Harmonizing Diverse Models: A Layer-wise Merging Strategy for Consistent Generation

Xujun Peng, Anoop Kumar, Jingyu Wu, Parker Glenn, Daben Liu
AI Foundations, Capital One
McLean, VA, USA
{ xujun.peng, anoop.kumar, jingyu.wu, parker.glenn, daben.liu}@capitalone.com

Abstract

Retrieval-Augmented Generation (RAG) systems leverage Large Language Models (LLMs) to generate accurate and reliable responses that are grounded in retrieved context. However, LLMs often generate inconsistent outputs for semantically equivalent inputs, a problem compounded by the scarcity of consistency-focused training data and the limitations of current fine-tuning techniques in enhancing output consistency. We propose a new approach combining systematic synthetic data generation, triplet loss for better embeddings, and a novel layer-wise model merging approach. Using consistency-aware weights derived from intermediate layer activations, our method effectively integrates knowledge from specialized models. Experimental results how that our merged model significantly enhances output consistency, achieving a 47.5% improvement in response similarity over the baseline, thus offering a practical solution for increasing the reliability of an industrial RAG system.

1 Introduction

LLMs have demonstrated remarkable capabilities in natural language understanding and generation, enabling breakthroughs across a broad spectrum of applications such as question answering and summarization. RAG has emerged as a powerful paradigm that combines the generative strength of LLMs with external knowledge retrieval to enhance factuality, reduce hallucination, and extend context beyond model limitations (Lewis et al., 2020; Wu et al., 2024).

Despite their potential, RAG systems often generate inconsistent responses, for minor and semantically insignificant variations in the input query or the prompt (Song and Zheng, 2024). This inconsistency manifests itself in various forms, including contradictory responses, variability in factual grounding, and fluctuations in the level of detail or confidence expressed by the model. This unpredictability not only undermines the reliability of RAG systems but also poses challenges for their adoption in high-stakes or knowledge-sensitive domains such as finance, healthcare, and scientific research. As shown in Figure 1, the mere presence or absence of a question mark can dramatically alter the response of a RAG based QA system. In an industrial production deployment, there could several such variations in how users query the system, posing challenges in the adoption of RAG systems.

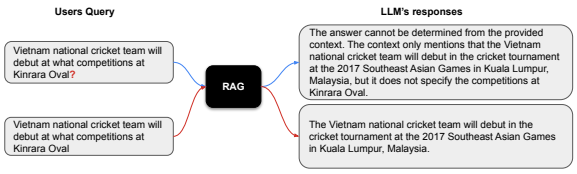

Figure 1: Variability in LLM Responses from Subtle Query Differences.

In RAG systems, two key models work together: the retriever and the generator. The retriever is responsible for fetching relevant content based on a user query or prompt, while the generator creates a coherent and contextually appropriate answer by leveraging both the query and the retrieved content.

Inconsistencies may arise during either the retrieval or the generation process, leading to varied responses. However, our empirical observations and Zuccon et al. (2016), Abdallah et al. (2025) indicate that generators are more sensitive and less consistent than retrievers to minor variations in queries. While retrievers tend to be consistent, even in the face of minor, semantically insignificant variations in the input query (e.g., different phrasings of the same question), generators exhibit higher variability (Cao et al., 2025). Small changes in phrasing or query structure can lead to different answers being generated, even when the retrieved content remains the same. This difference in behavior highlights the challenges faced by generative

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 954–962
November 4-9, 2025

models in maintaining consistency, especially in tasks where precise and coherent responses are required.

In this work, we focus on characterizing, measuring, and mitigating such inconsistency. Our main contributions are as follows:

1. We characterize different types of query variations that lead to inconsistent responses in an industrial RAG system.

2. We identify metrics and demonstrate inconsistency in the question answering task.

3. We develop novel layer-wise merging approach to reduce inconsistency without regressing on accuracy.

2 Related Work

We review the literature to establish a clear understanding of what constitutes consistency, how it is measured, and strategies to improve consistency in context of LLMs. A widely accepted concept, as proposed by Patwardhan et al. (2024), defines consistency as the semantic equivalence of two responses generated by the same LLM when prompted with identical or semantically similar queries multiple times.

