SHAPLEY-GUIDED SET-WISE RERANKING FOR RETRIEVAL-AUGMENTED GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation (RAG) relies on reranking to select a small set of documents for a downstream LLM under a limited context budget. Existing rerankers either score documents individually or rank candidates jointly, but lack a principled way to model both the utility of a document subset and each document's contribution to that utility. We formulate RAG reranking as a cooperative game, where documents are players and a coalition's payoff measures its usefulness for the query. We introduce a set-wise valuer to model this coalition utility and show that a local desirability condition on the valuer is sufficient to induce the desired document ordering under a semivalue framework. This result motivates a set-contrastive training objective that avoids explicitly computing attribution values during training. At inference, we instantiate the same local-comparison principle with a practical restricted regression attribution rule. Specifically, we sample small coalitions according to a truncated KernelSHAP-inspired distribution and fit an additive regression to their utilities. We further train an amortized estimator to predict these attribution scores in a single forward pass. Across five multi-hop QA benchmarks, our method outperforms strong point-wise, list-wise, and set-wise baselines in reranking recall and downstream answer quality in most cases, while the amortized estimator retains most of the gain at a latency comparable to that of a standard point-wise reranker, yielding a $7.2\times$ speedup over sampling-based attribution.

1 INTRODUCTION

Retrieval-augmented generation (RAG) answers a query by retrieving documents and conditioning a downstream LLM on them (Lewis et al., 2020; Gao et al., 2023). Because the LLM's context budget is limited and noisy contexts may degrade generation, a practical RAG system rarely feeds the retriever's output to the generator directly: a first-stage retriever returns a large candidate set (Karpukhin et al., 2020), and a reranker decides which small subset of documents is ultimately exposed to the model (Glass et al., 2022). This reranking stage is critical: its selection directly determines what evidence the LLM can draw on, and thereby governs the quality of the final answer (Liu et al., 2024; Cuconasu et al., 2024).

Most existing rerankers operate at one of three main granularities (Zhu et al., 2025). *Point-wise* rerankers score each candidate document independently (Nogueira & Cho, 2019; Nogueira et al., 2020); they are efficient and align well with common cross-encoder training objectives. *List-wise* rerankers instead process multiple candidates jointly and output an ordering, thereby exposing the model to cross-document information (Sun et al., 2023; Ma et al., 2023; Pradeep et al., 2023b). More recently, *set-wise* methods have also emerged that prompt or train models over groups of documents, primarily as a means to compare candidates more efficiently or robustly (Zhuang et al., 2024).

Each of these families, however, has characteristic limitations. Point-wise scoring treats documents in isolation and cannot model interactions among candidates, such as complementarity. List-wise reranking exposes such interactions but introduces its own difficulties: long contexts are hard to use reliably (Liu et al., 2024), and sliding-window strategies yield local rankings that are difficult to combine (Sun et al., 2023). Existing set-wise methods, in turn, use the set mainly as a *comparison unit* for producing an ordering, rather than as an object whose utility is modeled in its own right.

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.

These limitations expose a common missing capability: existing paradigms do not jointly model how useful a document subset is and how much each document contributes to that utility.

In this paper, we propose to answer both questions at once by formulating the reranking as a *cooperative game*, motivated by the Shapley value (Shapley, 1953): documents are players, any subset of documents forms a coalition, and the payoff of a coalition is its usefulness for answering the query. Under this formulation, reranking becomes a *credit assignment* problem, and each document's contribution is naturally measured by its attribution value within the candidate set. For ranking, only the *relative order* of contributions matters, which lets us work with the broader family of semivalues. At inference, we deploy a budget-restricted regression attribution rule over small coalitions. This restricted attribution rule matches the limited context the downstream LLM actually sees and preserves the same local-comparison ordering structure analyzed in our theory.

Instantiating this game requires its utility function, i.e., a utility over document subsets. To supply it, we introduce a *set-wise valuer*: a model that takes a query and a subset of candidate documents as input and outputs the joint utility of that subset. Unlike point-wise models, it is trained directly on subset-level comparisons; unlike list-wise models, it need not produce a full ordering over lengthy contexts; and unlike prior set-wise rerankers, it treats the subset as the object being valued rather than a vehicle for pairwise-style comparison. Crucially, the same cooperative-game perspective yields its training signal: we show that, under a symmetric semivalue aggregation, the ordering between a positive and a negative document is governed by their local utility advantages averaged over coalition contexts. This motivates a simple set-contrastive loss over sampled contexts paired with positive and negative documents, allowing the valuer to be trained without explicitly computing attribution scores. At inference, we operate in the small-coalition regime relevant to the downstream context budget using truncated KernelSHAP-style weighted regression. This yields budget-restricted attribution scores from sampled coalitions. While inspired by Shapley weighting, the truncated regression defines the practical attribution rule in our experiments. We further train an amortized estimator to predict these scores in a single forward pass. Empirically, set-wise valuation with attribution-based reranking improves over strong baselines, while amortization preserves most of the gain and even slightly improves quality at substantially lower inference cost. We summarize our contributions:

- **A principled formulation.** We cast reranking as a *cooperative game* over documents, with a learned set-wise valuer serving as the utility function over document coalitions. This formulation provides a principled view of reranking as document-level credit assignment under joint subset utility.
- **A training principle from the game.** We show that, for symmetric semivalue aggregation, the score gap between two documents can be characterized through their local utility advantages across coalition contexts. This motivates a set-contrastive objective that encourages the desired ordering without explicitly materializing attribution scores during training.
- **Practical attribution algorithms.** We instantiate budget-restricted attribution with truncated KernelSHAP-style regression and amortize the same attribution rule into a single forward pass.
- **Empirical validation.** We show that Shapley-guided set-wise reranking improves over strong point-wise, list-wise, and set-wise baselines, while amortization retains most of the gain at substantially lower inference cost.

2 Related Work

Reranking paradigms. RAG systems retrieve candidates from external corpora and rerank them before generation (Lewis et al., 2020; Jiang et al., 2023; Robertson & Zaragoza, 2009; Karpukhin et al., 2020; Gao & Callan, 2022; Gao et al., 2023; Glass et al., 2022). Because generation depends on the composition and ordering of the retrieved context (Liu et al., 2024; Cuconasu et al., 2024), rerankers increasingly model interactions among candidates. Point-wise methods score documents independently (Nogueira & Cho, 2019; Nogueira et al., 2020), while pair-wise methods model local preferences that must be aggregated into a ranking (Burges et al., 2005; Qin et al., 2024; Pradeep et al., 2021). List-wise methods jointly process the candidate list through ranking objectives or direct permutation generation (Cao et al., 2007; Xia et al., 2008; Sun et al., 2023; Ma et al., 2023),

but can be costly and sensitive to input order (Tang et al., 2024). Set-wise approaches capture cross-document interactions through permutation-invariant attention, group-wise selection, or tournament-style ranking (Pang et al., 2020; Schlatt et al., 2025; Zhuang et al., 2024; Chen et al., 2025).

Positioning of our work. Existing point-wise, pair-wise, list-wise, and set-wise rerankers ultimately produce document scores, an ordering, or a selected set, without defining a queryable utility over arbitrary document coalitions together with a decomposition of that utility into document-level credit. Our framework combines *explicit* coalition-utility learning with regression attribution motivated by semivalue ordering, converting context-dependent marginal contributions into a document ranking. It further supports both sampling-based attribution and an amortized estimator for efficient deployment.

3 PRELIMINARIES

Problem formulation. Given a query q and candidate set C_q, we define a query-conditioned cooperative game whose utility function $v(q, S)$ represents a surrogate utility for coalition $S \subseteq C_q$. In our implementation, this utility is learned from relevance supervision rather than observed directly from downstream answer correctness. Our theoretical results characterize the ranking induced by this learned utility; they do not assume that the learned utility function is a perfect measure of end-to-end answer utility. We first define the *ideal* full-game reranking score of document d_i as its Shapley value $\phi_i^{(q)}$ under this game, which captures the synergistic effect among *document subsets*.

Cooperative games and Shapley values. The Shapley value was first proposed by Shapley (1953) and became the central concept in cooperative games. It calculates each player's average marginal contribution across all possible coalitions. For brevity, we write $v(S) = v(q, S)$. Formally, in the context of our reranking, the Shapley value of document i is defined as

$$\phi_i^{(q)} = \sum_{S \subseteq \mathcal{C}_q \setminus \{d_i\}} \frac{|S|!\,(n - |S| - 1)!}{n!} (v(S \cup \{d_i\}) - v(S)) \tag{1}$$

Shapley value is the only solution satisfying efficiency, symmetry, additivity, and the dummy player property (Winter, 2002). However, computing the exact Shapley values is generally intractable, as Eq. 1 requires evaluating an exponential number of subsets S. Charnes et al. (1988) showed, in the context of cooperative game theory, that the Shapley value can be obtained as the solution to a constrained weighted least squares (WLS) problem. Decades later, this formulation was independently rediscovered by Lundberg & Lee (2017) in the context of machine learning model interpretation, giving rise to the KernelSHAP algorithm. The KernelSHAP reformulates the Shapley value computation in Eq. 1 as the following WLS problem

$$\arg \min_{\phi_0, \cdots, \phi_n} \sum_{S \in \mathcal{P}(\mathcal{C}_q)} k(n, |S|)(\phi_0 + \sum_{j \in \text{Idx}(S)} \phi_j - v(S))^2 \tag{2}$$

where $k(n, |S|) = \frac{n-1}{\binom{n}{|S|}|S|(n-|S|)}$ is regarded as the Shapley kernel. When the sum in Eq. 2 is evaluated over all 2^n subsets $S \subseteq \mathcal{C}_q$, the resulting $\phi_1, \ldots, \phi_n$ are theoretically guaranteed to coincide exactly with the Shapley values (Charnes et al., 1988; Lundberg & Lee, 2017). The idea of KernelSHAP lies in approximating this WLS objective via Monte Carlo sampling of coalitions S, which turns an exponential-cost exact computation into a tractable, efficient estimation procedure. In the reranking setting, however, our focus is on interactions among relatively small document subsets rather than on coalitions of arbitrary size. We therefore impose a budget on the coalition size and introduce the resulting *budget-restricted semivalue*.

Definition 1 (Budget-restricted semivalue)**.** Fix a coalition budget B with $0 \le B \le n-1$ and a weight family $\{w(k)\}_{k=0}^{B}$ with $w(k) > 0$ for $0 \le k \le B$, the convention $w(k) = 0$ for $k > B$, and the normalisation $\sum_{k=0}^{B} \binom{n-1}{k} w(k) = 1$. The budget-restricted semivalue of document d_i is

$$\phi_i^{(q),B} = \sum_{\substack{S \subseteq \mathcal{C}_q \setminus \{d_i\} \\ |S| \le B}} w(|S|)\big(v(S \cup \{d_i\}) - v(S)\big).$$

Taking $B = n-1$ and $w(k) = k!\,(n-k-1)!/n!$ recovers the Shapley value of Eq. 1. For $B < n-1$, $\phi^{(q),B}$ is a semivalue with non-negative weights: it retains linearity, symmetry, the dummy-player property and monotonicity, and relaxes only efficiency, which is not needed for ranking.

