CONFIDENT RAG: ENHANCING THE PERFORMANCE OF LLMS FOR MATHEMATICS QUESTION ANSWERING THROUGH MULTI-EMBEDDING AND CONFIDENCE SCORING

Shi-Ting Chen[1], **Zijian Zhao**[2], **Jinsong Chen**[1]
[1] Faculty of Education, The University of Hong Kong
[2] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology
u3011355@connect.hku.hk; zzhaock@connect.ust.hk; jinsong.chen@live.com
*Equal Contributions: Shi-Ting Chen, Zijian Zhao
*Corresponding Authors: Shi-Ting Chen, Jinsong Chen

ABSTRACT

Recently, as Large Language Models (LLMs) have fundamentally impacted various fields, the methods for incorporating up-to-date information into LLMs or adding external knowledge to construct domain-specific models have garnered wide attention. Retrieval-Augmented Generation (RAG), serving as an inference-time scaling method, is notable for its low cost and minimal effort for parameter tuning. However, due to heterogeneous training data and model architecture, the variant embedding models used in RAG exhibit different benefits across various areas, often leading to different similarity calculation results and, consequently, varying response quality from LLMs. To address this problem, we propose and examine two novel approaches that combine the benefits of multiple embedding models, named Mixture-Embedding RAG and Confident RAG. Mixture-Embedding RAG simply sorts and selects retrievals from multiple embedding models based on standardized similarity; however, it does not outperform vanilla RAG. In contrast, Confident RAG generates responses multiple times using different embedding models and then selects the responses with the highest confidence level, demonstrating average improvements of approximately 10% and 5% over vanilla LLMs and RAG, respectively. The consistent results across different LLMs and embedding models indicate that Confident RAG is an efficient plug-and-play solution for mathematics question answering. The code of our paper is provided at `https://github.com/RS2002/Confident-RAG`.

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable capabilities across various domains (Lyu et al., 2025; Zhao et al., 2025; Gao et al., 2024), showing particular promise for educational applications. However, their tendency to hallucinate (Henkel et al., 2024) remains a significant barrier to reliable use in learning environments, especially in mathematics education where accuracy is crucial (Frieder, 2023). In mathematical reasoning, LLMs struggle with precision, often failing to process complex symbols and formulas correctly (Lewkowycz et al., 2022), which risks misleading students when these models are deployed as educational tools.

Retrieval-Augmented Generation (RAG) has emerged as a powerful, plug-and-play solution to mitigate hallucination by grounding LLMs in external knowledge, which has been widely employed in constructing foundation models (Chen et al., 2024) and practical agents (Arslan et al., 2024). Unlike fine-tuning, RAG enhances model performance without extensive retraining, making it both efficient and cost-effective. The main paradigm of RAG involves first calculating the similarities between a question and chunks in an external knowledge corpus, followed by incorporating the top K relevant chunks into the prompt to guide the LLMs (Lewis et al., 2020).

Despite the advantages of RAG, selecting the appropriate embedding models remains a crucial concern, as the quality of retrieved references directly influences the generation results of the LLM (Tu et al., 2025). Due to variations in training data and architecture, different embedding models capture semantic similarity in divergent ways. A model that excels on one type of mathematical problem may underperform on another, making the selection of a single optimal embedding model a difficult and uncertain choice. Consequently, this fundamental limitation makes vanilla RAG unreliable for comprehensive mathematics education applications.

Inspired by these developments, we address the embedding selection challenge by proposing two methods that combine the benefits of multiple embedding models to improve RAG in mathematics education. The first method is named Mixture-Embedding RAG, which sorts the retrieved materials from multiple embedding models based on normalized similarity and selects the top K materials as final references. The second method is named Confident RAG, where we first utilize vanilla RAG to generate answers multiple times, each time employing a different embedding model and recording the associated confidence metrics. We then select the answer with the highest confidence level as the final response. By validating our approach using multiple LLMs and embedding models, we illustrate the superior performance and generalization of Confident RAG, providing a reliable foundation for AI-powered mathematics education. The main contributions of this paper can be summarized as follows:

- A critical barrier to deploying LLMs in mathematics education with RAG is that different embedding models operate within their own prior domains. To leverage the strengths of various embedding models, we propose and test two novel RAG methods: Mixture-Embedding RAG and Confident RAG. These methods effectively utilize the retrieved results from different embedding models to their fullest extent.
- While Mixture-Embedding RAG performs similarly to vanilla RAG, the Confident RAG method exhibits superior performance compared to both the vanilla LLM and vanilla RAG, with average improvements of 9.9% and 4.9%, respectively, when using the best confidence metric. Additionally, we discuss the optimal number of embedding models for the Confident RAG method based on the results.
- Our results reveal two outstanding confidence metrics: self-certainty and Distributional Perplexity (DP), both showing average improvements of approximately 10% compared to the vanilla LLM. Specifically, among the LLMs examined, self-certainty achieves a maximum increase of 10.4%, while DP demonstrates a maximum increase of 12.4% compared to the vanilla LLM. The reasons behind the better performance of these two metrics are discussed based on their formulas.

2 Preliminary and Related Work

2.1 Retrieval Augmented Generation (RAG)

When constructing domain-specific foundation models or private agents, modifying a trained LLM is often necessary. However, the computational and resource costs associated with full fine-tuning render it impractical for a large number of users. Even though techniques such as Low-Rank Adaptation (LoRA) and prefix-tuning provide lighter training paradigms, RAG offers its own advantages in terms of plug-and-play capabilities. Furthermore, RAG allows external knowledge to remain independent of LLMs, facilitating the replacement of databases with other domains or more up-to-date versions. Currently, a common paradigm of vanilla RAG is illustrated in Fig. 1 and Appendix A.1.

