Reliable Evaluation Protocol for Low-Precision Retrieval

Anonymous ACL submission

Abstract

Lowering the numerical precision of model parameters and computations is widely adopted to improve the efficiency of retrieval systems. However, when computing relevance scores between the query and documents in low-precision, we observe *spurious ties* due to the reduced granularity. This introduces high variability in the results based on tie resolution, making the evaluation less reliable. To address this, we propose a more robust retrieval evaluation protocol designed to reduce score variation. It consists of: (1) High-Precision Scoring (HPS), which upcasts the final scoring step to higher precision to resolve tied candidates with minimal computational cost; and (2) Tie-aware Retrieval Metrics (TRM), which report expected scores, range, and bias to quantify order uncertainty of tied candidates. Our experiments test multiple models with three scoring functions on two retrieval datasets to demonstrate that HPS dramatically reduces tie-induced instability, and TRM accurately recovers expected metric values. This combination enables a more consistent and reliable evaluation system for lower-precision retrievals.[1]

1 Introduction

Recent studies on low-precision techniques have been widely explored (e.g., quantization and compression) to enhance the efficiency and scalability of neural networks while reducing computational cost (Nagel et al.; Kurtic et al., 2024; Zhu et al., 2024; Hao et al., 2025). Without sacrificing performance, these methods aim to lower the numerical precision of model weights, gradients, and activations in training and inference, along with the retrieval stage (Choi et al., 2024; Lee et al., 2025) of retrieval-augmented generation (RAG) (Wang et al., 2024; Zhang et al., 2024, 2025). To generate informative responses, retrieving accurate candidates is crucial; otherwise, the following stages may be negatively affected and result in incoherent outputs (Chen et al., 2024b; Yadav et al., 2024; Sharma, 2025).

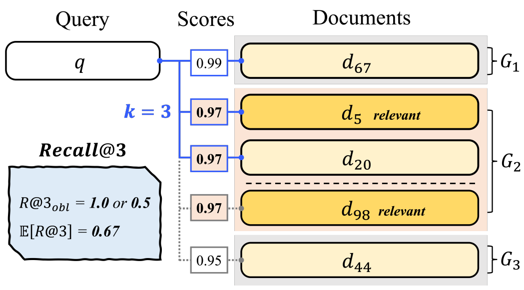


Figure 1: Example of tie-induced instability in evaluation metric. Three documents share the same score (G_2); two of them are relevant to the query. A tie-oblivious evaluation arbitrarily breaks the tie, so the reported R@3 depends on a random internal ordering. Instead, the tie-aware formulation deterministically reports the expectation over all permutations within the tie.

In neural retrieval systems, however, lowering numerical precision (e.g, `FP32` to `FP16`) inevitably reduces the granularity of representable floating point numbers (Shen et al., 2024; Hu et al., 2025) (see Appendix A); this coarser grid produces *spurious ties* among candidates by forcing many distinct relevance scores to quantize to the same value. Though resolving this issue can significantly affect evaluation scores (Figure 1), current mainstream retrieval evaluation systems (e.g., MTEB[2] (Muennighoff et al., 2023)) do not provide any principled mechanism for handling ties. Instead, they truncate the ranked list based on an arbitrary order (e.g., document IDs), which increases variances in results.

Thus, we propose a reliable evaluation protocol for low-precision retrieval. It is composed of (1) *High-Precision Scoring* (HPS) and (2) *Tie-aware Retrieval Metrics* (TRM). HPS upcasts the last scoring function into higher precision, to collapse spurious ties (§ 2.2). TRM is an expectation-based evalu-

[1]The source code will be available after the review period.

[2]https://github.com/embeddings-benchmark/mteb

ation augmented with extrema (i.e., maximum and minimum achievable scores) to quantify the order uncertainty of tied candidates (§ 2.3).

In our experiments, we demonstrate that evaluating low-precision models using conventional tie-oblivious metrics leads to misleading outcomes as shown in Figure 1. Adopting HPS significantly reduces score range variability, reducing MRR@10 range by 36.82%p. Meanwhile, TRM exposes biases inherent in tie-oblivious metrics, highlighting systematic overestimation by up to +9.08%p in BF16 evaluations. By contrast, our combined approach recovers near-FP32 stability and ordering, offering a consistent and discriminative framework for evaluating retrieval models in low-precision settings.

2 Reliable Evaluation Protocol

We first formalize the vulnerability of the current tie-oblivious evaluation, and then present *High-Precision Scoring* and *Tie-aware Retrieval Metric*. (See Appendix A for preliminaries.)

2.1 Spurious Ties in Low-Precision Evaluation

Let z denote the output of the linear layer after the last hidden state h. If the scoring function ϕ is softmax or sigmoid, then the cross-encoder takes the concatenated query and i-document pair $(q; d_i)$ as input and produces the logits z_i: two scalar values for softmax, or a single scalar value for sigmoid. If ϕ is a pairwise product, z_i denotes the pair of embeddings (h_q, h_{d_i}) obtained by encoding the query q and the document d_i independently with a bi-encoder. We denote the query-document relevance score $\tilde{s}_i$ as:

$$\tilde{s}_i = \phi^{(B)}(z_i) \tag{1}$$

where $\phi^{(B)}$ indicates that ϕ is operated entirely in a B-bit mantissa format.