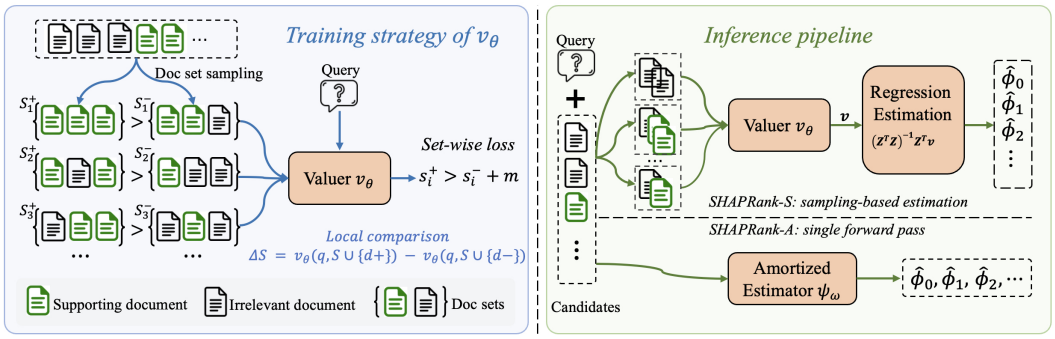


Figure 1: The framework of our Shapley-guided reranking method SHAPRANK.

4 METHODOLOGY

The overall framework is shown in Figure 1. We first train the set-wise valuer v_θ with a set-contrastive objective, whose validity is grounded in the ordering guarantee of Theorem 1. Building on the trained v_θ, we further train an amortized attribution estimator (Sec. 4.3). This yields two inference pipelines. The sampling-based pipeline estimates document budget-restricted semivalues as attribution scores from restricted coalition regression, while the amortized pipeline predicts the same regression scores directly in a single forward pass.

4.1 SHAPLEY-GUIDED SET-WISE CONTRASTIVE TRAINING

Set-wise valuer. Since we build our reranking framework guided by the cooperative game, we require a set-wise valuer $v_\theta(q, S)$ that estimates the utility of a subset S of documents for query q. Ideally, we would like the attribution scores of each document, as computed from v_θ, to faithfully reflect its gold relevance labels. This suggests a natural training strategy: at each training step, estimate each document's attribution scores using the current valuer v_θ, and update θ by supervising these estimates against the gold relevance labels. However, this strategy is intractable in practice due to the repeated cost of estimating these scores at each iteration. We show that this issue can be circumvented entirely: as established by Theorem 1 below, the correct *ordering* of documents is already guaranteed once v_θ satisfies a purely local condition on single-document marginal contributions. This allows v_θ to be trained efficiently via a set-wise contrastive objective directly.

Theorem 1 (Semivalue gap as an average local advantage)**.** *Let $v_\theta(q, \cdot)$ be the set valuer for query q, viewed as the utility function of the cooperative game on $\mathcal{C}_q$ with $|\mathcal{C}_q| = n$, and let $\phi_\bullet^{(q),B}$ be the budget-restricted semivalue of Definition 1 with any admissible weight family $w(\cdot)$. For $d_i, d_j \in \mathcal{C}_q$ define the local advantage of d_i over d_j on context S as*

$$\Delta_S \;:=\; v_\theta(q, S \cup \{d_i\}) - v_\theta(q, S \cup \{d_j\}).$$

Let $u(S) := w(|S|) + w(|S|+1)$ for $S \subseteq \mathcal{C}_q \setminus \{d_i, d_j\}$ with $|S| \le B$. Then u is a probability distribution on $\{S \subseteq \mathcal{C}_q \setminus \{d_i, d_j\} : |S| \le B\}$ that depends only on n and w, and

$$\phi_i^{(q),B} - \phi_j^{(q),B} \;=\; \mathbb{E}_{S\sim u}\big[\Delta_S\big]. \tag{3}$$

Consequently $\phi_i^{(q),B} > \phi_j^{(q),B}$ if and only if $\mathbb{E}_{S\sim u}[\Delta_S] > 0$.

Corollary 2 (A pointwise sufficient condition)**.** *If $\Delta_S > 0$ for every S in the support of u, then $\phi_i^{(q),B} > \phi_j^{(q),B}$.*

Remark. Corollary 2 is the classical local monotonicity of semivalues under the strict desirability relation of Isbell (1958), and we do not claim it as new. We state Theorem 1 rather than the corollary alone because the two conditions play different roles in our method: the corollary is what our training objective directly optimizes. It is local, sampling-friendly, and never materializes a Shapley or semivalue, whereas Eq. 3 is the strictly weaker condition the ranking actually requires. A learned valuer will violate $\Delta_S > 0$ on some contexts (Fig. 6); Eq. 3 says that those violations are outvoted as

long as the remaining contexts dominate in expectation, which is exactly the failure mode a pointwise valuer cannot recover from. Our two design choices supply the two factors of this expectation: set-wise training makes Δ_S typically positive for $|S| > 0$, and semivalue aggregation converts an average positive local advantage into the desired ordering.

Training strategy. Enforcing the condition over all document sets S is intractable, so we optimize a Monte Carlo surrogate that samples sets and replaces the strict inequalities with soft margins. For a query q, a sampled document set S, a positive d^+, and a negative d^-, write the set utilities as $s^+ = v_\theta(q, S \cup \{d^+\}), s^- = v_\theta(q, S \cup \{d^-\})$. We train the valuer v_θ with the set-wise contrastive objective as follows:

$$\mathcal{L}_{\text{set}}(\theta) = \mathbb{E}_{(q,d^+,d^-),\,S}\left[\max\big(m - s^+ + s^-, 0\big)\right]$$

where m is a margin. Minimizing $\mathcal{L}_{\text{set}}$ encourages positive local advantages on sampled coalition contexts. It therefore provides a stochastic surrogate for the local desirability condition in Theorem 1, without requiring attribution computation during training. The theorem explains why such local advantages are relevant to semivalue ranking: their weighted average determines the score gap between two documents.

Permutation robustness. A cooperative-game utility function is formally defined on unordered coalitions. Our implementation concatenates the documents in a coalition and is therefore not exactly permutation invariant by construction. We mitigate this dependence by uniformly randomizing document order during training. At inference, coalition evaluations can consequently be viewed as noisy evaluations of an order-averaged utility $\bar{v}_\theta(q, S) = \mathbb{E}_\pi[v_\theta(q, \pi(S))]$, which is permutation invariant. We additionally quantify the residual permutation sensitivity in Table 10 in Appendix C.3 by evaluating each coalition under multiple random permutations.

4.2 Sampling-based Restricted Attribution from Coalition Samples

Given a trained set-wise valuer, we compute document attribution scores using a restricted KernelSHAP-inspired regression rule. The construction follows the KernelSHAP weighting principle, but because we restrict the sampled coalition sizes, the resulting scores should be viewed as restricted regression attributions rather than exact full-game Shapley values. We draw T coalitions $S_1, \ldots, S_T \subseteq \mathcal{C}_q$ from the restricted KernelSHAP distribution

$$p_B(S) = \frac{k(n, |S|)\mathbb{I}[1 \le |S| \le B]}{\sum_{S' \subseteq C_q} k(n, |S'|)\mathbb{I}[1 \le |S'| \le B]},$$

and represent each sampled coalition by a binary indicator $z_t \in \{0,1\}^n$, where $z_{t,k} = 1$ iff $d_k \in S_t$.

We obtain the attribution vector as the least-squares solution: $\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^n} \|Z\boldsymbol{\beta} - \mathbf{v}\|_2^2$. Because coalition weights are incorporated through the sampling distribution, no additional sample weights are applied in this finite-sample regression. We solve this objective using a numerically stable least-squares routine. $\mathbf{v} \in \mathbb{R}^T$ collects the centered valuer outputs, $v_t = v_\theta(q, S_t) - v_\theta(q, \varnothing)$, so that the attributions are measured relative to the empty-coalition baseline.

Remark. Importantly, the restricted regression preserves the local-comparison structure that motivates our training objective. Theorem 3 in Appendix A.2 shows that the population regression scores β_i^* satisfy $\beta_i^* - \beta_j^* = \mathbb{E}_{S \sim u_{\text{reg}}}[v_\theta(q, S \cup \{d_i\}) - v_\theta(q, S \cup \{d_j\})]$. Thus, positive local advantages imply the desired ordering for the practical regression rule as well. *The sampled least-squares estimator converges to this population rule as T increases.* Hence, in practice we sort documents in descending order using $\hat{\boldsymbol{\beta}}_k$ and select the top-m as the context passed to the downstream LLM.

4.3 Amortized Attribution Estimation

The sampling-based procedure in Sec. 4.2 requires T forward passes of the valuer per query. We therefore amortize the same restricted regression attribution rule with a parametric estimator $\psi_\omega(q, C_q) \in \mathbb{R}^n$ to improve efficiency further. Rather than precomputing attribution labels for every query, we train the estimator directly from coalition utilities under the same regression objective used by the sampling-based procedure. At the population optimum, the predicted document scores therefore target the same restricted regression coefficients defined in Sec. 4.2. The estimator is

trained to explain the (centered) set utilities additively:

$$\mathcal{L}_{\text{est}}(\omega) = \mathbb{E}_q \, \mathbb{E}_{S\sim p_B(S)} \left[\left(v_\theta(q,S) - v_\theta(q,\emptyset) - \mathbf{1}_S^\top \psi_\omega(q,\mathcal{C}_q) \right)^2 \right],$$

where $\mathbf{1}_S \in \{0,1\}^n$ is the indicator vector of the coalition S (with $[\mathbf{1}_S]_k = 1$ iff $d_k \in S$); the k-th entry of vector $\psi_\omega(q,\mathcal{C}_q)$ is the predicted value of candidate d_k. We train the amortized estimator under the same restricted coalition distribution p_B used by the sampling-based pipeline. This matching is important because changing the coalition distribution changes the weighted regression objective and therefore the attribution scores being approximated.

We instantiate ψ_ω with a pretrained ModernBERT (Warner et al., 2025) encoder: the query is concatenated with all candidate documents following the order given by the first-stage retriever. Token embeddings of each document are aggregated by average pooling, and an extra MLP head maps the pooled representation to a scalar prediction. The framework is agnostic to this choice; any encoder producing one score per candidate applies.

4.4 Inference

At deployment, both pipelines target the same budget-restricted regression attribution rule. Sampling-based inference follows Sec. 4.2: it draws T coalitions from the restricted KernelSHAP-inspired distribution, evaluates their utilities, and solves the resulting least-squares problem. Amortized inference instead predicts the corresponding scores with $\psi_\omega(q, C_q)$ in a single forward pass, matching the cost of a standard neural reranker. Notably, the two do not exhibit a strict quality–efficiency trade-off: the amortized estimator matches and sometimes surpasses its sampling-based teacher (Sec. 6). This is consistent with prior amortized attribution results (Jethani et al., 2022), where learning across many coalitions can smooth finite-sample variation in per-query estimation.