Since embedding models can produce different similarity calculations, many studies have innovated in retriever embedding models to improve the performance of the RAG pipeline. For example, Invar-RAG (Liu et al., 2024) constructs an LLM-based retriever with LoRA-based representation learning and an invariance loss, directly addressing retriever embedding variance and locality, and demonstrates improved retrieval and response accuracy. Blended RAG (Sawarkar et al., 2024) combines dense semantic embeddings with sparse retrievers, empirically showing that this hybrid embedding space achieves higher QA accuracy, especially as corpus size scales. W-RAG (Nian et al., 2024) further validates that enhancing retriever embeddings using weak signals from the downstream QA task offers gains comparable to methods requiring full human annotations.

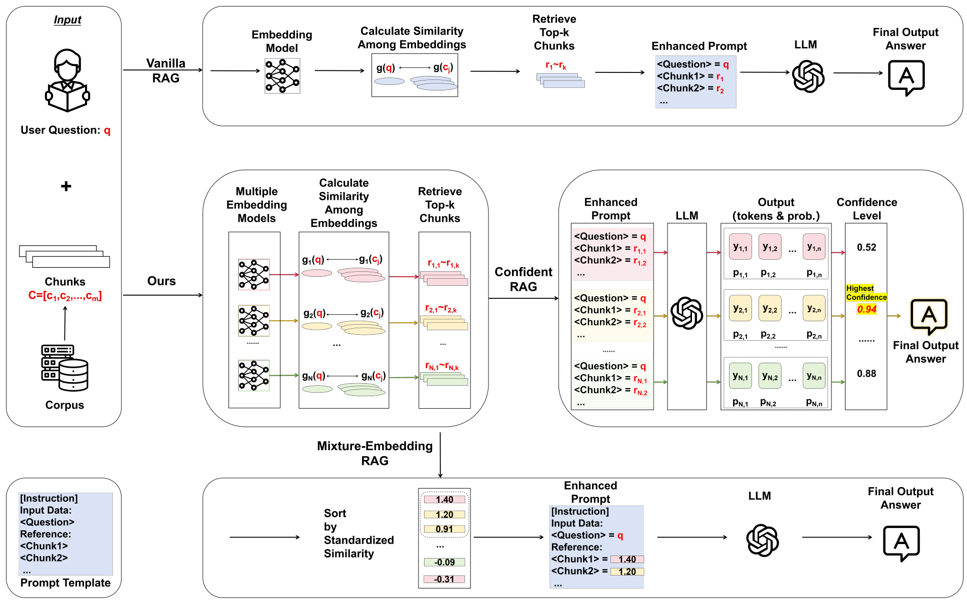

Figure 1: Workflow: This figure illustrates the workflows of vanilla RAG, as well as our proposed Mixture-Embedding RAG and Confident RAG.

However, most studies have focused on the innovation of a single embedding model. This presents a critical limitation for mathematical reasoning, where effective retrieval requires a nuanced understanding of logical structures, symbols, and problem-solving strategies. Since different embedding models capture these mathematical semantics in varied and non-uniform ways, selecting a single optimal model is inherently uncertain. Our approach will leverage the strengths of various embedding models to provide the most relevant information for LLMs.

2.2 Confidence of LLMs

The confidence level of the LLM output can indicate the response quality to some extent. Several approaches have been employed to measure model confidence. Some of them depend on model prompting, external similarity or model-expressed judgments. These include linguistic confidence measures (Shrivastava et al., 2023; Xiong et al., 2023) and LLM-as-judge or self-evaluation strategies (Shrivastava et al., 2025; Ren et al., 2023; Hager et al., 2025) However, these methods are often poorly calibrated, not directly related to the true predictive uncertainty of the model, and can be biased or inconsistent due to their reliance on language generation rather than statistical likelihood.

In contrast, probability-based metrics directly ground into models' token probability output (Chen & Mueller, 2023; Jiang et al., 2021; Lin et al., 2024; Chen et al., 2025). These include average log-probability, entropy-based measures and self-certainty (Kang et al., 2025). Currently, generative LLMs operate based on an auto-regressive process, which is shown in Appendix A.2.

Many studies have observed that the probability of each token reflects the LLMs' confidence. Intuitively, if the selected token has a higher chosen probability, or if the generated probability vector exhibits a less uniform distribution, the LLMs demonstrate higher confidence in their generated results. Our study will use the confidence measurement metrics summarized by Kang et al. (2025), which are Average Log-Probability (AvgLogP), Gini Impurity (Gini), Entropy, Distributional Perplexity (DP) and Self-Certainty. Detailed definitions of these metrics are shown in Appendix A.3.

3 METHODOLOGY

Based on the understanding that different embedding models excel in their respective domains, our study aims to develop a comprehensive RAG method that combines the strengths of various embedding models.

3.1 WORKFLOW OVERVIEW

In this work, given a student's mathematical question q, we utilize a single LLM f() for answer generation, while N embedding models $\{\mathrm{g}_1(), \mathrm{g}_2(), \ldots, \mathrm{g}_N()\}$ assist in retrieving question-related information from a corpus involving mathematics textbooks and question-answer pairs, which has been divided into chunks $C = [c_1, c_2, \ldots, c_m]$. This setup mirrors a scenario where a student seeks help from a tutor who consults multiple specialized reference books. Inspired by the stacking and bagging paradigms of ensemble learning, we propose two distinct methods: (1) Mixture-Embedding RAG, which first combines the retrieved results from multiple embedding models and incorporates them into the prompt for the LLM, and (2) Confident RAG, which employs vanilla RAG with different embedding models for initial answer generation and subsequently selects the answer with the highest confidence as the final response.