Applied with low-precision inference (e.g., `BF16` (Burgess et al., 2019), `FP16`, etc), this maps theoretically continuous values onto a discrete set of representable scores; distinct true scores may collide: $\tilde{s}_i = \tilde{s}_j$ even with $z_i \neq z_j$, creating a *tie*. After sorting by $\tilde{s}$, we obtain ordered tie groups G_n consisting of scores s_i equivalent to v_n:

$$G_n = \{i \mid \tilde{s}_i = v_n\}. \tag{2}$$

If the relevant document at cutoff rank k falls inside a tie group G_n where $|G_n| \geq 2$, Any evaluation that disregards ties (*tie-oblivious*) may become stochastic and yield unpredictable results, as shown in Figure 1.

2.2 High-Precision Scoring (HPS)

Scoring functions such as softmax, sigmoid, and pairwise product compress logits into a narrow range. This effect is exacerbated under lower-precision formats due to fewer representable values resulting in coarser bucketization in $(0, 1)$ range (See examples in Appendix B).

For lower-precision models, HPS upcasts only the final scoring operation to `FP32`, leaving other layers unchanged. Concretely we replace the low-precision scoring function (Equation 1) with a higher-precision scoring function:

$$\hat{s}_i = \phi(\text{upcast}(z_i)), \tag{3}$$

and retain a more fine-grained score $\hat{s}_i$ for document candidate sorting. This significantly reduces the probability of tie collisions while preserving latency, since only a small logits tensor is upcast, requiring no re-training.

Advantages. HPS (i) leaves the forward pass intact and upcasts logits right before scoring, (ii) adds negligible memory and time overhead, (iii) collapses large tie groups, and (iv) restores alignment with deterministic, high-precision production sorting.

2.3 Tie-aware Retrieval Metric (TRM)

Existing tie-oblivious evaluation methods truncate the sorted list after a predefined cutoff k. If multiple candidates receive the same score, they are ordered arbitrarily before truncation, affecting which items are included in the top-k set. As a result, the evaluation results may vary depending on how ties are resolved as illustrated in Figure 1. To mitigate this problem, TRM supplies exact *expectations*, *range*, and a *bias*.

Expected Score. Let $G_1, \ldots, G_N$ be the tie groups sorted in descending order, where each group G_n has $|G_n|$ items and r_n relevant items to the given query. To mitigate the random ordering in tie groups, we propose reporting expected values for evaluation $\mathbb{E}[M]$ where M denotes an evaluation metric. This score averages the performance values across all possible result orderings and removes simulation variance. Since generating result permutations requires super-exponential time, we utilize closed-form expressions for calculating expectation

Models	FP32	BF16				BF16 → FP32(+HPS)			
	M	M_{obl}	$\mathbb{E}[M]$	Range(▼)	Bias(▼)	M_{obl}	$\mathbb{E}[M]$	Range(▼)	Bias(▼)
MIRACLReranking, M = *nDCG@10*									
`Qwen3-Reranker-0.6B`$^\clubsuit$	73.53	75.04	68.38	25.59	6.66	73.59	73.35	**1.13**	**0.24**
`bge-reranker-v2-m3`$^\diamond$	74.61	75.59	74.54	3.90	1.05	74.63	74.57	**0.16**	**0.06**
`gte-multilingual-reranker-base`$^\diamond$	74.14	74.48	74.22	0.97	0.26	74.39	74.34	**0.14**	**0.05**
`Qwen3-Embedding-0.6B`$^\spadesuit$	63.94	64.52	63.98	1.90	0.54	64.01	64.01	**0.00**	**0.00**
`multilingual-e5-large-large`$^\spadesuit$	64.78	65.70	64.81	4.62	0.89	64.80	64.80	**0.00**	**0.00**
MIRACLReranking, M = *MRR@10*									
`Qwen3-Reranker-0.6B`$^\clubsuit$	77.48	78.45	69.37	38.03	9.08	77.43	77.22	**1.21**	**0.21**
`bge-reranker-v2-m3`$^\diamond$	79.58	80.68	79.17	6.72	1.51	79.66	79.56	**0.19**	**0.10**
`gte-multilingual-reranker-base`$^\diamond$	79.39	79.75	79.47	0.85	0.28	79.59	79.52	**0.18**	**0.07**
`Qwen3-Embedding-0.6B`$^\spadesuit$	68.97	69.54	68.91	2.23	0.63	69.02	69.02	**0.00**	**0.00**
`multilingual-e5-large-large`$^\spadesuit$	71.37	71.84	71.28	4.61	0.56	71.18	71.18	**0.00**	**0.00**
AskUbuntuDupQuestions, M = *MAP@3*									
`Qwen3-Reranker-0.6B`$^\clubsuit$	31.20	33.28	31.13	4.03	2.15	31.58	31.29	**0.57**	**0.29**
`bge-reranker-v2-m3`$^\diamond$	31.91	32.26	31.83	0.83	0.43	31.89	31.84	**0.09**	**0.05**
`gte-multilingual-reranker-base`$^\diamond$	30.83	31.23	30.75	0.93	0.48	30.69	30.67	**0.03**	**0.02**
`Qwen3-Embedding-0.6B`$^\spadesuit$	29.54	30.10	29.65	0.87	0.45	29.69	29.69	**0.00**	**0.00**
`multilingual-e5-large-large`$^\spadesuit$	29.13	31.31	29.47	3.54	1.84	29.70	29.70	**0.00**	**0.00**

Table 1: Results using metric M with its tie-oblivious version (M_{obl}), expectation ($\mathbb{E}[M]$), range ($M_{\max} - M_{\min}$), and bias ($M - \mathbb{E}[M]$) on MIRACLEReranking (nDCG@10 and MRR@10) and AskUbuntuDupQuestions (MAP@3) under three precision regimes, full `FP32`, `BF16`, and `BF16`→`FP32` (with High-Precision Scoring). In full `FP32` we empirically observe $M_{obl} = \mathbb{E}[M]$ with zero range and bias, so only M is shown. ♣, ◇ and ♠ indicate softmax, sigmoid and pairwise product, respectively. Lower range and |bias| scores represent better stablity.

values (McSherry and Najork, 2008). Explicit formulas are presented in Appendix C; the linear time complexity analysis is in Appendix D.