Evaluating and improving LLM consistency remains challenging due to a lack of targeted benchmarks and datasets, prompting research into specialized evaluation methods. Elazar et al. (2021) contributed a valuable resource for factual consistency measurement in pre-trained LLMs and a novel consistency loss to enhance performance even on new data. To explore prompt sensitivity, Raj et al. (2025b) introduced PromptSET, Qiang et al. (2024) applied perturbations to highlight the difficulties in predicting how minor prompt changes affect LLMs. For evaluation, Zhao et al. (2024) propose an automated consistency assessment tool using a custom data set. Addressing inconsistency in natural-language explanations, Chen et al. (2025a) developed EC-finetuning, that trains on synthetic data to increase consistency.

To address the need for reliable LLM output in NLG, especially under semantically equivalent inputs, Raj et al. (2023) proposed a framework to measure semantic consistency via output agreement, showing strong correlation with human judgments across domains. Complementing this, Wu et al. (2025) introduced a logit-based ensemble method aligned with human perceptions through a user study. Lee et al. (2025) examined LLM consistency as automated evaluators, focusing on their reliability in scoring identical items.

Recent work has focused on actively improving LLM consistency. Raj et al. (2025a) used multi-step prompting and synthetic data for semantic alignment. Sathe et al. (2025) enhanced self-consistency via multiple input tokenizations. Wang et al. (2023) improved reasoning by sampling diverse outputs and selecting by voting.

3 Methodology

In this section, we describe our methodology for improving the consistency of LLM responses, specifically within the context of a RAG system.

We observe that the retriever in our RAG system provides accurate context, but minor variations in the query often lead to inconsistent response from the generator. This work focuses on improving generator consistency under such variations, assuming stable retrieval quality. Addressing retrieval inconsistency is left for future work.

Our work aims to improve the RAG generator's consistency. By analyzing human-annotated data, we constructed diverse synthetic datasets to train multiple individual generator models. To achieve higher response consistency, we developed a novel consistency-focused, layer-wise model merging approach, building upon DARE-TIES (Yu et al., 2024; Yadav et al., 2023). This strategy allowed us to effectively combine knowledge from individual models trained on diverse synthetic data.

3.1 Synthetic Data Generation

A direct approach to improving LLM consistency is to train on all possible input variations. However, this is impractical due to the vast number of potential variations and limited data availability - especially in domains like healthcare, where data is fragmented and complex, and finance, where privacy regulations constrain access.

Given these limitations, synthetic data provides a practical way to improve consistency when real data is scarce. While prior work has documented a broad range of general query variations - such as typos, synonyms, keyboard proximity errors, and paraphrases (Zhang et al., 2025; Wu et al., 2025) - it does not capture several nuanced variations observed in large-scale industrial RAG systems. Our analysis of production queries shows that a small set of key variations accounts for most input diver-

Table 1: Illustrative Query Variations Leading to Different Answers.

Variation Type	Query	Query'
Variation - How do/to	**how do we** manage customer feedback at end of project	**how to** manage customer feedback at end of project
I vs. we	can **we** drive to a grocery store	can **I** drive to a grocery store
Singular vs. plural	delivering **packages** for shipment	delivering **package** for shipment
Article omissions	how to add **a contact** to **a** phone book	how to add **contacts** to phone **books**

sity, often involving subtle rephrasings (e.g., "how to we manage an account" vs. "how to manage an account") rather than simple surface-level errors. Table 1 lists these variation types with representative examples. Characterizing and accounting for such variations is critical to improving system robustness and building user trust in real-world RAG applications.

Based on the analysis of our dataset, we've identified three main types of query variations:

How to/do variations: These queries often involve rephrasing questions about methods or actions. We used regular expression rules to systematically create additional queries of this nature.

Singular/Plural/Article variations: This category covers changes in noun quantity (e.g., "apple" vs. "apples") and the use of articles (e.g., "a", "an", "the"). To synthesize more of these variations, we randomly interchanged singular and plural forms and substituted or modified articles.

Semantic variations: These are changes in wording that maintain the same core meaning but use different vocabulary or phrasing. For semantic variations, we leveraged a pretrained LLM (Llama-3.1-70B-Instruct) to paraphrase our queries (Grattafiori et al., 2024).