As shown in Fig. 1, the two proposed methods share the same preliminary process: first, they utilize different embedding models to identify the k chunks with the highest similarity, similar to the approach taken by the vanilla RAG described in Section 2.1. We define the retrieved chunks as $R = \{r_{i,j} \mid i = 1, 2, \ldots, N; j = 1, 2, \ldots, k\}$, where $r_{i,j}$ is defined as $c_{\mathcal{K}^i_j}$ and $\mathcal{K}^i$ is the index set of the selected k chunks by embedding model g_i. Additionally, we define the similarity between the question q and chunk $r_{i,j}$, calculated by embedding model g_i, as $w_{i,j}$. The differences between the Mixture-Embedding RAG and Confident RAG are detailed as follows.

3.2 MIXTURE-EMBEDDING RAG

In this method, we aim to directly find the chunks with the highest similarity to the question among the candidates selected by all embedding models. The goal is to mimic an expert tutor who synthesizes information from multiple sources to explain a concept. Intuitively, we can select chunks using the similarity score $w_{i,j}$. However, this approach can lead to two problems. First, the candidates from different embedding models may have repetitions. Therefore, we select chunks without repetition to avoid any performance loss. Second, due to the different structures of embedding models, they may have varying ranges of similarity, so that simply comparing similarity across different models may introduce potential bias. To address this, we propose to standardize the similarity score using Z-scores:

$$\hat{w}_{i,j} = \frac{w_{i,j} - \mu_i}{\sigma_i} \,, \tag{1}$$

where μ_i and σ_i represent the mean and standard deviation of all similarity scores from embedding model g_i.

3.3 CONFIDENT RAG

Inspired by Kang et al. (2025), we first let the LLM f generate N separate answers, using the retrieved chunks from the N embedding models separately. Then, we evaluate the confidence of the N answers using the metrics mentioned in Section 2.2. Finally, we select the answer with the highest confidence as our final answer for each question. This method is particularly valuable in educational contexts where a single incorrect solution can significantly impact student learning. By selecting the answer the LLM is most certain about, we provide a safer, more reliable mathematical assistant for students.

4 EXPERIMENT

4.1 EXPERIMENT SETUP

This section provides an overview of the experiment setup, including the model configuration, dataset, corpus, embedding models, and large language models.

- **Model configuration**: Utilizing six Nvidia A2 GPUs, it took approximately fourteen days to obtian the initial results from our experiment. These included vanilla LLM, vanilla RAG with different embedding models, mixture-embedding RAG with various combinations of embedding models, and the combined results of confident RAG method.
- **Dataset**: Gsm8k (Cobbe et al., 2021) is a dataset of approximately 8,500 high-quality, linguistically diverse grade school math word problems designed to support question answering on basic mathematical tasks. These problems typically require 2 to 8 steps of reasoning, mainly involving elementary arithmetic operations, and are solvable by middle school students. The solutions are provided in natural language, emphasizing sequential reasoning rather than complex concepts or variables. The train dataset of Gsm8k was used as the retrieved courpus for RAG, while the first 500 items from the test dataset were used as user questions.
- **Corpus**: Two types of corpora were selected for retrieval: (1) Mathematics textbooks from OpenStax[1] covering domains such as calculus, algebra and trigonometry. The content was segmented into sub-section. (2) Math QA items from the Gsm8k train dataset. One textbook sub-section and three QA items will be retrieved according to the similarity based on their similarity to the user-provided math question each time.
- **Embedding models**: We chose four embedding models for encoding, including: all-MiniLM-L6-v2 (Face, n.d.), ModernBERT-large (Warner et al., 2024), MathBERT (Peng et al., 2021) and stsb-roberta-large (Reimers & Gurevych, 2019) models.
- **LLMs**: To validate our proposed method and ensure repeatability, we selected three LLMs, following (Shao et al., 2025): Qwen2.5-Math-7B (Yang et al., 2024), Llama-3.1-8B (Grattafiori et al., 2024) and OLMo-2-1124-7B (OLMo et al., 2024).

4.2 EXPERIMENT RESULT

4.2.1 PERFORMANCE OF VANILLA RAG AND VANILLA LLMS

Table 1 presents a comparison of three different LLMs with and without RAG using different embedding models. All models achieve significant improvements in accuracy after applying RAG method, with an average improvement of 5%. The accuracy of Qwen2.5-Math-7B increases from 75.2% to 80.5%, Llama-3.1-8B from 16.6% to 21.3%, and OLMo-2-1124-7B from 21.0% to 26.0%. This confirms that grounding LLMs in external knowledge is a viable strategy for improving their reliability in mathematical contexts.

However, the performance of vanilla RAG was unstable when using different embedding models. This limitation can be vividly illustrated in Table 4 in Appendix A.4 for a 400-meter hurdle word problem. The vanilla LLM, without any retrieved context, produced an incorrect answer of 34.2. When augmented with RAG, the results were different: using embedding model 1 or 4 led to the same incorrect answer (34.2), while models 2 and 3 successfully produce the right answer (36).

4.2.2 PERFORMANCE OF MIXTURE-EMBEDDING RAG

The Mixture-Embedding RAG method aimed to solve this instability by fusing retrieved documents. However, as shown in Table 2, the average accuracies of the three LLMs did not perform well compared to vanilla RAG. Llama-3.1-8B and OLMo-2-1124-7B demonstrated similar accuracy as vanilla RAG, with the differences ranging from -0.2% to 0.5%. Surprisingly, Qwen2.5-Math-7B showed a decrease of 5.5% compared to vanilla RAG.