Score Range. $M_{\max}$ places the query-relevant items in each partially included tie group as early as possible; $M_{\min}$ as late as possible.

$$\text{Range}(M) = M_{\max} - M_{\min}. \tag{4}$$

Range(M) quantifies uncertainty due solely to unresolved internal orderings. A smaller range indicates that results are more stable and reliable.

Score Bias. Let M_{obl} denote the tie-oblivious metric obtained using the original implementation's fixed (typically index-preserving) ordering. We define the score bias as

$$\text{Bias}(M) = M_{\text{obl}} - \mathbb{E}[M]. \tag{5}$$

A large positive bias implies that M_{obl} does not reliably estimate the expected positive values, indicating overestimation of results, while negative values indicate underestimation.

Reporting Protocol. For each cutoff k (or full ranking if required), we propose to report the expectation value and the range of score variance:

$$\big(\mathbb{E}[M], \text{Range}(M)\big), \tag{6}$$

optionally reporting the tie-oblivious value M_{obl}, discrepancy Bias(M), the extrema $M_{\max}$ and $M_{\min}$.

Models	ϕ	Size
`Qwen3-Reranker-0.6B` (Zhang et al., 2025)	Softmax$^\clubsuit$	596M
`bge-reranker-v2-m3` (Chen et al., 2024a)	Sigmoid$^\diamond$	568M
`gte-multilingual-reranker-base` (Zhang et al., 2024)	Sigmoid$^\diamond$	306M
`Qwen3-Embedding-0.6B` (Zhang et al., 2025)	Product$^\spadesuit$	596M
`multilingual-e5-large` (Wang et al., 2024)	Product$^\spadesuit$	560M

Table 2: Models used in our experiments and their corresponding scoring function and size.

With expectation and range values, our proposed reporting protocol enables more reliable evaluation.

3 Experiments

We evaluate to what degree our proposed evaluation protocol exposes and corrects reliability failures of existing tie-oblivious evaluation.

3.1 Experimental Setting

More detailed explanations of experimental settings and implementation are presented in Appendix E

Models We cover five models widely used in reranking and embedding with three prevalent scoring functions: Softmax♣, sigmoid◇, and pairwise product♠ as in Table 2.

Evaluation Metric We evaluate the standard ranking metrics nDCG (Järvelin and Kekäläinen, 2002), MRR (VOORHEES, 2000), MAP (Salton, 1983), and Recall.[3]

[3]Results for all metrics are deferred to Appendix F.

Datasets We utilize two publicly available datasets, **MIRACLReranking** (Zhang et al., 2023) and **AskUbuntuDupQuestions** (Lei et al., 2016) that each supplies a fixed set of candidates per query. This enables us to assess the second-stage reranker or retriever independent of effects from the first-stage retriever.

3.2 Results

Spurious Ties in Low-Precision Evaluation. When using fully BF16, the results display significant uncertainty as shown in Table 1. Qwen3-Reranker model with softmax♣ shows the highest variation — 25.59%p in nDCG@10 and 38.03%p in MRR@10. Models using sigmoid◇ and pairwise product♠ also exhibit instability to a lesser extent. These ranges exceed the margins typically used to distinguish model superiority.

Crucially, a striking decision error appears. Under the BF16 and nDCG@10$_{obl}$ evaluation, Qwen3-Reranker seems to beat gte (75.04 > 74.48). However, tie-aware metric $\mathbb{E}$[nDCG@10] flips the ranking (68.38 < 74.22), and our proposed protocol (HPS + TRM) confirms the reversal (73.35 < 74.34) within a narrow range, rendering the naive evaluation rankings unreliable.

Albeit bias can be positive or negative, all BF16 biases are positive, implying that tie-oblivious M_{obl} is overestimated (up to +9.08%p). This positive trend is likely a result of errors in dataset construction, coupled with deterministic tie-breaking, as the positive items are more consistently placed earlier in the dataset to create preferential tie groups.

High-Precision with Low-Cost. High precision scoring (HPS) collapses the large tie groups while keeping the bulk of computation in BF16. Softmax ranges shrink from 25.59 to 1.13%p at nDCG@10 and from 38.03 to 1.21%p at MRR@10; sigmoid◇ model ranges drop roughly an order of magnitude (e.g., 3.90 to 0.16%p in nDCG@10); pairwise product models become perfectly deterministic (range = bias = 0). The remaining softmax residual range ($\sim$ 1%p) lies within ordinary inter-model differences, making rank reversals highly unlikely.

Compared to fully FP32 inference (stable but computationally costlier), HPS recovers near-FP32 stability and ordering with negligible overhead. Hence, pure low-precision scoring erodes evaluation reliability, and adopting our protocol, HPS with reporting ($\mathbb{E}[M]$, Range), restores precise and discriminative comparisons.

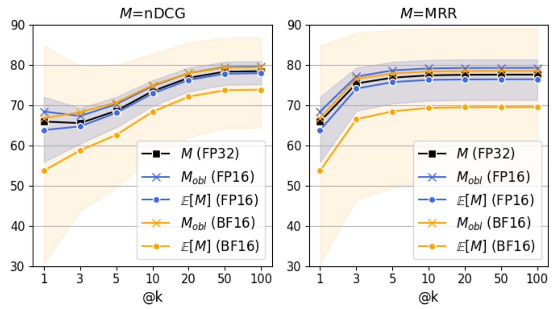


Figure 2: Tie-oblivious and expectation scores of nDCG and MRR at k of Qwen3-Reranker-0.6B♣ model when scored with each **dtype** on MIRACLReranking.