5 Experimental Setup

Baselines. We compare against point-wise rerankers (MonoT5 (Nogueira et al., 2020), BGE (Chen et al., 2024), and Qwen3-Reranker (Zhang et al., 2025)), list-wise rerankers (RankVicuna (Pradeep et al., 2023a), RankZephyr (Pradeep et al., 2023b), and RankGPT-style (Sun et al., 2023)), and the closest set-wise ones, Set-Encoder (Schlatt et al., 2025) and SETR (Lee et al., 2025). Our sampling-based and amortized variants are denoted by SHAPRANK-S and SHAPRANK-A, respectively.

Datasets & Metrics. We evaluate on five multi-hop QA benchmarks: MuSiQue (Trivedi et al., 2022), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MultiHopRAG (Tang & Yang, 2024), and MorehopQA (Schnitzler et al., 2024). These benchmarks stress the selection of complementary evidence across documents. We measure reranking quality with Recall@k (R@k), defined as the fraction of gold supporting documents in the top-k results, and downstream answer quality with EM/F1 (Gutiérrez et al., 2024; 2025).

Settings. We use Qwen3-Embedding-0.6B for base dense passage retrieval (DPR) as the first-stage retriever. From the top 200 retrieved documents, we rerank a pool of 30 without injecting gold documents. We finetune Qwen3-Reranker-0.6B and Qwen3-Reranker-4B as the set-wise valuer v_θ, and train the 0.4B estimator ψ_ω on the same 10k MuSiQue training examples. Both models are evaluated on all five test sets. Unless otherwise specified, coalition sizes are drawn from $\{1,2,3,4\}$ and $T = 200$. Downstream results are reported with Qwen3.5-Flash and DeepSeek-V3.2. Full baseline descriptions and implementation details [1] are provided in Appendix B.

6 Experimental Results

6.1 Main Reranking Comparison

Table 1 reports Recall@5/@10 across five benchmarks using the Qwen3-Reranker-0.6B set-wise valuer, while Table 7 reports the corresponding 4B results. All methods rerank the same candidate pool, and both v_θ and ψ_ω are trained only on MuSiQue; results on the other four datasets therefore measure zero-shot generalization. We draw three conclusions. **1) Set-wise valuation with**

[1] We will release our code and pretrained models upon acceptance.

Table 1: Reranking quality (Recall@5 / Recall@10) across five benchmarks. Best result in each column is in **bold**; second best is underlined. "Avg." averages the metric across all datasets. Entries of the form $a \pm b$ report the mean and standard deviation over 3 random seeds; entries without $\pm$ are deterministic given the trained model.

Method	#Params	MuSiQue		HotpotQA		2WikiMQA		MultiHopRAG		MorehopQA		Avg.	
		R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10
No reranking (Qwen3-Embedding)	-	55.75	63.98	80.40	87.70	69.85	72.90	69.43	82.52	82.95	87.00	71.67	78.82
Point-wise rerankers													
MonoT5	0.22B	57.58	66.54	85.70	92.45	71.73	74.80	76.06	86.41	88.55	92.20	75.92	82.48
BGE-v2-m3	0.57B	61.77	68.62	92.75	94.80	71.20	74.05	82.20	90.40	79.70	82.95	77.52	82.16
BGE-reranker-large	0.6B	59.45	66.57	91.55	94.45	70.67	72.82	73.82	85.42	74.40	78.05	73.97	79.46
Qwen3-Reranker-0.6B	0.6B	60.22	68.39	91.00	95.10	72.58	75.58	**84.40**	**93.54**	86.75	90.35	78.99	84.59
List/Set-wise rerankers													
RankVicuna	7B	54.92	62.06	86.00	91.90	72.78	74.72	69.12	82.32	87.25	88.50	74.01	79.90
RankZephyr	7B	58.09	65.46	87.65	92.45	71.98	74.92	64.16	76.67	89.80	90.75	73.37	79.26
RankGPT-style (Qwen3-8B)	8B	57.81	61.54	87.60	89.40	**77.65**	**78.73**	78.07	84.73	**92.85**	93.85	78.79	81.65
Set-Encoder	0.3B	56.42	61.89	88.20	90.26	71.76	74.23	66.34	85.28	87.15	92.18	73.97	80.76
SETR	8B	57.38	62.49	90.25	93.28	72.02	75.39	67.27	86.08	89.13	92.49	75.21	81.94
Trained on the same 10k MuSiQue data													
Qwen3-Reranker-0.6B	0.6B	60.52	67.92	89.25	92.65	69.67	72.62	83.72	90.94	77.85	79.90	76.20	80.80
Set-Encoder	0.3B	58.20	63.72	88.04	90.01	71.52	74.42	66.04	84.98	85.38	91.15	73.83	80.85
Ours (v_θ is set-wise finetuned from Qwen3-Reranker-0.6B)													
SHAPRANK-S	0.6B	66.81$_{\pm 0.09}$	**74.09**$_{\pm 0.14}$	92.95$_{\pm 0.15}$	95.60$_{\pm 0.00}$	73.20$_{\pm 0.06}$	77.00$_{\pm 0.08}$	84.01$_{\pm 0.01}$	92.89$_{\pm 0.15}$	88.15$_{\pm 1.10}$	92.10$_{\pm 0.72}$	81.02$_{\pm 0.28}$	86.33$_{\pm 0.21}$
SHAPRANK-A	0.4B	**67.07**$_{\pm 0.02}$	73.50$_{\pm 0.04}$	**93.10**$_{\pm 0.05}$	**95.95**$_{\pm 0.06}$	73.75$_{\pm 0.03}$	77.55$_{\pm 0.06}$	81.39$_{\pm 0.03}$	90.10$_{\pm 0.08}$	91.35$_{\pm 0.11}$	**94.80**$_{\pm 0.12}$	**81.33**$_{\pm 0.05}$	**86.38**$_{\pm 0.07}$

Shapley-guided attribution achieves the best overall reranking quality at both scales. With the 0.6B backbone, SHAPRANK-S and SHAPRANK-A reach 81.02/86.33 and 81.33/86.38 average R@5/R@10, respectively, outperforming the strongest point-wise baseline (78.99/84.59) and the 8B RankGPT-style baseline (78.79/81.65). The same ordering holds at 4B scale, where SHAPRANK-A obtains the best average result of 83.36/87.91. Table 1 also includes the controlled Qwen3 point-wise baseline and the controlled Set-Encoder baseline trained with the same data and optimization budget, which achieve only 76.20/80.80 and 73.83/80.85. Thus, in-domain finetuning alone does not explain the gains; the key differences are coalition-level utility modeling and document-level credit aggregation. **2) The gains are largest on benchmarks that emphasize cross-document interaction.** On MuSiQue, our variants improve R@5 from 60.52 for the controlled point-wise model to 66.81–67.07, while on MorehopQA SHAPRANK-A improves over the strongest point-wise reranker from 86.75 to 91.35. In contrast, the margin is smaller on HotpotQA, where a single strong passage may suffice more often. These results support the hypothesis that subset-level utility is most useful when answering requires complementary evidence. MultiHopRAG is the main exception, possibly because greater redundancy among supporting passages reduces the benefit of marginal-credit aggregation; directly measuring evidence substitutability is left to future work. **3) Amortization preserves the quality of sampling-based attribution.** The 0.4B amortized estimator matches or exceeds its sampling-based teacher on average, achieving 81.33 versus 81.02 R@5 at 0.6B and 83.36 versus 82.57 at 4B. It also substantially outperforms the teacher on MorehopQA at 4B (93.70 versus 89.35 R@5; Table 7). This resonates with literature results indicating that amortized attribution may smooth the variation of per-query estimation under the finite sampling budget (Jethani et al., 2022).

6.2 Downstream RAG Answer Quality

Table 2 reports end-to-end answer quality: each method selects the top-m reranked documents as context for a fixed downstream LLM, and we measure EM/F1 of the generated answers, so any difference is attributable to the context. We draw three conclusions. **1) Reranking gains translate into better answers.** Our amortized variant achieves the best average EM/F1 (46.25/51.99), improving over the strongest list-wise baseline (RankGPT-style with Qwen3-8B, 45.24/50.49) by $+1.0$/$+1.5$ and over the same-backbone point-wise reranker (Qwen3-Reranker-0.6B, 44.54/50.10) by $+1.7$/$+1.9$, while the sampling-based variant ranks second on average (45.43/51.16); all rerankers clearly outperform the no-reranking baseline (40.43/45.45), confirming the importance of the reranking stage itself. **2) The advantage is most pronounced where evidence must be composed across documents.** On MuSiQue, the most compositional benchmark, our two variants reach 34.2–34.8 EM and 44.73–45.25 F1, exceeding the best baseline (SETR, 32.8/42.91) by up to $+2.0$ EM / $+2.3$ F1, which is a wide margin under the generation scenario, and on MorehopQA the amortized estimator attains the best EM/F1 (46.51/46.84). The improvement on MuSiQue is larger than on HotpotQA. This asymmetry indicates that a complementary evidence set benefits generation more

Table 2: Downstream RAG answer quality (EM / F1) using the top-m documents, with a fixed downstream LLM (Qwen3.5-Flash, $m = 5$). Best result is in **bold**; second best is underlined.

Method	#Params	MuSiQue		HotpotQA		2WikiMQA		MultiHopRAG		MorehopQA		Avg.	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
No reranking (Qwen3-Embedding)	-	27.60	35.90	50.70	62.60	52.80	56.94	31.10	31.55	39.98	40.29	40.43	45.45
Point-wise rerankers													
MonoT5	0.22B	31.1	39.42	56.8	68.43	53.4	58.36	32.59	33.11	43.20	43.47	43.41	48.55
BGE-v2-m3	0.57B	30.6	40.37	**61.4**	**74.10**	53.7	58.09	33.88	34.32	37.65	37.92	43.44	48.96
BGE-reranker-large	0.6B	28.8	38.79	61.1	72.93	53.1	57.12	31.79	32.37	33.63	33.86	41.68	47.01
Qwen3-Reranker-0.6B	0.6B	31.7	41.36	60.8	73.19	54.1	59.27	34.01	34.22	42.12	42.47	44.54	50.10
List/Set-wise rerankers													
RankVicuna	7B	31.0	40.10	56.7	69.66	54.5	54.50	31.35	31.94	42.12	42.51	43.13	47.74
RankZephyr	7B	30.0	38.79	55.5	69.19	54.5	54.50	29.71	30.16	44.72	44.96	42.88	47.52
RankGPT-style (Qwen3-8B)	8B	31.2	39.60	58.5	71.01	**56.7**	**61.16**	33.50	33.90	46.33	46.78	45.24	50.49
Set-Encoder	0.3B	32.3	41.4	58.1	71.5	53.3	58.21	33.23	33.90	42.49	44.12	43.88	49.82
SETR	8B	32.8	42.91	58.82	72.03	53.03	59.21	32.12	33.58	43.21	44.81	43.99	50.50
Ours (set-wise)													
SHAPRANK-S	0.6B	34.2	44.73	60.8	73.43	53.9	58.70	**34.10**	**34.62**	44.19	44.33	45.43	51.16
SHAPRANK-A	0.4B	**34.8**	**45.25**	60.9	73.40	55.9	60.74	33.13	33.71	**46.51**	**46.84**	**46.25**	**51.99**

on heavier combinatorial scenarios, which is precisely the regime our subset-level objective targets. **3) Answer quality tracks set composition rather than recall alone.** The relative ordering of methods is largely preserved from Sec. 6.1, but not identically: the amortized estimator, whose recall is only marginally higher than the sampling-based variant, opens a clearer downstream gap (46.25 vs. 45.43 average EM), and list-wise rerankers with 7B parameters (RankVicuna, RankZephyr) fall behind our 0.4–0.6B models by 2.3–3.4 average EM. Both observations suggest that what the generator ultimately rewards is a well-composed context, which per-document relevance scores and holistic permutations capture only indirectly. Table 9 repeats this evaluation with a different LLM and shows consistent trends, indicating that the gains are not tied to a particular LLM.