The example in Appendix A.4 reveals the potential pedagogical danger of this approach. While using 2 embedding models (2 Embs) yielded the correct answer (36), fusing 3 or 4 embedding

[1] `https://openstax.org/subjects/math`

LLM Model	Vanilla LLM	Vanilla RAG					
		Emb1	Emb2	Emb3	Emb4	Avg	Improvement
Qwen2.5-Math-7B (Yang et al., 2024)	75.2%	81.0%	83.0%	77.6%	80.2%	80.5%	5.3%↑
Llama-3.1-8B (Grattafiori et al., 2024)	16.6%	21.8%	19.8%	19.4%	24.0%	21.3%	4.7%↑
OLMo-2-1124-7B (OLMo et al., 2024)	21.0%	26.0%	28.4%	25.0%	24.6%	26.0%	5.0%↑
Average	37.6%	42.9%	43.7%	40.7%	42.9%	42.6%	5.0%↑

Table 1: Model Performance Comparison Across Different Embedding Methods: The four embedding models (1–4) correspond to all-MiniLM-L6-v2 (Face, n.d.), ModernBERT-large (Warner et al., 2024), MathBERT (Peng et al., 2021), and stsb-roberta-large (Reimers & Gurevych, 2019), respectively, which will remain consistent in the following tables. The last two columns represent the average accuracy of RAG using the four embedding models and the improvement compared to the vanilla LLM. These details will remain the same in the subsequent tables.

models (3 Embs, 4 Embs) resulted in incorrect answer (34.2). This suggests that the fusion process can sometimes amplify noisy or incorrect information, leading the LLM to a wrong conclusion and therefore resulting in a critical failure for educational tools designed to build correct conceptual understanding.

LLM Model	Mix-Embedding RAG					
	2 Embs	3 Embs	4 Embs	Avg	v.s. Vanilla LLM	v.s. Vanilla RAG
Qwen2.5-Math-7B (Yang et al., 2024)	74.20%	76.00%	74.80%	75.0%	-0.2%↓	-5.5%↓
Llama-3.1-8B (Grattafiori et al., 2024)	20.90%	19.60%	22.80%	21.1%	4.5%↑	-0.2%↓
OLMo-2-1124-7B (OLMo et al., 2024)	26.20%	26.80%	26.40%	26.5%	5.5%↑	0.5%↑
Average	40.4%	40.8%	41.3%	40.9%	3.3%↑	-1.7%↓

Table 2: Performance of Mixture-Embedding RAG: This table illustrates the performance when using retrieved information from 2 to 4 different embedding models randomly (denoted as 2 Embs to 4 Embs), comparing the results with those of the vanilla LLM and vanilla RAG.

4.2.3 Performance of Confident RAG

The Confident RAG method demonstrated superior and consistent performance, addressing the core instability of previous approaches. After using different embedding models for RAG in multiple rounds, the results of Confident RAG method are shown in Table 3. Four main findings were obtained:

(1) The accuracy of Confident RAG method surpasses that of both the vanilla RAG (with an average improvement ranging from 3.2% to 4.9%) and vanilla LLM (with an average improvement ranging from 8.1% to 9.9%).

(2) For each LLM, after applying the Confident RAG method, the accuracy improved by nearly 10% when using the best confidence metric compared to the vanilla LLM.

(3) Among all confidence metrics, self certainty and distributional perplexity demonstrated the best performance, with average improvements of 9.9% and 9.7%, respectively, over the vanilla LLM. These two metrics also performed well across different LLMs. For instance, there was an increase of 9.1% for Qwen2.5-Math-7B using self certainty as the confidence metric, an increase of 10.4% for Llama-3.1-8B with both metrics, and an increase of 12.3% for OLMo-2-1124-7B using distributional perplexity.

The practical mechanism behind this success is clearly demonstrated in our running example (Appendix A.4). For the hurdle problem, Confident RAG generated multiple candidate answers. Crucially, the self-certainty metric correctly assigned a high score (16.76) to the correct answer (36) and significantly lower scores (9.3) to the incorrect one (34.2). In every combination of embedding models, we selected the answer with the high-confidence, which can largely minimizing the risk of propagating pedagogical errors and present the most reliable solution to the students.

(4) Regarding the optimal number of embedding models (N) for multi-rounds, no evidence suggests that a larger n yields better accuracy. In our experiments, the accuracy when N=3 was always larger than when N=2. Meanwhile, the accuracy for N=3 and N=4 was similar, with a maximum