Impact of Precision across Cutoffs. Figure 2 shows nDCG and MRR metrics across various k-rank cutoffs, illustrating increased variance ranges and biases under lower-precision computations. Consistent with our observations in Appendix A, the BF16 inference displays significant fluctuations and uncertainty (wide shaded areas), whereas FP16 demonstrates intermediate stability, and FP32 offers empirically stable results with negligible ties. This reflects the coarser bucketization induced by fewer mantissa bits in lower-precision formats (BF16 ≪ FP16 ≪ FP32).

Notably, under M_{obl}, the BF16 curves surpass the FP32 baseline at every cutoff. Such results would incorrectly indicate better performance, highlighting the unreliability of tie-oblivious evaluation due to reduced precision. Conversely, the tie-aware expectation $\mathbb{E}[M]$ consistently places BF16 below FP32, accurately reflecting the true model performance, shown in Appendix F.

4 Conclusion

We demonstrate that current retrieval evaluations under low-precision settings overlook tied candidates, resulting in unstable outcomes. To address this, we proposed two concise yet effective remedies: High-Precision Scoring (HPS) and Tie-aware Retrieval Metrics (TRM). HPS upcasts the final scoring function to collapse spurious ties with negligible cost, and TRM reports the expectation value of scores with range and bias. Our proposed combination mitigates spurious ties across precision formats and provides a more reliable alternative to previous naive methods. Our method enables more stable document retrieval in tasks such as retrieval augmented generation (RAG), while preserving the efficiency and memory savings offered by low-precision models.

Limitations

Our remedy targets the inference stage and does not explore how low-precision training influences ranking stability, nor whether mixed-precision training combined with HPS inference yields further gains. Finally, TRM's outputs, expectations with ranges, are richer than single scalars, yet we have not conducted user-centered studies to assess their interpretability in practical evaluation pipelines.

References

Neil Burgess, Jelena Milanovic, Nigel Stephens, Konstantinos Monachopoulos, and David Mansell. 2019. Bfloat16 processing for neural networks. In *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*, pages 88–91. IEEE.

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

Weijie Chen, Ting Bai, Jinbo Su, Jian Luan, Wei Liu, and Chuan Shi. 2024b. Kg-retriever: Efficient knowledge indexing for retrieval-augmented large language models. *CoRR*.

Chanyeol Choi, Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, and Jy-yong Sohn. 2024. Linq-embed-mistral technical report. *arXiv preprint arXiv:2412.03223*.

Zhiwei Hao, Jianyuan Guo, Li Shen, Yong Luo, Han Hu, Guoxia Wang, Dianhai Yu, Yonggang Wen, and Dacheng Tao. 2025. Low-precision training of large language models: Methods, challenges, and opportunities. *arXiv preprint arXiv:2505.01043*.

Weiming Hu, Haoyan Zhang, Cong Guo, Yu Feng, Renyang Guan, Zhendong Hua, Zihan Liu, Yue Guan, Minyi Guo, and Jingwen Leng. 2025. M-ant: Efficient low-bit group quantization for llms via mathematically adaptive numerical type. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 1112–1126. IEEE.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Eldar Kurtic, Alexandre Marques, Shubhra Pandit, Mark Kurtz, and Dan Alistarh. 2024. " give me bf16 or give me death"? accuracy-performance trade-offs in llm quantization. *arXiv preprint arXiv:2411.02355*.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, and 1 others. 2025. Gemini embedding: Generalizable embeddings from gemini. *arXiv preprint arXiv:2503.07891*.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. 2016. Semi-supervised question retrieval with gated convolutions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1279–1289.

Frank McSherry and Marc Najork. 2008. Computing information retrieval performance measures efficiently in the presence of tied scores. In *European conference on information retrieval*, pages 414–421. Springer.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. arxiv 2021. *arXiv preprint arXiv:2106.08295*, 4.

Gerard Salton. 1983. Modern information retrieval. *(No Title)*.

Chaitanya Sharma. 2025. Retrieval-augmented generation: A comprehensive survey of architectures, enhancements, and robustness frontiers. *arXiv preprint arXiv:2506.00054*.

Haihao Shen, Naveen Mellempudi, Xin He, Qun Gao, Chang Wang, and Mengni Wang. 2024. Efficient post-training quantization with fp8 formats. *Proceedings of Machine Learning and Systems*, 6:483–498.

EM VOORHEES. 2000. The trec-8 question answering track report. In *Proc. Eighth Text REtrieval Conference (TREC-8), NIST Special Publication 500-246*, pages 77–82.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Wikipedia contributors. 2025. Bfloat16 floating-point format.

Neemesh Yadav, Sarah Masud, Vikram Goyal, Md Shad Akhtar, and Tanmoy Chakraborty. 2024. Tox-bart: Leveraging toxicity attributes for explanation generation of implicit hate speech. In *ACL (Findings)*.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, and 1 others. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1393–1412.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. Miracl: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131.

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

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577.