6.3 Efficiency Analysis

Figures 2 and 5 plot R@5 against per-query reranking latency (log scale) for all methods on each dataset and on average; all timings are measured under the same hardware. We draw three conclusions. **1) The amortized estimator occupies the desirable high-quality, low-cost region.** On average it attains the best R@5 (81.33) at 0.68s per query, i.e., point-wise latency: it is on par with the fastest cross-encoders (BGE-large, 0.65s; BGE-v2-m3, 0.67s) while outperforming them by $+7.4$ and $+3.8$ R@5, and it is $3.2\times$ faster than its own backbone Qwen3-Reranker-0.6B (2.18s, 78.99) with $+2.3$ higher quality. Compared with its sampling-based teacher (T=200, 4.87s), it delivers slightly better quality at a $7.2\times$ speedup, so amortization removes essentially all of the sampling-estimation overhead without sacrificing accuracy. **2) Sampling-based attribution is accurate but pays a moderate, controllable cost.** Its latency grows with the coalition budget, yet quality saturates quickly: reducing T from 200 to 100 halves the cost (4.87s $\rightarrow$ 2.44s) while average R@5 drops only marginally (81.02 $\rightarrow$ 80.98), indicating that a modest budget already yields stable estimates and that the quality–latency trade-off can be tuned smoothly at deployment time. **3) List-wise rerankers are dominated on both axes.** RankVicuna and RankZephyr require 9.9s and 14.4s per query, i.e., 15–21$\times$ slower than our amortized estimator, while achieving markedly lower average quality. Their autoregressive decoding of a permutation over long inputs makes them the slowest methods in every panel without a corresponding quality benefit. The same picture holds per dataset, with the largest latency gaps on MorehopQA, whose longer documents inflate list-wise decoding time to 22.6s (RankZephyr) versus 0.95s for the amortized estimator. The set-wise baselines lag behind on performance as well. Overall, the amortized estimator turns a principled but sampling-heavy attribution procedure into a deployable reranker whose cost matches standard point-wise models while retaining the quality of explicit sampling-based attribution and on average slightly exceeding it.

6.4 Ablation Study

We examine whether the gains arise from set-wise contrastive training, sampling-based attribution, or their combination. All variants use the same Qwen3-Reranker-0.6B backbone;

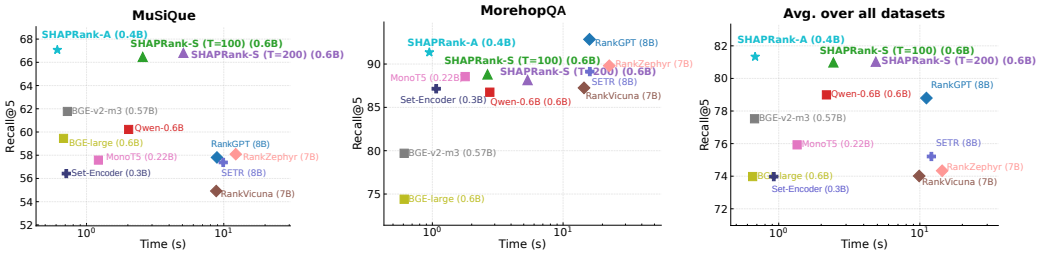


Figure 2: Reranking quality (Recall@5) versus per-query latency (log scale) on MuSiQue, MorehopQA, and averaged over all five benchmarks. Marker area is proportional to the number of parameters; methods closer to the upper-left corner are better. All timings are measured on the same single A100 GPU with batch size 1 and averaged over 500 queries.

the point-wise counterpart is trained with the same data, margin loss, and optimization budget but scores documents individually. Figure 3 reports results of the comparison averaged across the five benchmarks, with per-dataset results provided in Tables 5, 6, and 8.

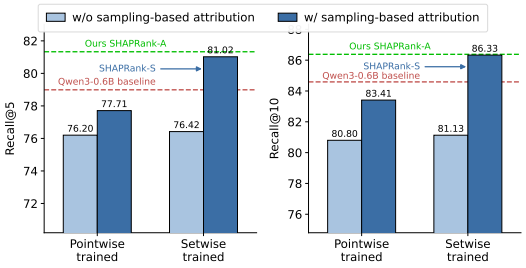


Figure 3: Average results across all datasets, comparing point-wise and set-wise training with and without sampling-based attribution. Green line indicates ours SHAPRANK-A, and the second bar in *Setwise trained* group is our SHAPRANK-S.

The *full method* (SHAPRANK-*S*) achieves 81.02/86.33 average R@5/R@10. Removing attribution and ranking documents directly by their singleton utilities reduces performance to 76.42/81.13, whereas applying attribution to the point-wise-trained model yields only 77.71/83.41. With both components removed, performance further decreases to 76.20/80.80. Neither component alone outperforms the off-the-shelf Qwen3-Reranker-0.6B (78.99/84.59); the improvement emerges only when they are combined. These results support the intended division of labor: set-wise training produces meaningful utilities over document coalitions, while attribution aggregates context-dependent marginal advantages into a reliable document ranking. Additional sensitivity analyses of the coalition-size range and sampling budget are presented in Appendix C.1 (Figure 4).

7 LIMITATIONS

First, our theoretical guarantee concerns the relationship between the learned valuer's local comparisons and the semivalue ordering they induce; it does not guarantee that the valuer itself is a faithful characteristic function relative to gold relevance, only that violations of local desirability are outvoted in expectation. Second, our budget-restricted attribution method is evaluated with sampling subset sizes of at most four documents; we have not tested how the approach scales w.r.t the downstream context budget. Third, our evaluation is restricted to English multi-hop QA benchmarks chosen because they stress complementary evidence. Finally, our deployed pipeline requires training two models, which adds engineering overhead.

8 CONCLUSION

In this paper, we cast reranking as a cooperative game and introduced a set-wise valuer over document coalitions. We derived a local-comparison principle for semivalue ranking and instantiated it in practice with a restricted KernelSHAP-inspired regression attribution rule. We further amortized this attribution rule into a single-pass estimator. Across five multi-hop QA benchmarks, the resulting rerankers improve both retrieval recall and downstream answer quality, while the amortized variant retains most of the attribution benefit at the inference cost of a standard neural reranker.

9 USE OF GENERATIVE AI

We used LLMs solely to improve the readability of the manuscript, including polishing wording, grammar, and phrasing of text that was originally written by the authors. Generative AI tools were not used for any task requiring disclosure under the ICLR policy. Specifically, they did not contribute to research ideation, methodology or experimental design, theoretical developments, etc. All AI-assisted edits were reviewed by the authors to ensure that the technical content and claims remained accurate and unchanged. We take full responsibility for the final content of this work.

REFERENCES

Alibaba Cloud. Qwen3.5-flash. Alibaba Cloud Model Studio, 2026. Qwen3.5-Flash model documentation.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pp. 89–96, 2005.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pp. 129–136, 2007.

Abraham Charnes, Boaz Golany, M Keane, and J Rousseau. Extremal principle solutions of games in characteristic function form: core, chebychev and shapley value generalizations. In *Econometrics of planning and efficiency*, pp. 123–133. Springer, 1988.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 4(5), 2024.

Yiqun Chen, Qi Liu, Yi Zhang, Weiwei Sun, Xinyu Ma, Wei Yang, Daiting Shi, Jiaxin Mao, and Dawei Yin. Tourrank: Utilizing large language models for documents ranking with a tournament-inspired strategy. In *Proceedings of the ACM on Web Conference 2025*, pp. 1638–1652, 2025.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 719–729, 2024.

DeepSeek-AI. Deepseek-v3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Luyu Gao and Jamie Callan. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2843–2853, 2022.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. Re2g: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2701–2715, 2022.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*, 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

John R Isbell. A class of simple games. 1958.

Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. Fastshap: Real-time shapley value estimation. In *International conference on learning representations*, 2022.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Dahyun Lee, Yongrae Jo, Haeju Park, and Moontae Lee. Shifting from ranking to set selection for retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17606–17619, 2025.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. Zero-shot listwise document reranking with a large language model, 2023. *URL https://arxiv. org/abs/2305.02156*, 2023.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the association for computational linguistics: EMNLP 2020*, pp. 708–718, 2020.

Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. Setrank: Learning a permutation-invariant ranking model for information retrieval. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 499–508, 2020.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667*, 2021.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*, 2023a.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze! *arXiv preprint arXiv:2312.02724*, 2023b.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, et al. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 1504–1518, 2024.

Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc, 2009.

Ferdinand Schlatt, Maik Fröbe, Harrisen Scells, Shengyao Zhuang, Bevan Koopman, Guido Zuccon, Benno Stein, Martin Potthast, and Matthias Hagen. Set-encoder: Permutation-invariant inter-passage attention for listwise passage re-ranking with cross-encoders. In *European Conference on Information Retrieval*, pp. 1–19. Springer, 2025.

Julian Schnitzler, Xanh Ho, Jiahao Huang, Florian Boudin, Saku Sugawara, and Akiko Aizawa. Morehopqa: More than multi-hop reasoning. *arXiv preprint arXiv:2406.13397*, 2024.

Lloyd S Shapley. A value for n-person games. 1953.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 14918–14937, 2023.

Raphael Tang, Crystina Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Türe. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2327–2340, 2024.

Yixuan Tang and Yi Yang. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547, 2025.

Eyal Winter. The shapley value. *Handbook of game theory with economic applications*, 3:2025–2054, 2002.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pp. 1192–1199, 2008.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. Large language models for information retrieval: A survey. *ACM Transactions on Information Systems*, 44(1):1–54, 2025.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 38–47, 2024.