We used these synthetic queries to run our IR system, capturing updated contexts for our RAG system. This process generated enriched training and test datasets with a wide array of input variations. Rather than training/fine-tuning a single LLM with all the real-world and synthetic data, we opted to train/fine-tune multiple specialized models, each focusing on a different category of input variations. This approach allows each model to excel at the specific underlying tasks associated with its particular query type.

3.2 Triplet Loss Training

Unlike traditional LLM fine-tuning that relies solely on cross-entropy loss, we incorporate triplet loss during our fine-tuning phase.

Triplet Loss (Schroff et al., 2015a) is a widely used loss function in metric learning, used in tasks such as face recognition and semantic search, to learn embeddings that pull similar items closer while pushing dissimilar ones apart. The core idea of Triplet Loss is to train on triplets of data points: an anchor A, a positive P that is similar to the anchor, and a negative N that is dissimilar. The objective is to ensure that the distance between A and P is smaller than that between A and N. The triplet Loss function is formulated as:

$$L(A, P, N) = \max(0, d(f(A), f(P)) - d(f(A), f(N)) + \alpha) \quad (1)$$

More details of triplet loss can be found in (Schroff et al., 2015b; Reimers and Gurevych, 2019).

In our implementation, triplets were constructed by first choosing an anchor query (A). We then selected its corresponding positive (P) and negative (N) data points by randomly sampling from its top 10 and bottom 10 nearest neighbors, respectively, within the feature space generated by a semantic feature extractor.

The final loss function employed during our training and fine-tuning process is a combination of cross-entropy loss and triplet loss, defined as:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \alpha \cdot \mathcal{L}_{\text{Triplet}}, \quad (2)$$

where α is a predefined weighting factor designed to balance the contribution of triplet loss.

3.3 Model Merging

With a suite of specialized models, each trained on distinct synthetic datasets, the challenge became generating a single, consistent response without sacrificing accuracy. While conventional ensemble approaches for multiple pre-trained or fine-tuned LLMs involve parallel execution and output combination, they incur significant computational costs and inference latency (Chen et al., 2025b).

To address these limitations, model merging offers a solution by consolidating knowledge from multiple pre-trained or fine-tuned models into a single consolidated model. These techniques range from simple averaging to complex algorithms that

align features and selectively transfer knowledge. Here, we introduce a novel model merging approach, building on the DARE-TIES merge method (Yu et al., 2024; Yadav et al., 2023), with the main goal of substantially boosting the consistency of the unified model's responses.

DARE-TIES merging is a sophisticated model merging algorithm designed to overcome the limitations of simple weight averaging, especially when combining fine-tuned models that originate from a common pre-trained base model but have diverged during training on different tasks or datasets. It operates on the principle of merging the $\Delta\theta_k = \theta_{F_k} - \theta_P$ that fine-tuned models apply to a common pre-trained model, rather than directly merging the absolute weights, where θ_P is the base model's parameters and θ_{F_k} denotes the parameters of the k-th fine-tuned model. By applying sparsification, sign matching and inverse-scaling on the $\Delta\theta_k$, DARE-TIES yields the merged model's parameters by:

$$\theta_{\text{merged}} = \theta_P + \sum_{k=1}^{N} \Delta\theta_k. \tag{3}$$

To improve consistency with semantically identical inputs, we analyzed the consistency of each LLM layer, then assigned dynamic weights in Equation 3 for merging.

To accomplish this, we first formed a development set $\mathcal{S}_{dev}$ of T diverse data points. Then, for each model k and each layer l, we extracted the activations $\alpha_k^{(l)} \in \mathbb{R}^{D\times T}$ from development set $\mathcal{S}_{dev}$, where D represents the output feature dimension of layer l. For sequential outputs, we used max-pooling to extract these activations. This process enabled us to compute a similarity matrix $\Sigma_k^{(l)} \in \mathbb{R}^{T\times T}$ for the activations of each data point at every layer of model k.

Ideally, a model exhibiting high consistency with semantically identical inputs should produce similar activations within a single layer. Conversely, if inputs are semantically distinct, their activations should diverge significantly. Therefore, a consistent model would ideally yield similar similarity matrices $\Sigma_k^{(l)}$ across different layers when presented with the same set of inputs.