A Preliminaries

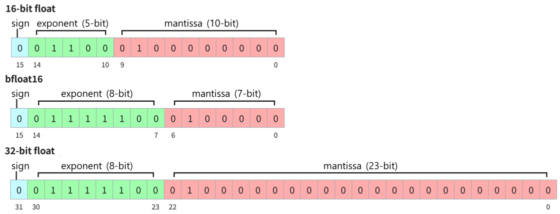


Figure 3: Bit layouts of FP16, BF16, and FP32 formats (Wikipedia contributors, 2025)

Floating-Point Value A floating-point value is a way to represent numbers in computer systems, and typically encoded as three fields—*sign*, *exponent*, and *mantissa* (also called the fraction)—as illustrated in Figure 3. The exponent determines the dynamic range, the largest and smallest magnitudes that can be represented, whereas the mantissa governs the *precision* attainable within that range. Since a shorter mantissa implies coarser quantization, multiple real numbers inevitably collapse into the same representable bin, producing tied values.

After the common 1-bit sign, FP16 allocates 5 exponent bits and 10 mantissa bits, BF16 uses 8 and 7 bits respectively, and FP32 retains 8 exponent bits alongside a much longer 23-bit mantissa. By preserving the full 8-bit exponent of FP32, BF16 inherits the same dynamic range as single precision, which is widely credited with stabilizing training and thereby aiding generalization.

However, when outputs are confined to the range (0, 1)—as with the probabilities emitted by softmax or sigmoid scoring functions—the short 7-bit mantissa of BF16, and to a lesser extent the 10-bit mantissa of FP16, sharply reduces resolution. This loss of granularity, particularly severe in BF16, exacerbates the tied-score phenomenon and makes it difficult to distinguish among retrieval candidates that quantize to identical values.

B Examples of Relevance Scores

The example lists below show raw relevance scores for the first query of the MIRACLReranking test split produced by the `Qwen3-Reranker-0.6B` model where relevant values for the given are in **bold**. The first list (`scores_bf16`) is obtained with both the model and scoring function executed entirely in BF16, while the second (`scores_hps`) applies High Precision Scoring (HPS). Tie group sizes shrink considerably under HPS.

scores_bf16 = [
1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000,
1.00000000, 1.00000000, 1.00000000, 1.00000000, 1.00000000,
0.99609375, 0.99609375, **0.99609375**, 0.99609375, 0.99609375,
0.99609375, 0.99609375, 0.99609375, 0.99609375, 0.99609375,
0.99609375, 0.99609375, 0.99609375, 0.99609375, 0.99609375,
0.99609375, 0.99609375, 0.99609375, 0.99609375, 0.99218750,
0.99218750, 0.99218750, 0.99218750, 0.99218750, 0.99218750,
0.99218750, 0.99218750, 0.99218750, 0.99218750, 0.99218750,
0.99218750, 0.99218750, 0.98828125, 0.98828125, **0.98828125**,
0.98828125, 0.98828125, 0.98828125, 0.98828125, 0.98437500,
0.98437500, 0.98046875, 0.97656250, 0.97656250, 0.97265625,
0.96875000, 0.96875000, 0.96875000, 0.96875000, 0.96875000,
0.96875000, 0.96093750, 0.96093750, 0.96093750, 0.96093750,
0.95703125, 0.95703125, 0.95703125, 0.95703125, 0.95703125,
0.95312500, 0.95312500, 0.94921875, 0.94921875, 0.94531250,
0.94531250, 0.94531250, 0.94140625, 0.94140625, 0.93359375,
0.92578125, 0.92578125, 0.91796875, 0.91406250, 0.88671875,
0.88671875, 0.87890625, 0.87890625, 0.87500000, 0.86718750,
0.77734375, 0.60937500, 0.51562500, 0.46875000, 0.34960938,
0.30664062, 0.28125000, 0.17285156, 0.08496094, 0.02441406,
]

scores_hps = [
0.99948066, 0.99933332, 0.99929035, 0.99919587, 0.99914408,
0.99883050, 0.99883050, 0.99875510, 0.99858958, 0.99829930,
0.99767691, **0.99767691**, 0.99752742, 0.99752742, 0.99736834,
0.99719906, 0.99719906, 0.99701905, 0.99682730, 0.99662340,
0.99662340, 0.99592990, 0.99566853, 0.99566853, 0.99509466,
0.99509466, 0.99477994, 0.99444515, 0.99444515, 0.99408901,
0.99408901, 0.99408901, 0.99330717, 0.99330717, 0.99330717,
0.99242276, 0.99142247, 0.99142247, 0.99142247, 0.99087441,
0.99087441, 0.99029154, 0.98967183, **0.98901308**, 0.98901308,
0.98901308, 0.98831278, 0.98831278, 0.98756832, 0.98593640,
0.98409361, 0.98201376, 0.97838473, 0.97702265, 0.97404259,
0.97068775, 0.96885622, 0.96885622, 0.96885622, 0.96691406,
0.96691406, 0.96267307, 0.96036118, 0.96036118, 0.96036118,
0.95791227, 0.95791227, 0.95791227, 0.95791227, 0.95531917,
0.95257413, 0.95257413, 0.94966936, 0.94966936, 0.94659668,
0.94659668, 0.94659668, 0.93991333, 0.93991333, 0.93245327,
0.92414182, 0.92414182, 0.91964257, 0.91490096, 0.88720459,
0.88720459, 0.88079703, 0.88079703, 0.87407720, 0.86703575,
0.77729988, 0.60766321, 0.51561993, 0.46879065, 0.34864515,
0.30735803, 0.28140560, 0.17328820, 0.08509904, 0.02442309,
]

