise comparison across the evaluated models.

Table 9: Circuit Consistency Score (CCS) across different layers for Llama 3.1 8B.

Dataset	Layer 15	Layer 18	Layer 20	Layer 22	Layer 25
SQuAD	0.903	0.933	0.942	0.952	**0.959**
MS MARCO	0.942	0.940	0.939	0.944	**0.952**
QTSumm	0.847	0.855	0.863	0.880	**0.883**

A.3 Correlation Analysis

In this section, we provide the scatter plots depicting the Pearson correlation between **QueryEval** with the existing metrics like Rouge and BertScore and referencee free metrics like LLM as a judge. The plot between **QueryEval** and Rouge-1(R-1) and Rouge-2(R-2) is provided in Figure 14. The plots between **QueryEval** and Rouge-l(R-L) and BertScore is provided in Figure. 16. The plot between **QueryEval** and LLM as a judge is provided in Figure **??**

Table 10: Circuit Consistency Score (CCS) across different layers for Qwen 2.5 7B.

Dataset	Layer 15	Layer 18	Layer 20	Layer 22	Layer 25
SQuAD	0.926	0.927	**0.943**	0.941	0.938
MS MARCO	**0.939**	**0.939**	0.938	0.931	0.924
QTSumm	0.747	0.755	0.778	0.808	**0.824**

Table 11: Circuit Consistency Score (CCS) across different layers for Gemma2 9B.

Dataset	Layer 19	Layer 23	Layer 26	Layer 29	Layer 33
SQuAD	0.752	0.767	0.795	0.822	**0.911**
MS MARCO	0.899	0.946	0.961	0.956	**0.963**
QTSumm	0.625	0.620	0.643	0.680	**0.697**

Table 12: Layer-wise AQA scores for Llama 3.1 across different datasets.

Model	Dataset	Layer 15	Layer 18	Layer 20	Layer 22	Layer 25
Llama 3.1	SQuAD	0.0012	0.0004	0.0001	0.0013	0.0014
Llama 3.1	MS MARCO	0.0060	0.0010	0.0240	0.0011	0.0064
Llama 3.1	QTSumm	0.0106	0.0071	0.0116	0.0134	0.0055

Table 13: Layer-wise AQA scores for Qwen 2.5 across different datasets.

Model	Dataset	Layer 13	Layer 15	Layer 18	Layer 20	Layer 22
Qwen 2.5	SQuAD	0.0006	0.0000	0.0002	0.0001	0.0001
Qwen 2.5	MS MARCO	0.0009	0.0003	0.0024	0.0019	0.0008
Qwen 2.5	QTSumm	0.1997	0.1890	0.1800	0.1912	0.1867

Table 14: Layer-wise AQA scores for Gemma 2 across different datasets.

Model	Dataset	Layer 19	Layer 23	Layer 26	Layer 29	Layer 33
Gemma 2	SQuAD	0.0001	0.0000	0.0001	0.0002	0.0001
Gemma 2	MS MARCO	0.1917	0.0098	0.0091	0.1667	0.0024
Gemma 2	QTSumm	0.0064	0.0078	0.0066	0.0039	0.0037

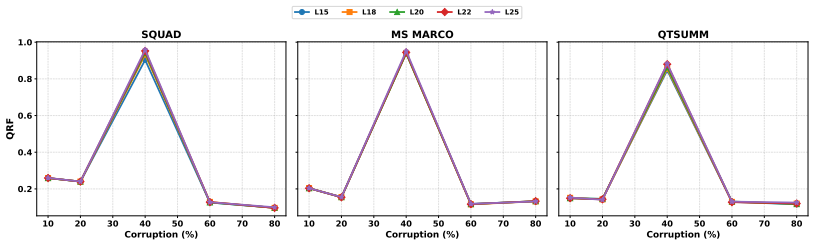


Figure 11: Llama corruption analysis with different corrupt ratios.

Table 15: Sensitivity of QRF to different query corruption ratios for Qwen2.5 7B.