A THEORETICAL DERIVATION

A.1 PROOF OF THE SEMIVALUE GAP THEOREM

Proof. For brevity, write $v(S) = v_\theta(q, S)$ and extend the weight family by setting $w(k) = 0$ for $k > B$. From Definition 1,

$$\phi_i^{(q),B} - \phi_j^{(q),B} = \sum_{T \subseteq C_q \setminus \{d_i\}} w(|T|)\big(v(T \cup \{d_i\}) - v(T)\big) - \sum_{T \subseteq C_q \setminus \{d_j\}} w(|T|)\big(v(T \cup \{d_j\}) - v(T)\big).$$

For every $S \subseteq C_q \setminus \{d_i, d_j\}$, group the terms whose coalitions exclude the other distinguished document and those whose coalitions include it. The common utility terms cancel, giving

$$\phi_i^{(q),B} - \phi_j^{(q),B} = \sum_{S \subseteq C_q \setminus \{d_i, d_j\}} \big(w(|S|) + w(|S|+1)\big)\big(v(S \cup \{d_i\}) - v(S \cup \{d_j\})\big).$$

Define $u(S) = w(|S|) + w(|S|+1)$, then using Pascal's identity and $w(B+1) = 0$, we have

$$\begin{aligned}\sum_{S \subseteq C_q \setminus \{d_i, d_j\}} u(S) &= \sum_{k=0}^{B} \binom{n-2}{k} w(k) + \sum_{k=1}^{B} \binom{n-2}{k-1} w(k) \\ &= \sum_{k=0}^{B} \binom{n-1}{k} w(k) = 1.\end{aligned}$$

Hence u is a probability distribution, and therefore

$$\phi_i^{(q),B} - \phi_j^{(q),B} = \mathbb{E}_{S \sim u}\left[v_\theta(q, S \cup \{d_i\}) - v_\theta(q, S \cup \{d_j\})\right].$$

The ordering claim follows immediately. □

A.2 REGRESSION SCORE GAP ANALYSIS UNDER RESTRICTED COALITION REGRESSION

We first define the population version of the budget-restricted regression. Let $\widetilde{v}(S) = v_\theta(q, S) - v_\theta(q, \emptyset)$ denote the centered coalition utility. For each coalition size $1 \le k \le B$, let $a_k > 0$ be the regression weight assigned to every coalition of size k. We define the population regression coefficients as

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^n} \sum_{\substack{S \subseteq C_q \\ 1 \le |S| \le B}} a_{|S|} \left(\widetilde{v}(S) - \sum_{d_\ell \in S} \beta_\ell\right)^2.$$

Theorem 3 (Regression score gap)**.** *For any two distinct documents $d_i, d_j \in C_q$,*

$$\beta_i^* - \beta_j^* = \mathbb{E}_{S \sim u_{\mathrm{reg}}}\left[v_\theta(q, S \cup \{d_i\}) - v_\theta(q, S \cup \{d_j\})\right],$$

where u_{reg} is a probability distribution over $S \subseteq C_q \setminus \{d_i, d_j\}$ with $|S| \le B-1$, given by $u_{\mathrm{reg}}(S) = a_{|S|+1}/D_B$, with $D_B = \sum_{r=0}^{B-1} \binom{n-2}{r} a_{r+1}$.

Proof. For each coalition $S \subseteq C_q$, let $\mathbf{z}_S \in \{0,1\}^n$ denote its indicator vector. The normal equations are $M\boldsymbol{\beta}^* = \mathbf{b}$, where

$$M = \sum_{\substack{S \subseteq C_q \\ 1 \le |S| \le B}} a_{|S|} \mathbf{z}_S \mathbf{z}_S^\top, \qquad \mathbf{b} = \sum_{\substack{S \subseteq C_q \\ 1 \le |S| \le B}} a_{|S|} \mathbf{z}_S \widetilde{v}(S).$$

Because coalition weights depend only on cardinality, all diagonal entries of M are equal, as are all off-diagonal entries. In particular,

$$M_{ii} = \sum_{k=1}^{B} \binom{n-1}{k-1} a_k =: A, \qquad M_{ij} = \sum_{k=2}^{B} \binom{n-2}{k-2} a_k =: C \quad (i \ne j).$$

Therefore $M = (A-C)I + C\mathbf{1}\mathbf{1}^\top$. Because $(\mathbf{e}_i - \mathbf{e}_j)^\top \mathbf{1} = 0$, left-multiplying the normal equations by $(\mathbf{e}_i - \mathbf{e}_j)^\top$ yields

$$(A - C)(\beta_i^* - \beta_j^*) = b_i - b_j.$$

Applying Pascal's identity $\binom{n-1}{k-1} = \binom{n-2}{k-1} + \binom{n-2}{k-2}$, with out-of-range binomial coefficients taken to be zero, gives

$$A - C = \sum_{k=1}^{B} \binom{n-2}{k-1} a_k = \sum_{r=0}^{B-1} \binom{n-2}{r} a_{r+1} = D_B.$$

We next simplify the difference $b_i - b_j$. By definition,

$$b_i - b_j = \sum_{\substack{T \subseteq C_q \\ 1 \le |T| \le B}} a_{|T|} \big(\mathbf{1}\{d_i \in T\} - \mathbf{1}\{d_j \in T\}\big) \widetilde{v}(T).$$

Coalitions containing both d_i and d_j cancel, while coalitions containing neither contribute zero. Every remaining coalition can be paired uniquely as $S \cup \{d_i\}$ and $S \cup \{d_j\}$ for $S \subseteq C_q \setminus \{d_i, d_j\}$ with $|S| \le B - 1$. Hence

$$b_i - b_j = \sum_{\substack{S \subseteq C_q \setminus \{d_i, d_j\} \\ |S| \le B-1}} a_{|S|+1} \left[\widetilde{v}(S \cup \{d_i\}) - \widetilde{v}(S \cup \{d_j\})\right].$$

The common baseline $v_\theta(q, \emptyset)$ cancels in each difference, so

$$b_i - b_j = \sum_{\substack{S \subseteq C_q \setminus \{d_i, d_j\} \\ |S| \le B-1}} a_{|S|+1} \left[v_\theta(q, S \cup \{d_i\}) - v_\theta(q, S \cup \{d_j\})\right].$$

Combining this expression with $(A - C)(\beta_i^* - \beta_j^*) = b_i - b_j$ and $A - C = D_B$ gives

$$\beta_i^* - \beta_j^* = \frac{1}{D_B} \sum_{\substack{S \subseteq C_q \setminus \{d_i, d_j\} \\ |S| \le B-1}} a_{|S|+1} \left[v_\theta(q, S \cup \{d_i\}) - v_\theta(q, S \cup \{d_j\})\right].$$

Finally, the weights are nonnegative and satisfy

$$\sum_{\substack{S \subseteq C_q \setminus \{d_i, d_j\} \\ |S| \le B-1}} u_{\text{reg}}(S) = \frac{1}{D_B} \sum_{r=0}^{B-1} \binom{n-2}{r} a_{r+1} = 1.$$

Thus u_{reg} is a probability distribution, and the preceding weighted sum is exactly the stated expectation. □

Corollary 4 (Ordering under local desirability)**.** *Under the conditions of Theorem 3, if*

$$v_\theta(q, S \cup \{d_i\}) > v_\theta(q, S \cup \{d_j\})$$

for every S in the support of u_{reg}, then

$$\beta_i^* > \beta_j^*.$$

More generally, the same ordering holds whenever the local advantage is positive in expectation under u_{reg}.

For the restricted KernelSHAP sampling rule in Sec. 4.2, each coalition of size k receives population weight proportional to $k(n, k)$. The common normalization constant does not affect the regression solution, so Theorem 3 applies with

$$a_k \propto k(n, k).$$

Consequently,

$$u_{\text{reg}}(S) \propto k(n, |S| + 1).$$

Thus, sampling coalitions of sizes $1, \ldots, B$ induces a regression score gap that averages local document advantages over contexts of sizes $0, \ldots, B-1$.

Finally, let $\widehat{\boldsymbol{\beta}}_T$ denote the least-squares solution obtained from T independently sampled coalitions from the restricted distribution in Sec. 4.2. Since the coalition space is finite and the population design matrix above is positive definite, the empirical normal equations converge to their population counterparts. Therefore,

$$\widehat{\boldsymbol{\beta}}_T \longrightarrow \boldsymbol{\beta}^*$$

almost surely as $T \to \infty$. In particular,

$$\widehat{\beta}_{T,i} - \widehat{\beta}_{T,j} \longrightarrow \mathbb{E}_{S \sim u_{\text{reg}}} \left[v_\theta(q, S \cup \{d_i\}) - v_\theta(q, S \cup \{d_j\}) \right].$$

B Experimental and Implementation Details

B.1 Baselines

We compare against representative rerankers covering different interaction granularities.

Point-wise rerankers. MonoT5 (Nogueira et al., 2020) is a sequence-to-sequence reranker that scores each query–document pair independently. We also evaluate two families of strong cross-encoders: BGE-reranker-large (0.6B) and BGE-Reranker-v2-m3 (0.57B) from the BGE family (Chen et al., 2024), and Qwen3-Reranker-0.6B and Qwen3-Reranker-4B from the Qwen family (Zhang et al., 2025). These models span approximately 0.2B to 4B parameters, allowing us to distinguish gains due to our method from gains due solely to model capacity.

List-wise rerankers. We include RankVicuna (Pradeep et al., 2023a) and RankZephyr (Pradeep et al., 2023b), two 7B LLM-based rerankers that jointly process multiple candidates and generate an ordering. We additionally implement a RankGPT-style baseline (Sun et al., 2023) using Qwen3-8B as its backbone.

Set-wise rerankers. The systems closest to our setting are Set-Encoder (Schlatt et al., 2025), a 0.3B cross-encoder with permutation-invariant inter-passage attention, and SETR (Lee et al., 2025), an 8B LLM-based set-wise reranker. Our sampling-based and amortized methods are denoted by SHAPRANK-S and SHAPRANK-A, respectively.

B.2 Datasets and Metrics

We evaluate on five retrieval-augmented multi-hop question answering benchmarks: MuSiQue (Trivedi et al., 2022), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MultiHopRAG (Tang & Yang, 2024), and MorehopQA (Schnitzler et al., 2024). These benchmarks generally require combining evidence from multiple documents rather than relying on a single high-scoring passage, making them suitable for evaluating interactions among retrieved candidates.

Following prior work (Gutiérrez et al., 2024; 2025), we evaluate reranking with Recall@k and downstream answer generation with exact match (EM) and token-level F1. For a query, Recall@k is the fraction of its gold supporting documents contained in the top-k reranked candidates. We report the metric averaged over all evaluation queries.

B.3 Evaluation Protocol

The first-stage retriever returns the top 200 documents for each query. We take the first 30 documents as the candidate pool without injecting missing gold documents, ensuring that all rerankers operate on the same realistic retrieval output.