C Closed-form Expectations

Let the tie groups be $G_1, \dots, G_N$ in descending score order. Each group G_n has size $|G_n|$ and r_n relevant items ($0 \le r_n \le |G_n|$). Define the per-group relevance probability

$$p_n = \frac{r_n}{|G_n|}, \tag{7}$$

and the cumulative size

$$c_n = \sum_{m \le n} |G_m|, \qquad c_0 = 0. \tag{8}$$

For a cutoff rank k, the number of items from group G_n that appear within the top-k list is

$$t_n = \max\{0,\ \min(|G_n|,\ k - c_{n-1})\}. \tag{9}$$

Count-based Metrics

With $N_+ = \sum_m r_m$,

$$\mathbb{E}[\text{Hits@}k] = \sum_{n:t_n>0} p_n t_n, \tag{10}$$

$$\mathbb{E}[\text{Recall@}k] = \frac{\sum_n p_n t_n}{N_+}, \tag{11}$$

$$\mathbb{E}[\text{Precision@}k] = \frac{\sum_n p_n t_n}{k}, \tag{12}$$

$$\mathbb{E}[\text{F1@}k] = \frac{2\sum_n p_n t_n}{k + N_+}. \tag{13}$$

nDCG

With binary gains and weights $w_r = \frac{1}{\log_2(r+1)}$, define

$$W(a,b) = \sum_{r=a}^{b} w_r. \tag{14}$$

Then

$$\mathbb{E}[\text{DCG@}k] = \sum_{n:t_n>0} p_n W(c_{n-1}+1,\ c_{n-1}+t_n), \tag{15}$$

$$\text{IDCG@}k = \sum_{r=1}^{\min(N_+,k)} w_r, \tag{16}$$

$$\mathbb{E}[\text{nDCG@}k] = \frac{\mathbb{E}[\text{DCG@}k]}{\text{IDCG@}k}. \tag{17}$$

Reciprocal Rank

Let $n^* = \min\{n \mid r_n > 0\}$ be the first group containing a relevant item and $\binom{x_a}{x_b}$ be the binomial coefficient. If $k \le c_{n^*-1}$ then $\mathbb{E}[\text{RR@}k] = 0$; otherwise

$$u = \min(|G_{n^*}| - 1,\ k - c_{n^*-1} - 1) \tag{18}$$

$$r_t = c_{n^*-1} + t + 1 \tag{19}$$

$$\pi_t = \frac{\binom{|G_{n^*}| - r_{n^*}}{t}}{\binom{|G_{n^*}|}{t}} \tag{20}$$

$$\lambda_t = \frac{r_{n^*}}{|G_{n^*}| - t} \tag{21}$$

$$\mathbb{E}[\text{RR@}k] = \sum_{t=0}^{u} \frac{1}{r_t} \pi_t \lambda_t. \tag{22}$$

Average Precision

For rank $r = c_{n-1} + t + 1$ with $0 \le t < t_n$ in group G_n,

$$A_{n,t} = R_{n-1} + 1 + t\frac{r_n - 1}{|G_n| - 1}, \tag{23}$$

$$D_{n,t} = c_{n-1} + t + 1, \tag{24}$$

where $R_{n-1} = \sum_{m<n} r_m$. The expected AP@k is

$$\mathbb{E}[\text{AP@}k] = \frac{1}{N_+} \sum_{n:r_n>0} \sum_{t=0}^{t_n-1} p_n \frac{A_{n,t}}{D_{n,t}}. \tag{25}$$

D Time Complexity

Let the ranked list for one query contain L candidate documents and let the evaluation cutoff be k. The list is partitioned into N tie groups $G_1, \dots, G_N$ of sizes $|G_1|, \dots, |G_N|$ with $\sum_{n=1}^{N} |G_n| = L$. All complexities below are per query.

High-Precision Scoring (HPS). Only the final logits are upcast to `FP32` and passed once through a scoring function ϕ, so the time cost is $O(L)$ with negligible extra memory.

Tie-aware Metrics (TRM). All computations occur after sorting, so no additional $\log L$ factor is introduced. (i) A single left-to-right scan gathers the pairs $(|G_n|, r_n)$ for every tie group in $O(L)$ where r_n refers to the number of relevant items in the n-th tie group G_n. (ii) Closed-form expressions let nDCG, MAP, Recall, Precision, and F1 be evaluated in $O(\min\{k, N\})$ time. (iii) MRR examines only the first tie group containing a relevant document, costing $O(|G_{j^*}|) \le O(k)$ where j^* is the index of the tie group that includes the first relevant item. (iv) Max, min, and range scores need only the tie group that straddles rank k, again $O(k)$.

In total TRM adds at most $O(k + N) \subseteq O(L)$ lightweight arithmetic per query, far below the cost of the forward pass or initial sort, while providing tie-robust evaluation.

E Experimental Settings

E.1 Implementation Details

We use a maximum input length of 4,096 tokens[4] and a batch size of 16[5]. All models are run under three data types: `BF16`, `FP16`, and `FP32`. HPS is implemented by upcasting the final scoring operation to `FP32`. Baseline tie-oblivious scores rely on the framework's predefined index order inside ties. In contrast, tie-aware expectations and extrema are computed with TRM (Section 2.3).

E.2 Datasets

MIRACLReranking We adopt the English subset of the **MIRACLReranking** test split (Zhang et al., 2023), derived from an open-domain Wikipedia. After discarding queries without a relevant passage, 717 of the original 799 queries remain; each with exactly 100 candidate passages ($\approx$ 2.9 relevant passages on average).