LLM	Emb. Model	AvgLogP	Self-certainty	Gini	Entropy	DP
Qwen2.5-Math-7B (Yang et al., 2024)	1,2	82.0%	**85.0%**	81.8%	82.8%	83.6%
	1,3	79.4%	**83.4%**	79.2%	79.6%	81.0%
	1,4	81.4%	**84.6%**	81.6%	81.8%	82.2%
	2,3	81.8%	**84.4%**	81.4%	81.8%	82.4%
	2,4	81.4%	**83.6%**	81.0%	81.4%	82.2%
	3,4	79.2%	**82.2%**	79.8%	79.8%	80.6%
	Avg (n=2)	80.9%	**83.9%**	80.8%	81.2%	82.0%
	1,2,3	79.4%	**85.0%**	79.0%	79.8%	81.6%
	1,2,4	79.6%	**85.0%**	79.6%	80.6%	82.0%
	1,3,4	79.0%	**84.8%**	79.6%	80.0%	80.8%
	2,3,4	79.2%	**84.4%**	79.0%	79.8%	81.0%
	Avg (n=3)	79.3%	**84.8%**	79.3%	80.1%	81.4%
	1,2,3,4	78.2%	**84.8%**	78.4%	79.2%	80.6%
	Avg (n=2,3,4)	80.1%	**84.3%**	80.0%	80.6%	81.6%
	v.s. Vanilla RAG	0.4%↓	**3.8%**↑	0.4%↓	0.1%↑	1.2%↑
	v.s. Vanilla LLM	4.9%↑	**9.1%**↑	4.8%↑	5.4%↑	6.4%↑
Llama-3.1-8B (Grattafiori et al., 2024)	1,2	**27.2%**	**27.2%**	27.0%	27.0%	**27.2%**
	1,3	26.8%	26.8%	26.6%	27.0%	**27.2%**
	1,4	**26.8%**	26.4%	26.4%	26.4%	26.4%
	2,3	23.2%	**24.0%**	**24.0%**	**24.0%**	23.6%
	2,4	26.6%	**26.8%**	26.4%	26.0%	26.6%
	3,4	25.8%	26.4%	**26.6%**	**26.6%**	26.4%
	Avg (n=2)	26.1%	**26.3%**	26.2%	26.2%	26.2%
	1,2,3	28.6%	29.2%	28.8%	29.4%	29.4%
	1,2,4	**28.8%**	28.6%	28.0%	28.0%	28.2%
	1,3,4	27.4%	27.4%	27.2%	**27.6%**	**27.6%**
	2,3,4	25.8%	**26.4%**	26.0%	26.0%	**26.4%**
	Avg (n=3)	27.7%	**27.9%**	27.5%	27.8%	**27.9%**
	1,2,3,4	**27.8%**	27.6%	27.0%	27.4%	27.6%
	Avg (n=2,3,4)	26.8%	**27.0%**	26.7%	26.9%	**27.0%**
	v.s. Vanilla RAG	5.6%↑	**5.7%**↑	5.5%↑	5.6%↑	**5.7%**↑
	v.s. Vanilla LLM	10.2%↑	**10.4%**↑	10.1%↑	10.3%↑	**10.4%**↑
OLMo-2-1124-7B (OLMo et al., 2024)	1,2	31.0%	31.2%	30.4%	31.2%	32.2%
	1,3	29.8%	29.6%	29.8%	**30.0%**	**30.0%**
	1,4	29.4%	29.2%	28.6%	29.6%	**31.0%**
	2,3	31.6%	30.8%	30.2%	31.4%	**32.8%**
	2,4	32.6%	32.0%	31.0%	**33.0%**	33.8%
	3,4	29.6%	30.2%	29.6%	30.6%	**31.4%**
	Avg (n=2)	30.7%	30.5%	29.9%	31.0%	**31.9%**
	1,2,3	32.6%	32.0%	31.4%	32.2%	**33.2%**
	1,2,4	32.8%	32.8%	31.6%	33.2%	**34.8%**
	1,3,4	31.2%	31.2%	30.6%	32.0%	**34.6%**
	2,3,4	32.6%	32.2%	30.8%	33.6%	**36.6%**
	Avg (n=3)	32.3%	32.1%	31.1%	32.8%	**34.8%**
	1,2,3,4	32.8%	32.4%	31.0%	33.2%	**35.8%**
	Avg (n=2,3,4)	31.5%	31.2%	30.5%	31.8%	**33.3%**
	v.s. Vanilla RAG	5.5%↑	5.2%↑	4.5%↑	5.8%↑	**7.3%**↑
	v.s. Vanilla LLM	10.5%↑	10.2%↑	9.5%↑	10.8%↑	**12.3%**↑
Average	Avg (n=2)	45.9%	**46.9%**	45.6%	46.1%	46.7%
	Avg (n=3)	46.4%	**48.3%**	46.0%	46.9%	48.0%
	Avg (n=4)	46.3%	**48.3%**	45.5%	46.6%	48.0%
	Avg (n=2,3,4)	46.1%	**47.5%**	45.7%	46.4%	47.3%
	v.s. Vanilla RAG	3.5%↑	**4.9%**↑	3.2%↑	3.9%↑	4.7%↑
	v.s. Vanilla LLM	8.5%↑	**9.9%**↑	8.1%↑	8.8%↑	9.7%↑

Table 3: Accuracy Comparison Across Multi-RAG with Different Embedding Models: Avg(n) denotes the average accuracy across different combinations of n embedding models. Each line uses underline to indicate the best embedding combination within each LLM. Each row uses **bold** to signify the best metric for confidence evaluation.

difference of 1% across the three LLMs. Considering factors such as time cost, GPU capacity, and other constrains, this finding suggests that researchers should seek a suitable trade-off based on their own condition when choosing the optimal N.

5 Analysis

5.1 Mixture-embedding RAG Showed Limited Improvements from Vanilla RAG

Our results indicate that Mixture-Embedding RAG did not provide the anticipated improvement. For general LLMs (e.g., Llama-3.1-8B and OLMo-2-1124-7B) without math fine-tuning, their internal math knowledge is limited, leading to lower accuracy in direct answer generation (Lewkowycz et al., 2022). In these cases, even noisy references retrieved by RAG are more reliable than the LLMs' own outputs, as RAG at least provides partially correct information. However, while the mixture-embedding RAG method may optimize the retrieval ranking process and improve the quality of the references, the general LLMs' capabilities prevent them from fully leveraging higher-quality references, resulting in performance similar to vanilla RAG. Additionally, if different embedding models return highly diverse references, directly combining the top-ranked documents may cause information overload or contextual confusion, negating the potential benefits of mixture-embedding method. Therefore, the performance of general LLMs matches that of vanilla RAG rather than surpassing it.

On the other hand, for the LLMs that have been fine-tuned based on math corpora (e.g., Qwen2.5-Math-7B), vanilla RAG may result in smaller improvements, and lower-quality references can lead to poorer answer performance (Lewis et al., 2020). For these types of LLMs, the mixture-embedding method may introduce additional noise in mathematical contexts, resulting in lower accuracy compared to vanilla RAG. The decline in accuracy may be caused by several factors: (1) Mathematical symbols and formulas may vary drastically across embedding models, making similarity calculations unstable. (2) Different models may encode math terms differently, causing the top-ranked reference to be suboptimal. (3) An embedding model might incorrectly rank an irrelevant math material highly, while better references from other models are ignored. If the LLM generates hallucinated answers based on incorrect references, its performance can degrade below that of the vanilla LLM. (4) Information overload or contextual confusion may also occur, similar to what happens with general LLMs.