Leveraging this intuition, we can quantify a model's consistency by comparing the $\Sigma_k^{(l)}$ from different layers. Our approach begins by using a semantic feature extractor (specifically, a sentence transformer) to obtain features for each query in our development set, $\mathcal{S}_{dev}$. From these features, we computed a reference similarity matrix Σ_r. Subsequently, we quantified the discrepancy between each layer's similarity matrix $\Sigma_k^{(l)}$ and this reference using the absolute difference: $d_k^{(l)} = |\Sigma_k^{(l)} - \Sigma_r|$. Hence, for a specific layer l across our various LLMs, we obtain a set of distance values, $\text{DM}^{(l)} = [d_1^{(l)}, d_2^{(l)}, \ldots, d_N^{(l)}]$. To convert these distances into weights that indicate a layer's contribution to consistency, we apply an inverted non-linear normalization approach. First, we computed the inverted distance for each layer's distance $d_k^{(l)}$ by subtracting it from the maximum distance observed for that layer across all models:

$$\tilde{d}_k^{(l)} = \max(\text{DM}^{(l)}) - d_k^{(l)}$$

Next, these inverted distances are normalized to obtain $r_k^{(l)}$:

$$r_k^{(l)} = \frac{\tilde{d}_k^{(l)}}{\sum_{j=1}^{N} \tilde{d}_j^{(l)}}$$

Finally, we apply a sigmoid function to these normalized inverted distances to derive the final weight $w_k^{(l)}$ for layer l of model k:

$$w_k^{(l)} = \sigma(a \cdot r_k^{(l)} + b) \tag{4}$$

Here, $\sigma(\cdot)$ denotes the sigmoid function, and a and b are predefined scaling and offset parameters.

Based on the derived consistency-oriented layer weights $w_k^{(l)}$ for each model k, we modified Equation 3 to incorporate these weights into the layer-wise model merging process:

$$\theta_{\text{merged}}^{(l)} = \theta_P^{(l)} + \sum_{k=1}^{N} w_k^{(l)} \cdot \Delta\theta_k^{(l)}. \tag{5}$$

The final merged LLM is constructed by applying Equation 5 in a layer-wise manner.

We outline the algorithm for model merging:

Algorithm 1 Consistency-Aware Model Merging

Input: Base model θ_P, fine-tuned models $\{\theta_{F_k}\}_{k=1}^{N}$, dev set $\mathcal{S}_{dev}$
Output: Merged model θ_{merged}
Compute reference similarity matrix Σ_r using a sentence encoder on $\mathcal{S}_{dev}$
for each model k and layer l **do**
    Extract activations and compute similarity matrix $\Sigma_k^{(l)}$
    Compute distance $d_k^{(l)} = |\Sigma_k^{(l)} - \Sigma_r|$
end for
for each layer l **do**
    Normalize distances $d_k^{(l)}$ to weights $w_k^{(l)}$ using inverted scaling and sigmoid
    Merge: $\theta_{\text{merged}}^{(l)} = \theta_P^{(l)} + \sum_k w_k^{(l)} \cdot (\theta_{F_k}^{(l)} - \theta_P^{(l)})$
end for
return θ_{merged}

4 Experiments

Our experimental setup utilized a QA engine built on a RAG architecture. For the evaluation of our consistency improvement method, the retriever component was held constant, and the generator component underwent fine-tuning.

4.1 Datasets

To fine-tune and evaluate our LLM generator, we used 2,738 representative queries and their retrieved contexts that resemble a production IR system. Domain experts annotated the expected answers, and the data was split into 1,421 training and 1,317 test samples.

To get more varied inputs for training our model, we applied the methods detailed in Section 3.1 to create three distinct types of synthetic data. Our synthetic training dataset included 150 "how to/do" variation queries, 1,421 paraphrased queries, and 952 singular/plural/article variation queries. We submit all query variations to the IR system to retrieve their corresponding contexts, which are then used to construct the final inputs.

Alongside the 1,317 test samples to measure accuracy, we created a test set to evaluate consistency using our data synthesis methods (Section 3.1). This produced 1,579 variations-176 "how to/do", 912 paraphrases, and 491 singular/plural/article changes-paired with original queries and expected answers for consistency testing.

4.2 Metrics

To assess the overall accuracy of the results of our RAG system, we employed the ROUGE-L (Lin, 2004) and BLEU metrics with up to 4-grams (Papineni et al., 2002), comparing the LLM-generated responses against the references provided.