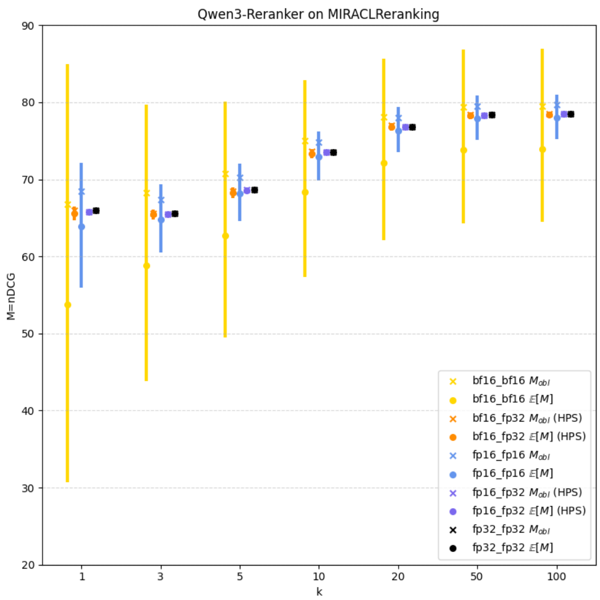
AskUbuntuDupQuestions For evaluation, each query is a concise **AskUbuntuDupQuestions** (Lei et al., 2016) question with at least one manually annotated duplicate. The test split contains 375 queries, each accompanied by 20 candidate questions ($\approx$ 6 true duplicates on average).

[4]Only `multilingual-e5-large` is truncated to 512 tokens due to its length constraints.

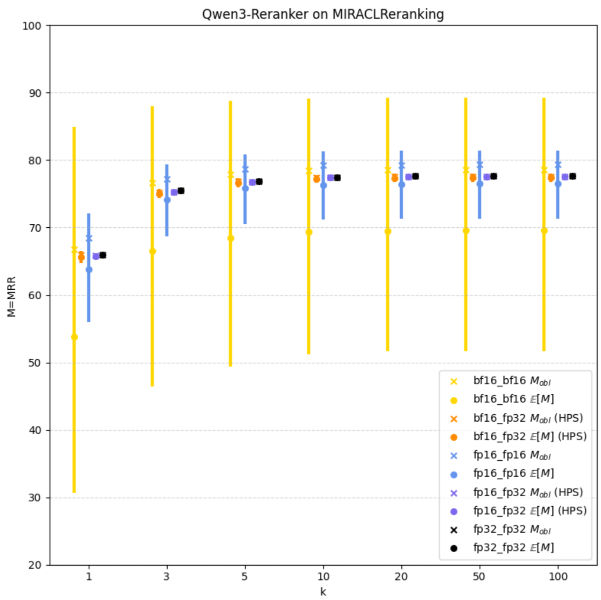
[5]Batch size affects the representations produced by low-precision inference, even with identical inputs.

F Full Experimental Results

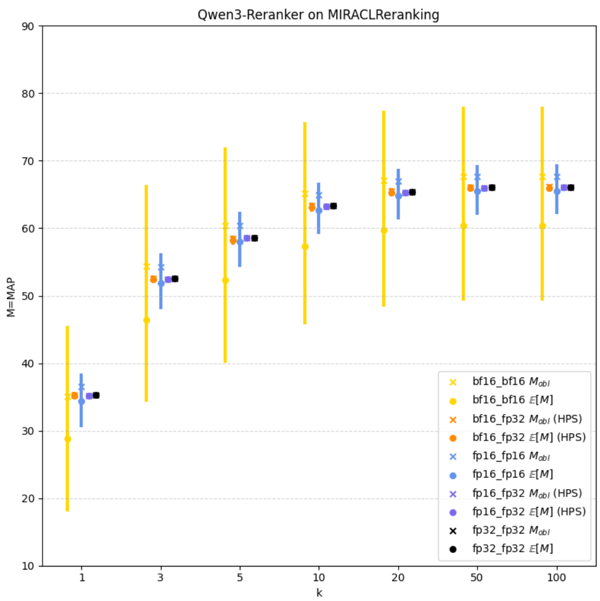
We present the detailed experimental results for both datasets in Figure 4 and 5. We attach results for `Qwen3-Reranker-0.6B`, which is known as the state-of-the-art in general text retrieval tasks. Panels (a)-(d) report nDCG, MRR, MAP, and Recall. Each marker shows the tie-oblivious score M_{obl} ($\times$) and the tie-aware expectation $\mathbb{E}[M]$ (●). The legend entry indicates the data types of the model and scoring function, respectively. For example, `BF16_FP32` denotes that the model operates in `BF16` precision, while the scoring function is upcast to `FP32`, corresponding to the HPS setting.