We instantiate the set-wise valuer v_θ by finetuning Qwen3-Reranker-0.6B and Qwen3-Reranker-4B. The amortized estimator ψ_ω contains approximately 0.4B parameters. Both v_θ and ψ_ω are trained using the same 10k examples drawn exclusively from the MuSiQue training set. We then evaluate

the trained models on the test sets of all five benchmarks; performance on the other four datasets therefore measures cross-dataset generalization.

Unless otherwise specified, sampled coalition sizes are drawn from $\{1, 2, 3, 4\}$ and the sampling-based method uses $T = 200$ coalitions per query. The downstream generator receives the top-ranked documents selected by each reranker. We evaluate downstream answer quality with both Qwen3.5-Flash and DeepSeek-V3.2 to verify that the observed trends are not specific to a single generator.

B.4 Model Implementation and Training

Scoring a coalition with the valuer. We instantiate v_θ on top of Qwen3-Reranker without modifying its interface. Given a query q and a coalition S, we concatenate the documents of S into a single passage field and feed the result to the official Qwen3-Reranker prompt template, with the documents shuffled uniformly as described in Sec. 4.1. We then read the logits of the `yes` and `no` tokens at the last position, apply a softmax over this pair, and take the resulting `yes` probability as the coalition utility $v_\theta(q, S) \in (0, 1)$.

Training tuple construction. For each training query, we construct training tuples (q, S, d^+, d^-) from the retrieved candidate pool. A positive document d^+ is sampled from the gold supporting documents contained in the candidate pool, while the negative document d^- is sampled from the top-retrieved non-supporting candidates, which serves as hard-negative sampling. The context coalition S is then randomly sampled from the remaining candidate documents, with d^+ and d^- explicitly excluded from S. Rather than sampling coalition sizes uniformly, we sample the size of S according to the cardinality probability mass induced by the theoretical coalition weighting used in our restricted semivalue formulation. In the default configuration, the allowed coalition sizes are $\{1, 2, 3, 4\}$. For each query, we independently sample three tuples per training epoch, and the tuples are resampled across epochs.

Hyperparameters. Both v_θ and ψ_ω are optimized with AdamW using a learning rate of 1×10^{-4}, a batch size of 16, and 3 epochs over the 10k training queries. The margin of the set-wise contrastive objective is set to $m = 0.3$ by default. Unless otherwise specified, coalition sizes are restricted to $\{1, 2, 3, 4\}$. For each query, three independently sampled training tuples are used in each epoch. At inference, the downstream generator receives the top $m = 5$ reranked documents unless stated otherwise. Downstream results are reported with Qwen3.5-Flash Alibaba Cloud (2026) and DeepSeek-V3.2 DeepSeek-AI (2025).

Compute. All training fits within the memory of a single A100 GPU. In practice we train on $4\times$ A100 GPUs, which takes roughly 2 hours for the set-wise valuer v_θ and roughly 1 hour for the amortized estimator ψ_ω. Both are trained once on the same 10k MuSiQue samples and reused across all five benchmarks, so this cost is paid a single time and amortized over every query at deployment.

C Extended Experimental Results

C.1 Extended Ablation Study

Is the gain due to the valuer or to the aggregation? A set-wise valuer admits a more direct use than attribution: one can simply search for the best context, $\arg\max_{|S|=m} v_\theta(q, S)$. Table 3 compares this greedy alternative against semivalue ranking under *identical* model weights, so any difference is attributable to the scoring rule alone. Greedy maximization reaches 78.37 average R@5, improving over the single-document readout (76.42) by +2.2 and confirming that the set-wise utility carries genuine cross-document signal that point-wise readout discards. It nevertheless trails regression-attribution ranking (81.02) by 2.65, at a comparable number of valuer calls ($nm = 150$ versus $T = 200$). We attribute the gap to two properties of the attribution view. First, greedy search commits to a single trajectory and inherits the valuer's error on every step it takes, whereas the attribution rule aggregates evaluations over many contexts, which is exactly the relaxation from *correct on every context* to *correct on average* identified in Sec. 4.1. Second, greedy search returns a set but no per-document score, so it cannot produce a ranking, cannot serve budgets m other than the one it optimized, and offers no attribution to inspect; the semivalue yields all three at no extra

Table 3: Alternative ways of using the *same* trained set-wise valuer v_θ (Qwen3-Reranker-0.6B backbone). All rows share identical model weights and differ only in how candidates are scored from v_θ; the gap between them therefore isolates the contribution of semivalue aggregation from that of set-wise training. Per-dataset numbers are Recall@5. Latency is per-query wall-clock on a single A100, batch size 1, averaged over 500 queries.

Scoring rule from v_θ	#Fwd.	Recall@5					Avg.	
		MuSiQue	HotpotQA	2WikiMQA	MultiHopRAG	MorehopQA	R@5	R@10
Single-document readout $v_\theta(q, \{d_i\})$	n	63.20	90.05	70.13	81.02	77.70	76.42	81.13
Greedy set maximization (m=5)	nm	65.01	91.52	71.08	82.15	82.10	78.37	82.81
Greedy set maximization (beam =5)	$5nm$	66.01	92.30	72.31	83.12	85.06	79.76	84.04
Restricted attribution, sampling (T=200)	T	66.81	92.95	73.20	84.01	88.15	81.02	86.33
Restricted attribution, amortized (ψ_ω)	1	67.07	93.10	73.75	81.39	91.35	81.33	86.38

Table 4: What the amortized estimator learns. All three rows use the identical 0.4B ModernBERT architecture, the same 10k MuSiQue queries and the same optimization budget, and differ only in the regression target; the teacher v_θ is the same set-wise trained valuer throughout. Per-dataset numbers are Recall@5.

Regression target for ψ_ω	Supervision	Recall@5					Avg.	
		MuSiQue	HotpotQA	2WikiMQA	MultiHopRAG	MorehopQA	R@5	R@10
Gold relevance label	annotations	62.51	90.53	70.54	82.55	81.01	77.43	82.03
$v_\theta(q, \{d_i\})$	single-doc utility	63.53	90.22	70.51	81.53	79.54	77.06	81.82
Regression attribution via $\mathcal{L}_{\text{est}}$ (ours)	coalition utility	67.07	93.10	73.75	81.39	91.35	81.33	86.38

cost. Beam search recovers 1.39 points of this gap, but still trails sampling-based attribution by 1.26 points.

Is the amortized estimator merely an extra small model? Table 4 holds the 0.4B ModernBERT architecture, the training data and the optimization budget fixed and varies only the regression target. When trained on gold relevance labels, the same architecture reaches 77.4 average R@5. When trained to reproduce the single-document utilities $v_\theta(q, \{d_i\})$ it reaches 77.1, both fall short of the 81.33 obtained with the coalition-level objective $\mathcal{L}_{\text{est}}$. The advantage of ψ_ω therefore does not come from adding a dedicated compact reranker, nor from distilling the valuer's per-document scores, but from the target it regresses: because $\mathcal{L}_{\text{est}}$ supervises ψ_ω to explain the utility of *sampled coalitions* additively, the network is exposed to the same cross-document evidence that the sampling pipeline integrates, and compresses it into one forward pass. This also explains why the amortized model can match and occasionally exceed its sampling-based counterpart (Sec. 6.1): it is trained across far more coalitions than any single query's budget of T samples affords, and thereby averages out Monte Carlo noise that per-query sampling cannot.

What about the sensitivity to coalition size and sampling budget? Figure 4 studies two inference-time hyperparameters: the range of sampled coalition sizes and the number of sampled coalitions T. Expanding the allowed sizes from $\{1, 2\}$ to $\{1, 2, 3\}$ consistently improves both R@5 and R@10 across sampling budgets, indicating that contexts containing three documents provide useful interaction information beyond singleton and pair contexts. Extending the range further to $\{1, 2, 3, 4\}$ yields little additional improvement, suggesting that the relevant higher-order interactions can already be captured with relatively small coalitions.

Performance also improves substantially when T increases from 50 to 100, but largely plateaus between $T = 100$ and $T = 200$. This trend is consistent with the small difference between the $T = 100$ and $T = 200$ variants in the efficiency analysis (Section 6.3). Overall, sampling approximately 100–150 coalitions provides a favorable quality–efficiency trade-off, while our default $T = 200$ serves as a conservative setting for the main experiments.

C.2 Further Analysis

Does the valuer actually learn set-level utility? We probe this directly: for each dataset we sample 20,000 pairs of document sets (S^+, S^-), where S^+ contains strictly more supporting documents than S^-, and measure the *satisfaction rate*, i.e., the fraction of pairs on which $v_\theta(S^+) > v_\theta(S^-)$;

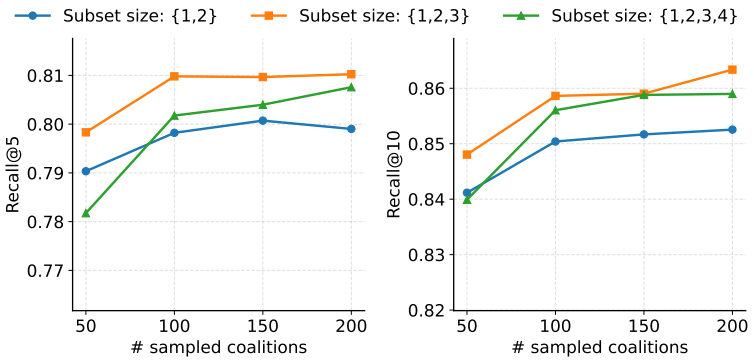


Figure 4: Sensitivity of average Recall@5 and Recall@10 to the allowed coalition sizes and the number of sampled coalitions T, averaged over all five benchmarks.

Table 5: Effect of set-wise training, compared with point-wise training. Both variants use identical data, loss form and optimization budget.

Input granularity	MuSiQue		HotpotQA		2WikiMQA		MultiHopRAG		MorehopQA		Avg.	
	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10
Qwen3-Reranker-0.6B												
Point-wise trained	60.52	67.92	89.25	92.65	69.67	72.62	83.72	90.94	77.85	79.90	76.20	80.80
Set-wise trained	66.81	74.09	92.95	95.60	73.20	77.00	84.01	92.89	88.15	92.10	81.02	86.33
$\Delta\uparrow$	**+6.29**	**+6.17**	**+3.70**	**+2.95**	**+3.53**	**+4.38**	**+0.29**	**+1.95**	**+10.30**	**+12.20**	**+4.82**	**+5.53**
Qwen3-Reranker-4B												
Point-wise trained	66.77	73.41	92.85	95.50	73.42	76.42	87.20	93.27	82.75	85.60	80.60	84.84
Set-wise trained	70.86	76.77	94.80	96.85	74.85	79.00	82.97	92.39	89.35	94.00	82.57	87.80
$\Delta\uparrow$	**+4.09**	**+3.36**	**+1.95**	**+1.35**	**+1.43**	**+2.58**	-4.23	-0.88	**+6.60**	**+8.40**	**+1.97**	**+2.96**

recall that v_θ is trained only on MuSiQue, so the other four datasets test out-of-distribution generalization. Figure 6 yields two conclusions. *1) Set-wise training substantially improves set-level discrimination, and the gain transfers across datasets.* The set-wise trained valuer reaches an average satisfaction rate of 86.6%, a $+8.9$-point gain over the raw backbone (77.8%), with consistent improvements on every dataset, e.g., from $+4.9$ on MultiHopRAG up to $+10.8$ on 2WikiQA, including the four datasets never seen during training. *2) Point-wise training does not confer this ability.* Trained on the same data and backbone, the point-wise model leaves the satisfaction rate essentially unchanged on average (77.1%, -0.6 vs. the raw backbone) and even degrades it on MultiHopRAG (-4.5) and MorehopQA (-0.9). Optimizing per-document relevance therefore does not implicitly teach a model to compare document *sets*; the set-wise objective is what equips v_θ to serve as a coherent characteristic function for the cooperative game, which in turn explains why Shapley aggregation only pays off on the set-wise trained valuer (Sec. 6.4).