To quantify the consistency of LLM response across input variations, we utilized three metrics: exact string match (EM), response similarity (RS) and Bert similarity (BS) measures. Given an original query Q and its variant Q', with S and S' representing the respective LLM responses, the exact string match is formally defined as:

$$\text{EM}(S, S') \iff S = S'.$$

For Response Similarity (RS), we determine semantic equivalence by thresholding the ROUGE score between the LLM's responses S and S':

$$RS(S, S') \iff \text{Rouge}(S, S') > T,$$

where T represents an empirically determined threshold used to ascertain whether two responses are considered semantically identical.

Furthermore, we define Bert Similarity (BS) between two LLMs responses S and S' to quantify the semantic similarity of them, as:

$$BS(S, S') \iff \text{Bert}(S, S').$$

4.3 Model Training and Merging

Our experimental setup involved several distinct fine-tuning stages for the Llama-3.1-8B-Instruct model (Grattafiori et al., 2024) and Gemma-3-12B-Instruct model (Team et al., 2025).

We started by fine-tuning a baseline Llama-3.1-8B-Instruct and Gemma-3-12B-Instruct models for two epochs, using the original 1,421 training samples and only a cross-entropy loss function.

To investigate how triplet loss could boost LLM consistency, all subsequent fine-tuning experiments combined both cross-entropy loss and triplet loss, keeping the hyperparameters consistent with our initial baseline setup.

Following this, we fine-tuned five distinct Llama-3.1-8B-Instruct LLMs. One was fine-tuned on our base training set exclusively. The other three were fine-tuned on this base set, each augmented with a specific synthetic data type: 176 "how to/do" variations, 912 paraphrased samples, or 491 singular/plural/article variations (more details on this in Section 3.2). The final model was fine-tuned using

Table 2: Comparison of Overall Accuracy and Consistency Metrics.

	Llama-3.1-8B-Instruct based LLMs					Gemma-3-12B-Instruct based LLMs				
	ROUGE	BLEU	EM	RS	BS	ROUGE	BLEU	EM	RS	BS
B	0.5123	0.2928	0.1051	0.2799	0.9246	0.4692	0.2338	0.0678	0.2609	0.9227
B + SFT	0.5208	0.3125	0.1482	0.3325	0.9266	0.5266	0.3297	0.2242	0.4009	0.9323
B + SFT + TL	0.5460	0.3460	0.1822	0.3530	0.9276	0.5206	0.3194	0.2331	0.4041	0.9337
B + SFT + TL + HTD	**0.5493**	**0.3495**	0.2250	0.3867	0.9264	0.5276	0.3255	0.2483	0.4364	0.9351
B + SFT + TL + SEM	0.5330	0.3339	0.2366	0.3965	0.9281	0.4966	0.3042	0.2673	0.4262	0.9314
B + SFT + TL + SPA	0.5364	0.3370	0.2111	0.3692	0.9262	0.5130	0.3170	0.2603	0.4231	0.9332
B + SFT + TL + ALL	0.5198	0.3230	0.2510	0.3986	0.9289	0.4879	0.2974	**0.3382**	**0.4731**	0.9357
Merged	0.5379	0.3380	**0.2521**	**0.4129**	**0.9292**	**0.5356**	**0.3416**	0.2932	0.4674	**0.9373**

Abbreviations: B = Baseline (Llama-3.1-8B-Instruct or Gemma-3-12B-Instruct), SFT = Supervised Fine-tuned, TL = Triplet-loss, HTD = "How to/do" variation, SEM = Semantic variation, SPA = Singular/Plural/Article variation, ALL = All training data, Merged = Merged model.

all available training data combined. Finally, we merged these three individually fine-tuned LLMs using the methodology described in Section 3.3. We repeated the same fine-tuning and merging steps for the Gemma-3-12B-Instruct LLMs to ensure consistent evaluation across model's architectures.

All fine-tuned models, including the Llama-3.1-8B-Instruct and Gemma-3-12B-Instruct baselines, were comprehensively evaluated in two dedicated test sets designed to assess both accuracy and consistency measures, as described in Section 4.1. We present complete experimental results in Table 2.