5.2 Confident RAG Demonstrated Consistent Improvements through Confidence Selection

As shown in Figure 2, there exists a positive correlation between confidence and accuracy, which also aligns with the finding from Chen et al. (2025). Therefore, the Confident RAG method improves overall accuracy by integrating multiple embedding models to generate answers and selecting the highest-confidence results using the most effective metric. This process effectively filters out low-confidence incorrect answers. Meanwhile, the combined effect of RAG and confidence filtering enhances the robustness, leading to significant improvements compared to vanilla LLMs. When employing the optimal confidence metric, all LLMs achieved an accuracy increase of nearly 10%, demonstrating the method's universality. In the experiments, when the number of embedding models N ¿ 3, the accuracy improvement became limited, likely due to redundant or noisy retrievals introduced by additional models. At N=3, the method achieved an optimal balance between diversity and computational efficiency. Increasing the number of models further yields only marginal benefits.

Self-Certainty and DP outperform other metrics since they directly measure the concentration and divergence of the probability distribution. Specifically, Self-Certainty measures how far the predicted distribution deviates from uniform. By scaling the probabilities by $|v|$ and taking the negative logarithm, it heavily penalizes uniform-like distributions, favoring sharp peaks. This makes it highly discriminative for high-confidence answers. Additionally, DP is an exponential version of entropy. The exponentiation amplifies differences in entropy, making it more sensitive to the sharpness of the distribution. Low DP values indicate tightly clustered high-probability tokens, which strongly correlate with correct answers. In contrast, other metrics are less sensitive because they either average out uncertainties (AvgLogP) or lack normalization across vocabularies (Gini). While entropy can be useful, it is linear and less discriminative compared to DP's exponential scaling. Therefore, Self-Certainty and DP are more sensitive to subtle variations in model confidence.

6 LIMITATIONS AND FUTURE WORK

This study has two main limitations. First, the experiment was only designed for mathematics domain using the GSM8K benchmark and a corpus of mathematical textbooks. This design prevents generalization of our findings to other subjects in educational context. Future study should validate the Confident RAG framework across diverse academic domains. This will test its robustness and determine whether it is a truly general-purpose solution for educational AI.

Second, this study provides a validated methodological framework (Confident RAG) that researchers and developers can reference to build more reliable AI math assistants. However, we have not yet embedded this method into a complete, user-facing educational system. Therefore, its usability and practical impact in a real learning environment remain unexplored. Future study should build a functional AI math tutoring system that incorporates the Confident RAG framework. This will allow us to validate the method's effectiveness not just on benchmark accuracy, but on its ability to support student learning through interactive use, and to refine its design based on user feedback from actual classrooms.

7 CONCLUSION

In this paper, we proposed two novel RAG-enhanced approaches using multiple embedding models to improve the mathematics question response performance of LLMs: (1) Mixture-Embedding RAG and (2) Confident RAG. Our findings clearly indicate that Confident RAG is a superior and robust solution, with an average improvement of approximately 10% while utilizing the best confidence metric, compared to vanilla LLM. It selects the highest-confidence response from multiple RAG outputs, each generated using a different embedding model. The consistent success of confidence metrics like self-certainty and DP provides a reliable mechanism for identifying correct answers. Crucially, the number of embedding models (N) can be determined by researchers based on their specific conditions, as no evidence suggesting that a larger N necessarily yields a better performance. In contrast, Mixture-Embedding RAG showed no improvement over vanilla RAG, by sorting and selecting retrieved references from different embedding models based on their standardized similarity before guiding LLM. The findings indicate that Confident RAG can be served as a plug-and-play method for enhancing LLM performance in mathematics question answering.

ACKNOWLEDGMENTS

This work was supported by General Research Fund by the Hong Kong Research Grants Council under Grant 17611725.

REFERENCES

Muhammad Arslan, Saba Munawar, and Christophe Cruz. Sustainable digitalization of business with multi-agent rag and llm. *Procedia Computer Science*, 246:4722–4731, 2024.

Haolong Chen, Hanzhi Chen, Zijian Zhao, Kaifeng Han, Guangxu Zhu, Yichen Zhao, Ying Du, Wei Xu, and Qingjiang Shi. An overview of domain-specific foundation model: key technologies, applications and challenges. *SCIENCE CHINA Information Sciences*, 2024.

Jiuhai Chen and Jonas Mueller. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness. *arXiv preprint arXiv:2308.16175*, 2023.

Tingwei Chen, Jiayi Chen, Zijian Zhao, Haolong Chen, Liang Zhang, and Guangxu Zhu. First token probability guided rag for telecom question answering. *arXiv preprint arXiv:2501.06468*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Hugging Face. sentence-transformers/all-minilm-l6-v2, n.d. URL `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`.

Simon Frieder. Mathematical capabilities of chatgpt. *arXiv preprint arXiv:2301.13867*, 2023.

Lin Gao, Jing Lu, Zekai Shao, Ziyue Lin, Shengbin Yue, Chiokit Ieong, Yi Sun, Rory James Zauner, Zhongyu Wei, and Siming Chen. Fine-tuned large language model for visualization system: A study on self-regulated learning in education. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

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

Sophia Hager, David Mueller, Kevin Duh, and Nicholas Andrews. Uncertainty distillation: Teaching language models to express semantic confidence. *arXiv preprint arXiv:2503.14749*, 2025.

Owen Henkel, Zach Levonian, Chenglu Li, and Millie Postle. Retrieval-augmented generation to improve math question-answering: Trade-offs between groundedness and human preference. In *Proceedings of the 17th International Conference on Educational Data Mining*, pp. 315–320, 2024.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*, 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.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Contextualized sequence likelihood: Enhanced confidence scores for natural language generation. *arXiv preprint arXiv:2406.01806*, 2024.