What about the Consistency between the two inference pipelines? We further verify that the amortized estimator is a faithful surrogate of sampling-based restricted attribution in Figures 9 and 10, showing that the two pipelines produce correlated rankings. The agreement is strongest among the top-ranked candidates, and the average top-k overlap stays high for the small k used to form the LLM context, so the amortized estimator can be deployed interchangeably where latency matters.

Case study. A qualitative example in Figure 7 and 8 illustrates the interpretability of the attributions: documents that directly support the multi-hop chain receive large positive attribution scores under both pipelines, redundant passages restating the same fact receive attenuated credit, and topically related distractors (e.g., documents about the correct entities but irrelevant facts) receive near-zero or negative values, visibly separating genuinely useful evidence from plausible-looking noise.

Table 6: Effect of sampling-based attribution on set-wise trained models.

Input granularity	MuSiQue		HotpotQA		2WikiMQA		MultiHopRAG		MorehopQA		Avg.	
	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10
Qwen3-Reranker-0.6B (Set-wise trained)												
w/o sampling-based attribution	63.20	69.44	90.05	93.60	70.13	72.85	81.02	89.93	77.70	79.85	76.42	81.13
w/ sampling-based attribution	66.81	74.09	92.95	95.60	73.20	77.00	84.01	92.89	88.15	92.10	81.02	86.33
$\Delta\uparrow$	**+3.61**	**+4.65**	**+2.90**	**+2.00**	**+3.07**	**+4.15**	**+2.99**	**+2.96**	**+10.45**	**+12.25**	**+4.60**	**+5.20**
Qwen3-Reranker-4B (Set-wise trained)												
w/o sampling-based attribution	64.31	72.97	91.80	95.05	69.25	75.12	77.08	88.51	82.20	85.55	76.93	83.44
w/ sampling-based attribution	70.86	76.77	94.80	96.85	74.85	79.00	82.97	92.39	89.35	94.00	82.57	87.80
$\Delta\uparrow$	**+6.55**	**+3.80**	**+3.00**	**+1.80**	**+5.60**	**+3.88**	**+5.89**	**+3.88**	**+7.15**	**+8.45**	**+5.64**	**+4.36**

Table 7: Reranking quality (Recall@5 / Recall@10) across five benchmarks. Best result in each column is in **bold**; second best is underlined. "Avg." averages the metric across all datasets. The set valuer is fine-tuned from the *Qwen3-Reranker-4B model*.

Method	#Params	MuSiQue		HotpotQA		2WikiMQA		MultiHopRAG		MorehopQA		Avg.	
		R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10
No reranking (DPR)	-	55.75	63.98	80.40	87.70	69.85	72.90	69.43	82.52	82.95	87.00	71.67	78.82
Point-wise rerankers													
BGE-v2-gemma	3B	66.30	72.66	94.70	96.40	74.68	77.42	82.24	91.64	86.60	89.55	80.90	85.53
Qwen3-Reranker-4B	4B	65.95	73.02	94.05	96.50	75.52	77.92	83.14	**93.70**	91.50	94.40	82.03	87.10
List/Set-wise rerankers													
RankVicuna	7B	54.92	62.06	86.00	91.90	72.78	74.72	69.12	82.32	87.25	88.50	74.01	80.05
RankZephyr	7B	58.09	65.46	87.65	92.45	71.98	74.92	64.16	76.67	89.80	90.75	73.37	79.26
RankGPT-style (Qwen3-8B)	8B	57.81	61.54	87.60	89.40	**77.65**	78.73	78.07	84.73	92.85	93.85	78.79	81.65
Set-Encoder	0.3B	56.42	61.89	88.20	90.26	71.76	74.23	66.34	85.28	87.15	92.18	73.97	80.76
SETR	8B	57.38	62.49	90.25	93.28	72.02	75.39	67.27	86.08	89.13	92.49	75.21	81.94
Trained on the same 10k MuSiQue data													
Qwen3-Reranker-4B	4B	66.77	73.41	92.85	95.50	73.42	76.42	**87.20**	93.27	82.75	85.60	80.60	84.84
Set-Encoder	0.3B	58.20	63.72	88.04	90.01	71.52	74.42	66.04	84.98	85.38	91.15	73.83	80.85
Ours (v_θ is set-wise trained from Qwen3-Reranker-4B)													
SHAPRANK-S	4B	70.86$_{\pm 0.05}$	**76.77**$_{\pm 0.08}$	**94.80**$_{\pm 0.09}$	96.85$_{\pm 0.07}$	74.85$_{\pm 0.06}$	**79.00**$_{\pm 0.04}$	82.97$_{\pm 0.01}$	92.39$_{\pm 0.05}$	89.35$_{\pm 0.51}$	94.00$_{\pm 0.42}$	82.57$_{\pm 0.14}$	87.80$_{\pm 0.13}$
SHAPRANK-A	0.4B	**71.14**$_{\pm 0.04}$	76.70$_{\pm 0.05}$	94.65$_{\pm 0.06}$	**96.95**$_{\pm 0.06}$	75.10$_{\pm 0.03}$	78.93$_{\pm 0.06}$	82.23$_{\pm 0.05}$	91.07$_{\pm 0.06}$	**93.70**$_{\pm 0.21}$	**95.90**$_{\pm 0.25}$	**83.36**$_{\pm 0.08}$	**87.91**$_{\pm 0.10}$

C.3 Permutation Robustness of the Set-wise Valuer

Although a cooperative-game characteristic function is formally defined on unordered coalitions, our valuer serializes the documents in a coalition into a sequence and is therefore not exactly permutation invariant by construction. As described in Sec. 4.1, we mitigate this dependence by uniformly shuffling document order during training. We quantify the residual order sensitivity here.

For each dataset, we sample query–coalition pairs and generate multiple random permutations of the documents within each coalition. We evaluate the same trained valuer on every permutation. For each coalition, we compute the standard deviation. To assess whether residual score variation materially affects reranking, we additionally compare the document rankings induced by independent permutation realizations using top-5 overlap and Spearman rank correlation.

Table 10 shows that the learned valuer exhibits only limited variation across document orders, while the induced rankings remain highly consistent. These results suggest that random-order augmentation substantially reduces the practical effect of serialization order, although the model is not exactly permutation invariant by architecture.

Table 8: Effect of sampling-based attribution on point-wise trained models.

Input granularity	MuSiQue		HotpotQA		2WikiMQA		MultiHopRAG		MorehopQA		Avg.	
	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10
Qwen3-Reranker-0.6B (Point-wise trained)												
w/o sampling-based attribution	60.52	67.92	89.25	92.65	69.67	72.62	83.72	90.94	77.85	79.90	76.20	80.80
w/ sampling-based attribution	62.76	70.30	89.20	94.35	70.53	73.92	82.33	91.59	83.75	86.90	77.71	83.41
Δ	+2.24	+2.38	-0.05	+1.70	+0.86	+1.30	-1.39	+0.65	+5.90	+7.00	+1.51	+2.61
Qwen3-Reranker-4B (Point-wise trained)												
w/o sampling-based attribution	66.77	73.41	92.85	95.50	73.42	76.42	87.20	93.27	82.75	85.60	80.60	84.84
w/ sampling-based attribution	68.56	75.64	93.60	96.25	67.60	73.98	84.26	91.91	81.85	86.35	79.17	84.82
Δ	+1.79	+2.23	+0.75	+0.75	-5.82	-2.44	-2.94	-1.36	-0.90	+0.75	-1.42	-0.02

Table 9: Downstream RAG answer quality (EM / F1) using the top-m documents, with a fixed downstream LLM. Best result in each column is in **bold**; second best is underlined. Results use DeepSeek-V3.2 as the downstream generator.

Method	#Params	MuSiQue		HotpotQA		2WikiMQA		MultiHopRAG		MorehopQA		Avg.	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
No reranking (retriever top-m)	-	26.70	39.05	52.40	65.70	49.80	54.85	27.14	30.00	41.95	42.66	39.60	46.45
Point-wise rerankers													
MonoT5	0.22B	29.60	41.78	57.00	70.43	47.40	52.21	27.27	30.00	45.44	46.27	41.34	48.14
BGE-v2-m3	0.57B	29.60	42.45	59.00	73.40	46.60	51.19	28.69	31.00	41.23	41.91	41.03	47.99
BGE-reranker-large	0.6B	28.50	41.26	57.90	72.16	46.20	50.75	27.27	29.94	39.36	39.95	39.85	46.81
Qwen3-Reranker-0.6B	0.6B	29.10	42.24	60.00	73.00	48.60	53.98	28.65	31.05	46.00	47.01	42.47	49.45
List/Set-wise rerankers													
RankVicuna	7B	30.00	42.30	57.60	70.94	49.20	54.15	26.34	29.66	45.71	46.66	41.76	48.74
RankZephyr	7B	31.90	43.81	57.30	70.57	50.20	55.93	25.05	28.31	47.09	48.66	42.31	49.45
RankGPT-style (Qwen3-8B)	8B	28.50	41.12	59.20	72.08	50.10	56.00	27.85	30.09	49.00	49.50	42.92	49.75
Set-Encoder	0.3B	29.60	41.32	59.19	72.02	50.12	54.83	28.01	30.72	48.14	48.92	43.01	49.56
SETR	8B	30.10	41.68	58.12	72.00	50.35	54.97	27.81	29.85	47.01	48.27	42.67	49.35
Ours (set-wise)													
SHAPRANK-S	0.6B	32.10	45.12	**61.10**	**74.27**	47.30	53.13	**29.67**	**32.19**	47.67	48.30	43.56	50.60
SHAPRANK-A	0.4B	**33.40**	**46.16**	60.40	74.09	**51.30**	**57.00**	28.51	31.41	**49.46**	**50.50**	**44.61**	**51.83**

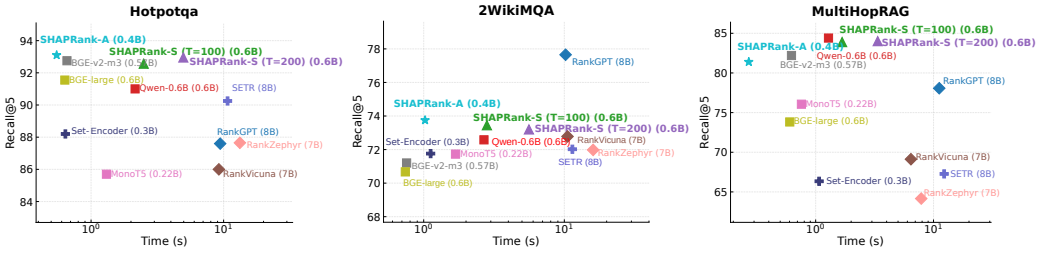


Figure 5: Efficiency results for more datasets.