Corrupt (%)	Dataset	L13	L15	L18	L20	L22
10	SQuAD	**0.2622**	0.2621	0.2620	**0.2622**	0.2620
10	MS MARCO	0.1995	0.2004	0.2006	0.1994	**0.2007**
10	QTSumm	0.1397	0.1412	0.1410	0.1403	**0.1443**
20	SQuAD	0.2530	0.2531	0.2532	**0.2533**	0.2530
20	MS MARCO	0.1420	0.1424	0.1423	**0.1430**	0.1424
20	QTSumm	0.1336	0.1349	0.1378	0.1385	**0.1394**
40	SQuAD	0.9260	0.9271	**0.9433**	0.9416	0.9385
40	MS MARCO	**0.9394**	0.9398	0.9380	0.9315	0.9247
40	QTSumm	0.7478	0.7557	0.7787	0.8081	**0.8242**
60	SQuAD	0.1394	0.1409	**0.1414**	0.1409	0.1398
60	MS MARCO	0.1042	0.1039	0.1050	**0.1053**	0.1045
60	QTSumm	0.1423	0.1405	0.1388	0.1310	**0.1439**
80	SQuAD	0.0667	0.0659	0.0670	0.0669	**0.0674**
80	MS MARCO	0.0891	0.0893	**0.0900**	0.0896	0.0893
80	QTSumm	0.1099	0.1122	0.1124	0.1119	**0.1143**

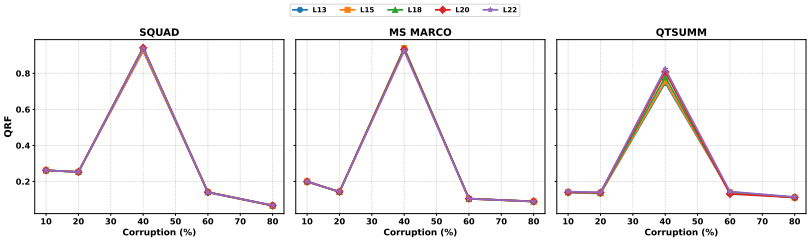


Figure 12: Qwen corruption analysis with different corrupt ratios.

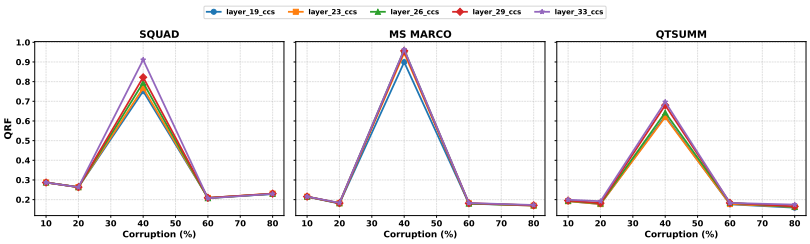


Figure 13: Gemma corruption analysis with different corrupt ratios.

Table 16: Sensitivity of QRF to different query corruption ratios for Llama 3.1 8B.

Corrupt (%)	Dataset	L15	L18	L20	L22	L25
10	SQuAD	0.2592	0.2593	0.2590	0.2594	**0.2603**
10	MS MARCO	0.2039	**0.2043**	0.2038	0.2038	0.2041
10	QTSumm	0.1489	0.1512	0.1492	0.1493	**0.1530**
20	SQuAD	**0.2400**	**0.2400**	**0.2400**	**0.2400**	**0.2400**
20	MS MARCO	0.1546	0.1552	0.1550	0.1544	**0.1554**
20	QTSumm	**0.1458**	0.1425	0.1453	0.1433	0.1406
40	SQuAD	0.9032	0.9328	0.9420	0.9523	**0.9588**
40	MS MARCO	0.9421	0.9402	0.9390	0.9443	**0.9519**
40	QTSumm	0.8469	0.8549	0.8632	0.8799	**0.8828**
60	SQuAD	0.1258	**0.1291**	0.1261	0.1281	0.1276
60	MS MARCO	0.1180	0.1161	0.1171	0.1170	**0.1185**
60	QTSumm	0.1290	0.1292	**0.1312**	0.1278	0.1306
80	SQuAD	0.0969	0.0964	0.0972	0.0966	**0.0999**
80	MS MARCO	0.1326	0.1321	0.1332	**0.1333**	0.1317
80	QTSumm	0.1165	0.1201	0.1170	0.1202	**0.1254**

Table 17: Sensitivity of CCS to different query corruption ratios for Gemma2 9B. The highest CCS for each dataset and corruption level is highlighted in bold.