Ziwei Liu, Liang Zhang, Qian Li, Jianghua Wu, and Guangxu Zhu. Invar-rag: Invariant llm-aligned retrieval for better generation. *arXiv preprint arXiv:2411.07021*, 2024.

Tengfei Lyu, Siyuan Feng, Hao Liu, and Hai Yang. Llm-oddr: A large language model framework for joint order dispatching and driver repositioning. *arXiv preprint arXiv:2505.22695*, 2025.

Jinming Nian, Zhiyuan Peng, Qifan Wang, and Yi Fang. W-rag: Weakly supervised dense retrieval in rag for open-domain question answering. *arXiv preprint arXiv:2408.08444*, 2024.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. Mathbert: A pre-trained model for mathematical formula understanding. *arXiv preprint arXiv:2105.00377*, 2021.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL `http://arxiv.org/abs/1908.10084`.

Jie Ren, Yao Zhao, Tu Vu, Peter J Liu, and Balaji Lakshminarayanan. Self-evaluation improves selective generation in large language models. In *Proceedings on*, pp. 49–64. PMLR, 2023.

Kunal Sawarkar, Abhilasha Mangal, and Shivam Raj Solanki. Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers. In *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 155–161. IEEE, 2024.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in rlvr, 2025. URL `https://arxiv.org/abs/2506.10947`.

Vaishnavi Shrivastava, Percy Liang, and Ananya Kumar. Llamas know what gpts don't show: Surrogate models for confidence estimation. *arXiv preprint arXiv:2311.08877*, 2023.

Vaishnavi Shrivastava, Ananya Kumar, and Percy Liang. Language models prefer what they know: Relative confidence estimation via confidence preferences. *arXiv preprint arXiv:2502.01126*, 2025.

Yiteng Tu, Weihang Su, Yujia Zhou, Yiqun Liu, and Qingyao Ai. Rbft: Robust fine-tuning for retrieval-augmented generation against retrieval defects. *arXiv preprint arXiv:2501.18365*, 2025.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024. URL `https://arxiv.org/abs/2412.13663`.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Zijian Zhao, Dian Jin, Zijing Zhou, and Xiaoyu Zhang. Automatic stage lighting control: Is it a rule-driven process or generative task? *arXiv preprint arXiv:2506.01482*, 2025.

A APPENDIX

A.1 VANILLA RAG

A common paradigm of vanilla RAG is illustrated in Fig.1. Given a question q, an external corpus C divided into chunks as $C = [c_1, c_2, \dots, c_m]$, and an embedding model g(), we first generate the embeddings for the question and the corpus, respectively, as follows:

$$\begin{aligned} e_q &= \mathrm{g}(q) \ , \\ e_{c_j} &= \mathrm{g}(c_j) \ , \end{aligned} \tag{2}$$

where $e_q, e_{c_j} \in \mathbb{R}^{d_g}$ are the embedding results for question q and chunk c_j, and d_g is the output dimension of the model g().

Based on the understanding that two sentences with higher similarity yield more similar embeddings, we calculate the similarity between the question and each chunk using the cosine similarity metric, defined as:

$$s_{q,c_j} = \frac{e_q \cdot e_{c_j}}{\|e_q\| \|e_{c_j}\|} \ , \tag{3}$$

where s_{q,c_j} represents the cosine similarity between question q and chunk c_j, $\cdot$ denotes the dot product, and $\|\cdot\|$ denotes the Euclidean norm. Finally, we select k chunks with the highest similarity and incorporate them into the prompt, expressed as:

$$\mathrm{h}(t, q, c_j | j \in \mathcal{K}) \ , \tag{4}$$

where h() is a montage function that combines the question and the retrieved chunks using a specified template t, and $\mathcal{K}$ represents the index set of the selected k chunks. In this way, we construct a prompt that includes external knowledge from the corpus C that is most similar to question q.

A.2 AUTO-REGRESSIVE PROCESS OF GENERATIVE LLMS

Currently, generative LLMs operate based on an auto-regressive process, formulated as:

$$y_i \sim \mathrm{f}(x, y_{<i}) \ , \tag{5}$$

where f() is an LLM, x is the prompt, $y_{<i}$ represents the first generated $i-1$ tokens, and y_i denotes the i^{th} token. The output of f() is the probability distribution over each word in the vocabulary, from which y_i is sampled according to this probability vector. This process can be viewed as a Markov Decision Process (MDP), where the state consists of the prompt x along with the currently generated tokens $y_{<i}$, and the action space corresponds to the vocabulary, allowing each token to be chosen with a probability given by

$$p(v_j | x, y_{<i}; \pi_f) = \mathrm{f}(x, y_{<i})[j] \ , \tag{6}$$

where v_j is the j^{th} element in the vocabulary v, and π_f represents the strategy. For simplicity, we will neglect the π_f term in the following discussion.

A.3 CONFIDENCE METRICS USED IN THIS STUDY

- **Average Log-Probability (AvgLogP)**: This metric computes the mean logarithmic likelihood of the generated tokens, reflecting the model's confidence in the entire sequence:

$$\mathrm{AvgLogP} = \frac{1}{n} \sum_{i=1}^{n} \log p(y_i | x, y_{<i}) \ , \tag{7}$$

  where n represents the answer length. Higher log-probabilities suggest more reliable predictions.
- **Gini Impurity (Gini)**: This metric measures the concentration of the predicted distribution:

$$\mathrm{Gini} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{|v|} \left(p(v_j | x, y_{<i})\right)^2 \ , \tag{8}$$

  where $|v|$ represents the vocabulary size. Higher values indicate more peaked distributions, reflecting greater certainty in the model's predictions.

- **Entropy**: This metric computes the uncertainty of the predicted distribution:

$$\text{Entropy} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{|v|} - p(j|x, y_{<i}) \log p(v_j|x, y_{<i}) \ . \tag{9}$$

  Lower entropy signifies higher confidence, as it indicates a more deterministic output distribution.