4.4 Results

In Table 1, we present four types of query variations that lead to response inconsistency. Table 2 quantitatively shows that the baseline model (Llama-3.1-8B-Instruct) achieves moderate overlap with human references (ROUGE: 0.5123, BLEU: 0.2928) but demonstrates the lowest consistency (EM: 0.1051, RS: 0.2799, BS: 0.9246). This demonstrates the model often fails to generate consistent responses to semantically equivalent queries.

The fine-tuned model, as shown in Table 2, demonstrates a modest improvement over the baseline, w.r.t accuracy and consistency. While it yields somewhat better text overlap and initial gains in consistency, its performance, particularly in EM, RS and BS, suggests that general fine-tuning provides only limited progress towards truly consistent response for varied inputs.

Incorporating triplet loss significantly boosts performance across all metrics, as seen by comparing the triplet-loss model to the fine-tuned model in Table 2. The triplet-loss model achieved higher ROUGE (0.5460) and BLEU (0.3460) scores, indicating better content and lexical alignment. The model shows improvement in consistency, the EM score dramatically improved by 73.4% to 0.1822 (from 0.1051), while RS score saw a substantial 26.1% increase to 0.3530 (from 0.2799). These results underscore the effectiveness of integrating triplet loss in fine-tuning strategies for LLMs, leading to significantly more robust and consistent response generation.

As shown in the Table 2, individual variation models—specifically the *How to/Do*,*Semantic*, and *Singular/Plural/Article* variation models—consistently outperform the baseline in both accuracy and consistency. This demonstrates the effectiveness of specialized fine-tuning with synthetically generated data

Surprisingly, models fine-tuned on individual synthetic datasets outperformed the combined-data model in accuracy. However, the combined model achieved higher consistency, suggesting that merging diverse variation types may introduce conflicting signals or biases that impact accuracy.

The merged model consistently delivers the most robust and balanced performance across all metrics, with notable strength in response consistency. In terms of consistency metrics, the merged model achieves the highest scores for both EM at 0.2521, RS at 0.4129 and BS at 0.9292. This performance significantly surpasses all other models. For EM, it represents an impressive 139.87% improvement over the baseline model and still leads the next best model ("B + SFT + TL + ALL" model at 0.2510) by approximately 0.44%. Similarly, its RS score is a 47.52% improvement over the baseline and approximately 3.59% higher than the second best model. This indicates that the merging strategy is highly effective in ensuring LLM responses are more reliably identical or semantically equivalent even when faced with varied inputs.

Regarding accuracy-based metrics, while the merged model's ROUGE score (0.5379) and BLEU score (0.3380) are marginally lower than the top performer (the "B + SFT + TL + HTD" with ROUGE 0.5493 and BLEU 0.3495), they are still very good and highly competitive. This demonstrates that the merging process successfully enhances consistency without a significant trade-off in overall accuracy or fluency. The merged model effectively combines the strengths of its constituent specialized models, making it the most well-rounded and high-performing solution for both accurate and consistent RAG system responses.

Table 2 also reports accuracy and consistency metrics for Gemma-3-12B-Instruct models. Overall, the trend of model improvements closely mirrors that of the Llama-3.1-8B-Instruct experiments: baseline models exhibit moderate ROUGE and BLEU scores with the lowest consistency (EM: 0.0678, RS: 0.2609, BS: 0.9227), fine-tuning improves both accuracy and consistency, incorporating triplet loss further boosts response reliability, and models fine-tuned on individual synthetic variations outperform the baseline in both accuracy and consistency. For Gemma, the "B + SFT + TL + ALL" model achieves the highest consistency metrics (EM: 0.3382, RS: 0.4731), similar to the trend observed for Llama, where combined-data models also prioritize consistency over raw accuracy. The merging strategy consistently delivers the most robust and balanced performance across all metrics.

Key differences are notable, however. The Gemma baseline shows lower initial accuracy and EM than the Llama baseline, suggesting a larger model does not automatically guarantee consistent responses. The merged Gemma model attains the highest ROUGE (0.5356), BLEU (0.3416), and BS (0.9373), slightly outperforming Llama's merged model on accuracy and semantic similarity, though EM is slightly lower than Gemma's "B + SFT + TL + ALL" model, indicating a minor trade-off in exact match consistency.