Corrupt (%)	Dataset	L19	L23	L26	L29	L33
10	SQuAD	**0.2872**	**0.2872**	**0.2872**	0.2870	0.2870
10	MS MARCO	0.2147	0.2148	**0.2154**	**0.2154**	0.2153
10	QTSumm	0.1919	0.1920	0.1943	0.1954	**0.1995**
20	SQuAD	**0.2633**	**0.2633**	**0.2633**	**0.2633**	**0.2633**
20	MS MARCO	0.1800	0.1815	0.1818	0.1822	**0.1836**
20	QTSumm	0.1776	0.1774	0.1802	0.1821	**0.1919**
40	SQuAD	0.7515	0.7674	0.7947	0.8221	**0.9111**
40	MS MARCO	0.8992	0.9461	0.9613	0.9563	**0.9634**
40	QTSumm	0.6249	0.6197	0.6430	0.6801	**0.6968**
60	SQuAD	**0.2096**	0.2087	0.2073	0.2089	0.2068
60	MS MARCO	0.1795	0.1808	0.1800	0.1818	**0.1828**
60	QTSumm	0.1762	0.1776	**0.1840**	0.1831	0.1833
80	SQuAD	0.2283	**0.2306**	0.2296	**0.2306**	0.2292
80	MS MARCO	0.1697	0.1692	0.1710	0.1710	**0.1727**
80	QTSumm	0.1591	0.1627	0.1626	0.1657	**0.1747**

Table 18: Sensitivity of CCS to different Top-κ circuit thresholds for Llama 3.1 8B.

Top-κ	Dataset	L15	L18	L20	L22	L25
0.25	SQuAD	0.0910	0.0908	0.0909	**0.0914**	0.0910
0.25	MS MARCO	0.0764	0.0774	0.0765	0.0759	**0.0778**
0.25	QTSumm	0.0632	0.0629	0.0639	0.0638	**0.0655**
0.50	SQuAD	0.9032	0.9328	0.9420	0.9523	**0.9588**
0.50	MS MARCO	0.9421	0.9402	0.9390	0.9443	**0.9519**
0.50	QTSumm	0.8469	0.8549	0.8632	0.8799	**0.8828**
0.75	SQuAD	0.2349	0.2346	0.2346	**0.2351**	0.2338
0.75	MS MARCO	0.1728	0.1727	0.1731	0.1733	**0.1739**
0.75	QTSumm	0.2169	0.2128	0.2136	**0.2212**	0.2200

Table 19: Sensitivity of CCS to different Top-κ circuit thresholds for Qwen2.5 7B.

Top-κ	Dataset	L13	L15	L18	L20	L22
0.25	SQuAD	0.1091	0.1092	0.1099	0.1092	**0.1103**
0.25	MS MARCO	**0.0612**	0.0601	0.0589	0.0607	0.0595
0.25	QTSumm	0.0661	0.0639	**0.0680**	0.0652	0.0634
0.50	SQuAD	0.9260	0.9271	**0.9433**	0.9416	0.9385
0.50	MS MARCO	**0.9394**	0.9398	0.9380	0.9315	0.9247
0.50	QTSumm	0.7478	0.7557	0.7787	0.8081	**0.8242**
0.75	SQuAD	0.2832	0.2832	**0.2835**	**0.2835**	0.2834
0.75	MS MARCO	0.2293	0.2301	**0.2311**	**0.2311**	0.2295
0.75	QTSumm	0.2132	**0.2161**	0.2145	0.2121	0.2154

Table 20: Sensitivity of CCS to different Top-κ circuit thresholds for Gemma2 9B.

Top-κ	Dataset	L19	L23	L26	L29	L33
0.25	SQuAD	0.1269	**0.1278**	0.1269	0.1263	0.1270
0.25	MS MARCO	0.0975	0.0985	0.0986	**0.1000**	0.0998
0.25	QTSumm	0.0930	0.0912	0.0923	**0.0998**	0.0983
0.50	SQuAD	0.7515	0.7674	0.7947	0.8221	**0.9111**
0.50	MS MARCO	0.8992	0.9461	0.9613	0.9563	**0.9634**
0.50	QTSumm	0.6249	0.6197	0.6430	0.6801	**0.6968**
0.75	SQuAD	0.3192	0.3193	**0.3194**	0.3193	0.3192
0.75	MS MARCO	0.2586	0.2583	0.2590	0.2597	**0.2601**
0.75	QTSumm	0.2362	0.2369	0.2407	0.2401	**0.2422**