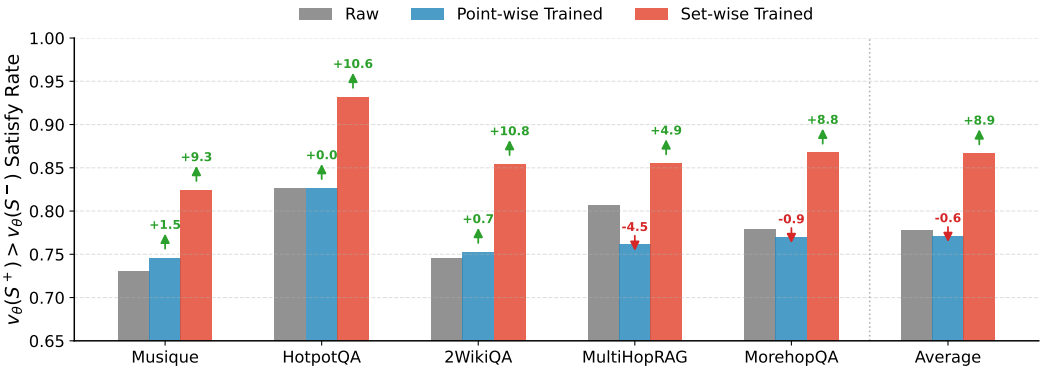


Figure 6: Score satisfaction rate of $v_\theta(S^+) > v_\theta(S^-)$. v_θ is trained only on the MuSiQue training set and evaluated on the test sets of all five benchmarks. Sampling 20000 (S^+, S^-) pairs for both positive sets and negative sets for each dataset.

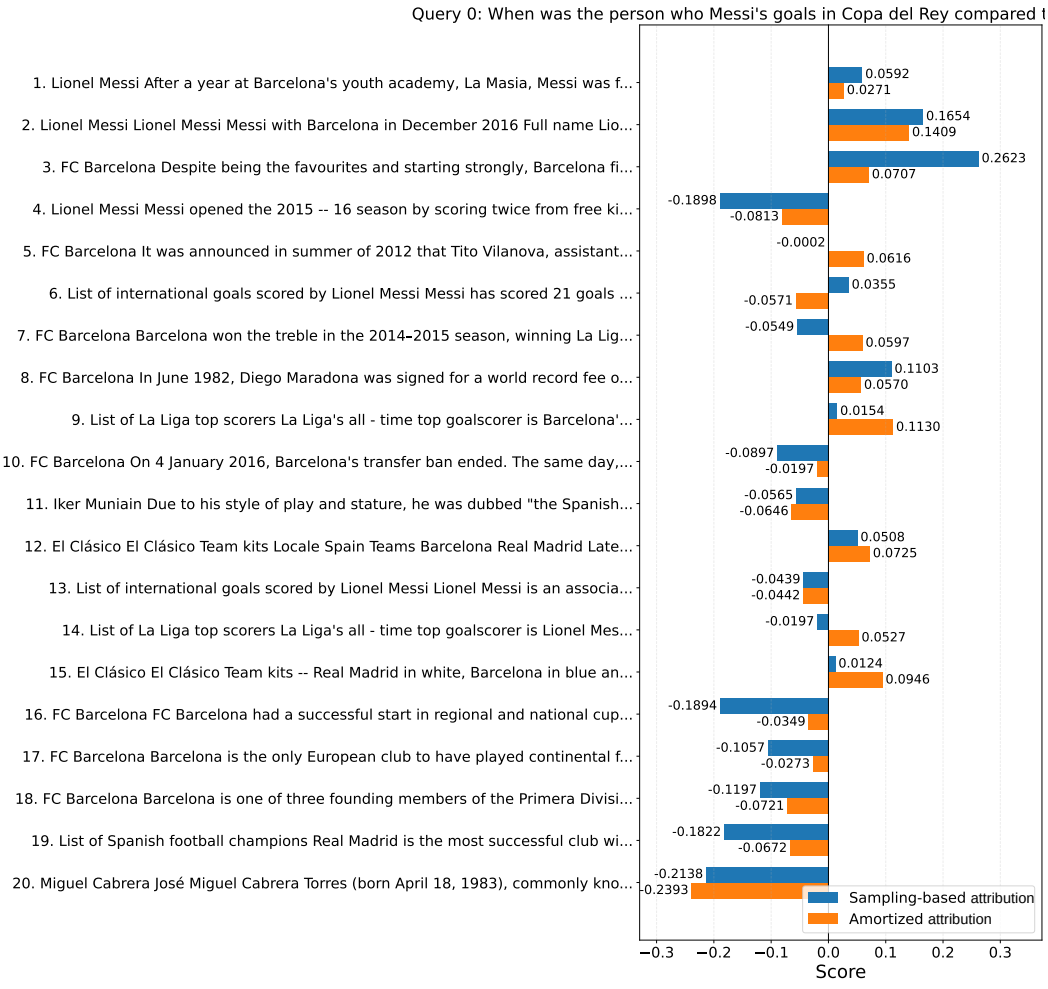


Figure 7: Case study for the score distribution on the MuSiQue dataset.

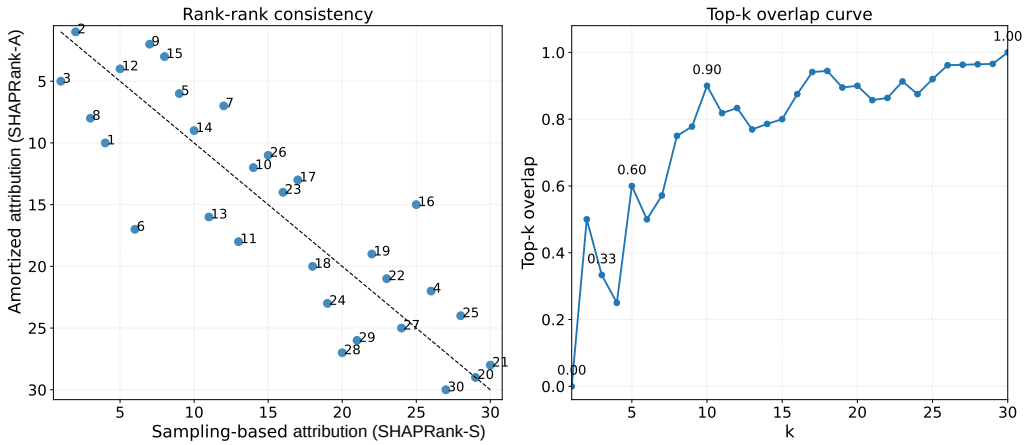


Figure 8: Case study for the score consistency on the MuSiQue dataset.

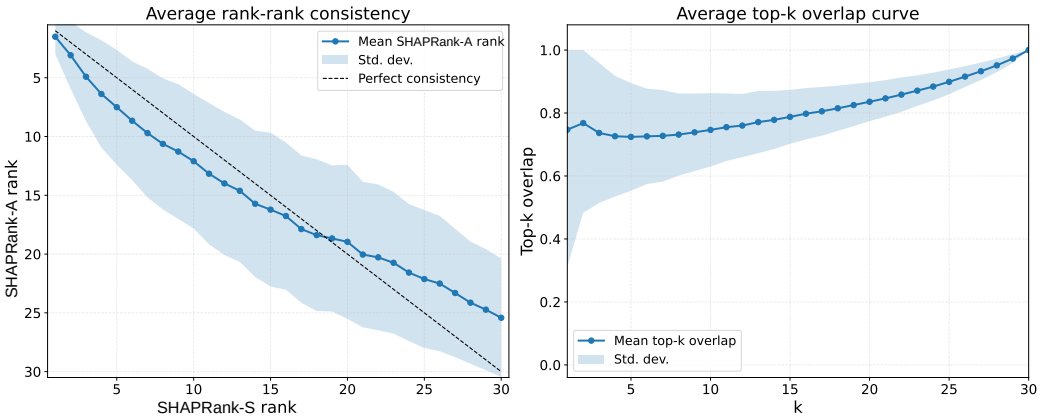


Figure 9: Consistency results on MuSiQue.

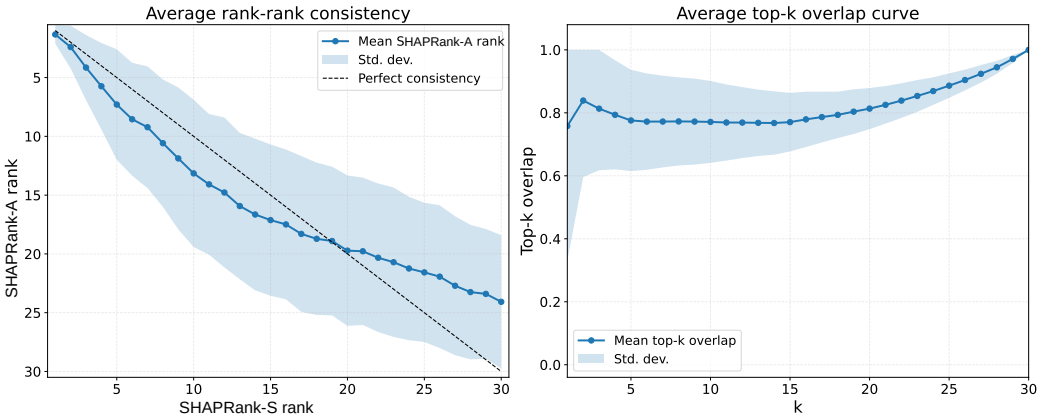


Figure 10: Consistency results on HotpotQA.

Table 10: Permutation robustness of the learned set-wise valuer. For each dataset, we evaluate sampled coalitions under multiple random document permutations. We report the variation of coalition utility scores and the agreement between the induced rankings.

Dataset	Score Mean	Score Std. ↓	Top-5 Overlap ↑	Spearman ρ ↑
MuSiQue	0.684	0.008	0.948	0.976
HotpotQA	0.731	0.006	0.957	0.982
2WikiMQA	0.662	0.010	0.931	0.965
MultiHopRAG	0.608	0.015	0.887	0.936
MorehopQA	0.701	0.011	0.921	0.958
Average	0.677	0.010	0.929	0.963