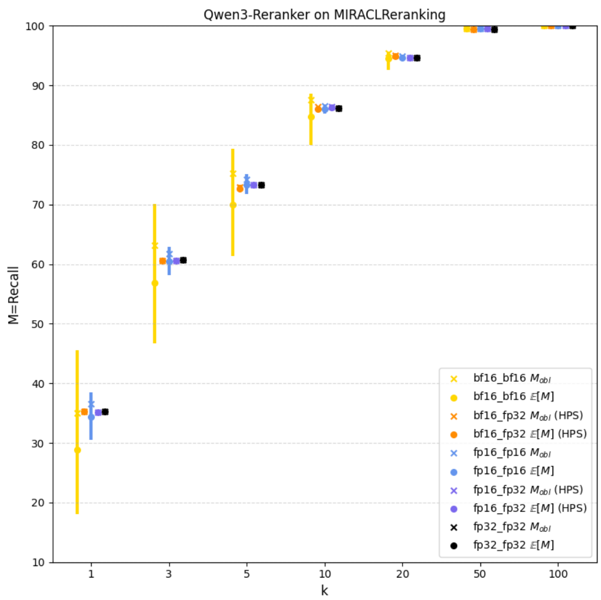
(a) nDCG



(b) MRR

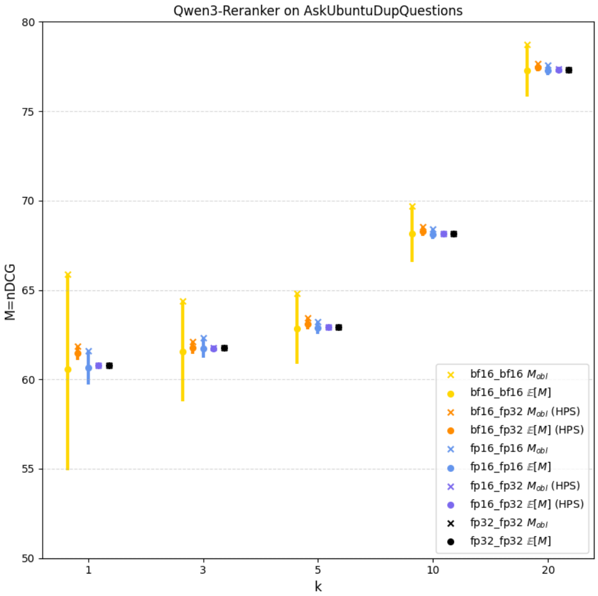


(c) MAP

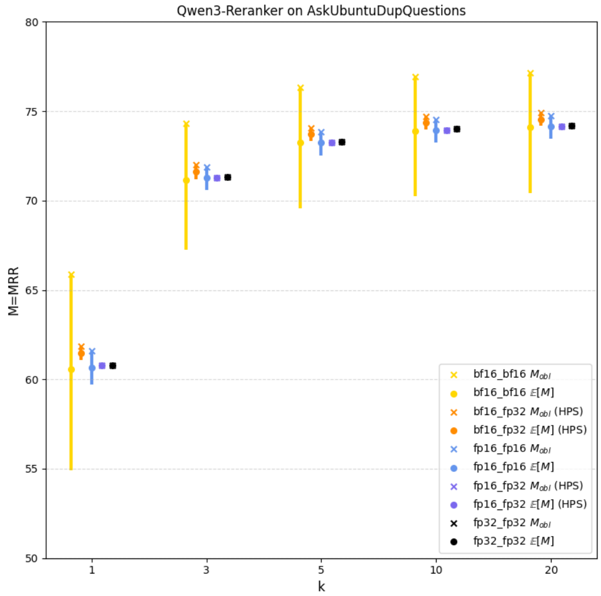


(d) Recall

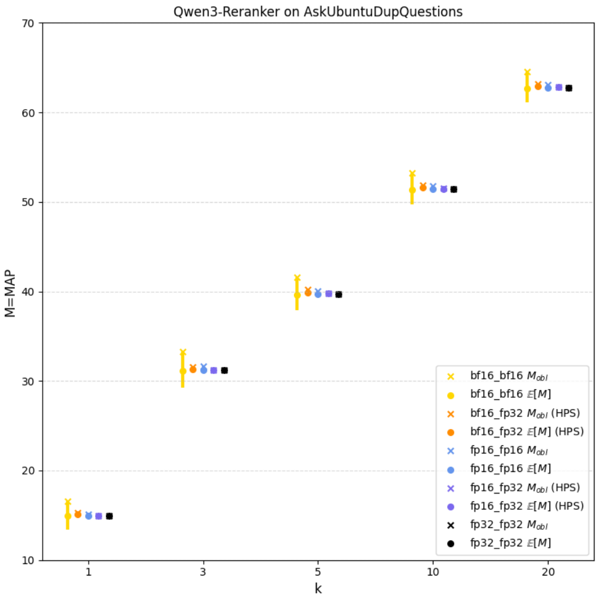
Figure 4: Metric scores for cutoff k of `Qwen3-Reranker-0.6B` on **MIRACLReranking** dataset.



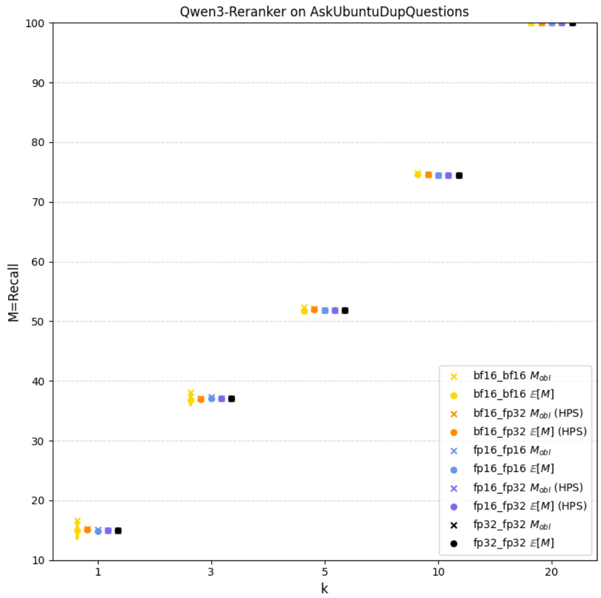
(a) nDCG



(b) MRR



(c) MAP



(d) Recall

Figure 5: Metric scores for cutoff k of `Qwen3-Reranker-0.6B` on **AskUbuntuDupQuestions** dataset. In this dataset, all tie-oblivious metrics attain their maximum possible value (being overestimated) because, during candidate construction, every relevant item is concatenated ahead of all non-relevant ones.[6]

[6] https://github.com/embeddings-benchmark/mteb/blob/1.38.38/mteb/evaluation/evaluators/RerankingEvaluator.py#L175