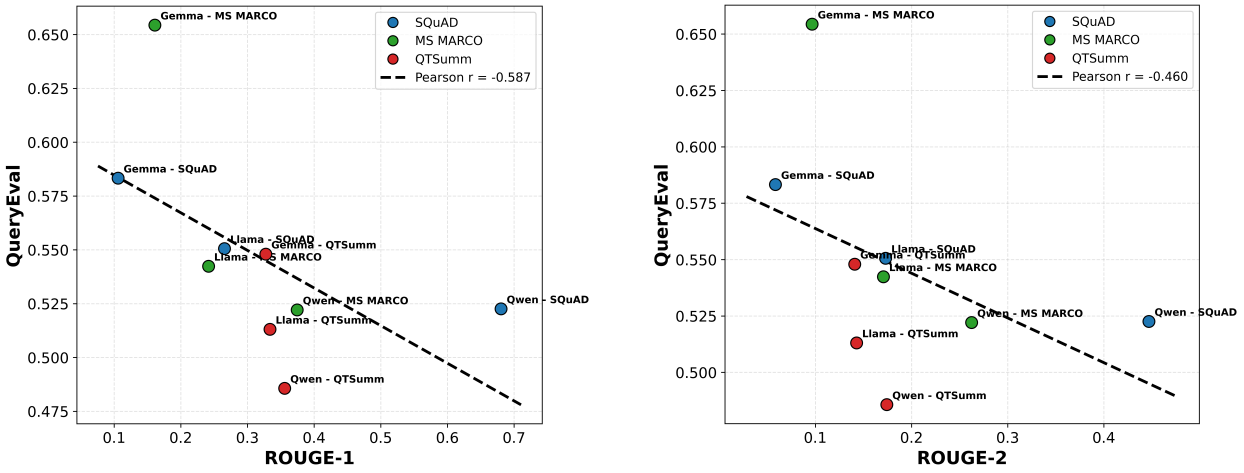


Figure 14: Scatter plots showing Pearson correlation between R1 and QueryEval(left) and R2 and QueryEval(right).

Dataset	Model	Layer	X_{layer} QRF	Y_{layer} QRF	Δ QRF
MS MACRO	Gemma2 9B	19	0.7205	0.6916	-0.0289
MS MACRO	Gemma2 9B	23	0.7471	0.7410	-0.0061
MS MACRO	Gemma2 9B	26	0.7543	0.6777	-0.0766
MS MACRO	Gemma2 9B	29	0.7914	0.8270	0.0356
MS MACRO	Gemma2 9B	33	0.7813	0.7815	0.0002
QTSUMM	Gemma2 9B	19	0.1053	0.3317	0.2264
QTSUMM	Gemma2 9B	23	0.1026	0.3213	0.2187
QTSUMM	Gemma2 9B	26	0.0953	0.2907	0.1954
QTSUMM	Gemma2 9B	29	0.0877	0.1869	0.0992
QTSUMM	Gemma2 9B	33	0.0950	0.2106	0.1156
SQUAD	Gemma2 9B	19	0.7630	0.7248	-0.0382
SQUAD	Gemma2 9B	23	0.7829	0.7653	-0.0176
SQUAD	Gemma2 9B	26	0.7863	0.7396	-0.0467
SQUAD	Gemma2 9B	29	0.8119	0.8648	0.0529
SQUAD	Gemma2 9B	33	0.7983	0.7612	-0.0371
MS MACRO	Qwen 2.5 7B	13	0.6740	0.5715	-0.1025
MS MACRO	Qwen 2.5 7B	15	0.7040	0.6094	-0.0946
MS MACRO	Qwen 2.5 7B	18	0.7184	0.6781	-0.0403
MS MACRO	Qwen 2.5 7B	20	0.7177	0.6884	-0.0293
MS MACRO	Qwen 2.5 7B	22	0.7405	0.7309	-0.0096
QTSUMM	Qwen 2.5 7B	13	0.1132	0.3525	0.2393
QTSUMM	Qwen 2.5 7B	15	0.1130	0.3316	0.2186
QTSUMM	Qwen 2.5 7B	18	0.1066	0.3322	0.2256
QTSUMM	Qwen 2.5 7B	20	0.1040	0.3327	0.2287
QTSUMM	Qwen 2.5 7B	22	0.1017	0.3155	0.2138
SQUAD	Qwen 2.5 7B	13	0.7278	0.6385	-0.0893
SQUAD	Qwen 2.5 7B	15	0.7234	0.7123	-0.0111
SQUAD	Qwen 2.5 7B	18	0.7641	0.7435	-0.0206
SQUAD	Qwen 2.5 7B	20	0.7591	0.7295	-0.0296
SQUAD	Qwen 2.5 7B	22	0.7759	0.7517	-0.0242
MS MACRO	LLAMA 3.1 8B	15	0.7040	0.6094	-0.0946
MS MACRO	LLAMA 3.1 8B	18	0.7184	0.6781	-0.0403
MS MACRO	LLAMA 3.1 8B	20	0.7177	0.6884	-0.0293
MS MACRO	LLAMA 3.1 8B	22	0.7405	0.7309	-0.0096
MS MACRO	LLAMA 3.1 8B	25	0.7510	0.6741	-0.0769
QTSUMM	LLAMA 3.1 8B	15	0.1130	0.3316	0.2186
QTSUMM	LLAMA 3.1 8B	18	0.1066	0.3322	0.2256
QTSUMM	LLAMA 3.1 8B	20	0.1040	0.3327	0.2287
QTSUMM	LLAMA 3.1 8B	22	0.1017	0.3155	0.2138
QTSUMM	LLAMA 3.1 8B	25	0.1021	0.3135	0.2114
SQUAD	LLAMA 3.1 8B	15	0.0000	0.0000	0.0000
SQUAD	LLAMA 3.1 8B	18	0.0000	0.0000	0.0000
SQUAD	LLAMA 3.1 8B	20	0.0000	0.0000	0.0000
SQUAD	LLAMA 3.1 8B	22	0.0000	0.0000	0.0000
SQUAD	LLAMA 3.1 8B	25	0.0000	0.0000	0.0000