- **Distributional Perplexity (DP)**: This metric generalizes perplexity to the entire predicted distribution:

$$\text{DP} = \frac{1}{n}\sum_{i=1}^{n}\exp\left(-\sum_{j=1}^{|v|} p(v_j|x, y_{<i}) \log p(v_j|x, y_{<i})\right) \ . \tag{10}$$

  Lower values of distributional perplexity indicate that the model is more confident about its predictions across the entire distribution, as it reflects a clearer understanding of token relationships.

- **Self-Certainty**: This metric assesses the KL-divergence from a uniform distribution:

$$\text{Self-certainty} = -\frac{1}{n|v|}\sum_{i=1}^{n}\sum_{j=1}^{V} \log\left(|v| \cdot p(v_j|x, y_{<i})\right) \ . \tag{11}$$

  Higher self-certainty values reflect greater confidence in the predictions, as they indicate a more concentrated distribution of outputs.

The Cumulative Distribution Function (CDF) relationship between these metrics and accuracy is provided in Fig. 2, where we utilize the negative of Entropy and DP.

A.4 Sample generation results

Question	Correct Answer	Model	Answer	Confidence (self-certainty)	Accuracy (0=incorrect; 1=correct)
Lee used to be able to run the 400-meter hurdles two seconds faster than Gerald would run the 400-meter hurdles. But Gerald changed his diet, which improved his speed by 10%. If Lee runs the 400-meter hurdles in 38 seconds, how fast can Gerald, with his improved diet, run the 400-meter hurdles, in seconds?	36	**Vanilla LLM**	34.2	/	0
		Vanilla RAG			
		Emb1	34.2	9.1794	0
		Emb2	36	16.7625	1
		Emb3	36	15.9693	1
		Emb4	34.2	9.3341	0
		Mixture-embedding RAG			
		2 Embs	36	/	1
		3 Embs	34.2	/	0
		4 Embs	34.2	/	0
		Confident RAG			
		1,2	36	16.7625	1
		1,3	36	15.9693	1
		1,4	34.2	9.3341	0
		2,3	36	16.7625	1
		2,4	36	16.7625	1
		3,4	36	15.9693	1
		1,2,3	36	16.7625	1
		1,2,4	36	16.7625	1
		1,3,4	36	15.9693	1
		2,3,4	36	16.7625	1
		1,2,3,4	36	16.7625	1

Table 4: Sample generation results for an item: The LLM used in this example is Qwen2.5-Math-7B. The four embedding models (1-4) are the same as the previous table. The Mixture-embedding RAG results illustrate the performance when using retrieved information from 2 to 4 different embedding models randomly (denoted as 2 Embs to 4 Embs). The Confident RAG results illustrate the performance when using different combinations of multiple embedding models and using self-certainty as the confidence metric.

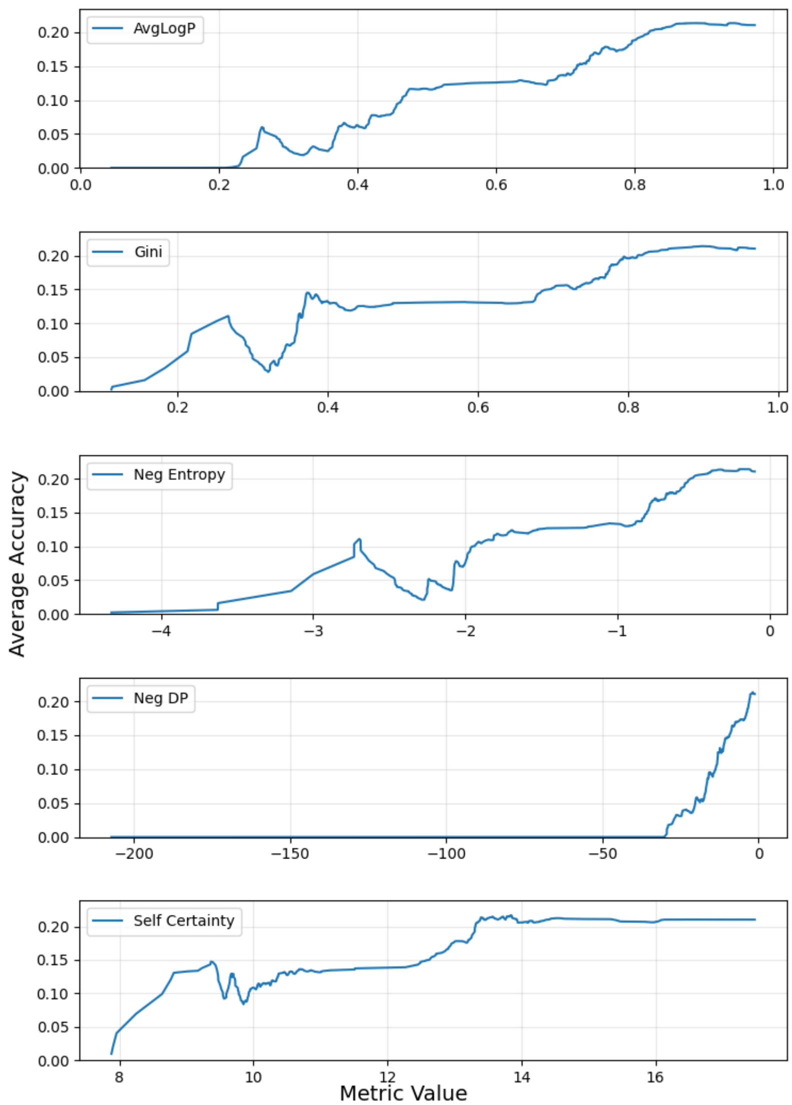

Figure 2: CDF of Accuracy for Different Metrics: The lines have been smoothed using a Gaussian filter.