Table 21: Layer-wise QRF scores for X_{layer} and Y_{layer} across Gemma2-9B, Qwen2.5-7B, and Llama3.1-8B.

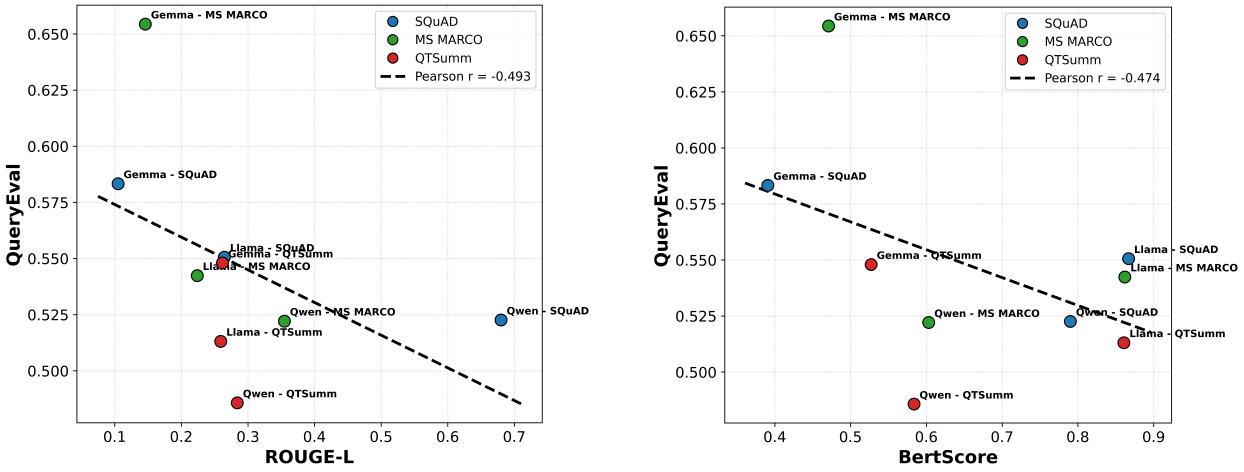


Figure 15: Scatter plots showing Pearson correlation between R-L and QueryEval(left) and BertScore and QueryEval(right).

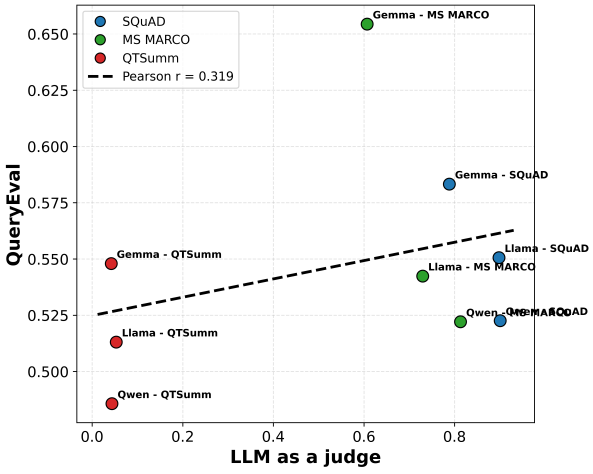


Figure 16: Scatter plots showing Pearson correlation between LLM as judge and QueryEval.