Overall, while the pattern of improvement, baseline $\rightarrow$ fine-tuned $\rightarrow$ triplet-loss $\rightarrow$ specialized variation $\rightarrow$ merged, is consistent across both model families, the larger Gemma-3-12B-Instruct benefits more from combined data fine-tuning, achieving higher accuracy and semantic similarity, while the merging strategy ensures robust and balanced performance similar to both Llama and Gemma models.

5 Conclusion

In this work, we identify four types of semantically insignificant query variations that cause inconsistent LLM responses. We quantify response similarity and show that baseline models and standard fine-tuning exhibit low consistency. To address this, we propose a novel approach combining synthetic data generation, Triplet Loss training, and layer-wise model merging guided by consistency-oriented weights.

Our experiments show the merged model significantly outperforms baselines and specialized models, achieving superior Exact Match and Response Similarity scores, thus demonstrating enhanced consistency while maintaining strong accuracy. This work presents a compelling pathway towards more trustworthy LLMs and opens avenues for future research, including adaptive merging, expanded consistency definitions, and the application of this method to diverse public datasets. We also plan to construct and publicly release benchmarks that mimic the identified query variations to further evaluate and demonstrate the effectiveness of our approach. Additional future directions include addressing inconsistency cases arising from retrievers, which were beyond the scope of this study.

6 Limitations

While the proposed work has been evaluated on industrial data, there is scope to create public benchmark and evaluate the method. We will explore creating creating benchmarks for evaluating consistency in responses due to variations in the query.

The work is limited to variations in the query where the retrievers results don't significantly change. This can explored as future direction for research.

Finally, we experimented with 2 Large Language Models with optimal settings for fine-tuning. There is scope to explore additional hyper parameter configurations.

References

Abdelrahman Abdallah, Jamshid Mozafari, Bhawna Piryani, Mohammed Ali, and Adam Jatowt. 2025. From retrieval to generation: Comparing different approaches. *arXiv preprint arXiv:2502.20245*.

Tianyu Cao, Neel Bhandari, Akhila Yerukola, Akari Asai, and Maarten Sap. 2025. Out of style: Rag's fragility to linguistic variation. *arXiv preprint arXiv:2504.08231*.

Yanda Chen, Chandan Singh, Xiaodong Liu, Simiao Zuo, Bin Yu, He He, and Jianfeng Gao. 2025a. Towards consistent natural-language explanations via explanation-consistency finetuning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7558–7568.

Zhijun Chen, Jingzheng Li, Pengpeng Chen, Zhuoran Li, Kai Sun, Yuankai Luo, Qianren Mao, Dingqi Yang, Hailong Sun, and Philip S Yu. 2025b. Harnessing multiple large language models: A survey on llm ensemble. *arXiv preprint arXiv:2502.18036*.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Noah Lee, Jiwoo Hong, and James Thorne. 2025. Evaluating the consistency of LLM evaluators. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10650–10659.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Aditya Patwardhan, Vivek Vaidya, and Ashish Kundu. 2024. Automated Consistency Analysis of LLMs . In *2024 IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, pages 118–127.

Yao Qiang, Subhrangshu Nandi, Ninareh Mehrabi, Greg Ver Steeg, Anoop Kumar, Anna Rumshisky, and Aram Galstyan. 2024. Prompt perturbation consistency learning for robust language models. *arXiv preprint arXiv:2402.15833*.

Harsh Raj, Vipul Gupta, Domenic Rosati, and Subhabrata Majumdar. 2025a. Improving consistency in large language models through chain of guidance. *Preprint*, arXiv:2502.15924.

Harsh Raj, Vipul Gupta, Domenic Rosati, and Subhabrata Majumdar. 2025b. Semantic consistency for assuring reliability of large language models. *Preprint*, arXiv:2308.09138.

Harsh Raj, Domenic Rosati, and Subhabrata Majumdar. 2023. Measuring reliability of large language models through semantic consistency. *Preprint*, arXiv:2211.05853.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Ashutosh Sathe, Divyanshu Aggarwal, and Sunayana Sitaram. 2025. Improving consistency in LLM inference using probabilistic tokenization. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4766–4778.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015a. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015b. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

Mingyang Song and Mao Zheng. 2024. A survey of query optimization in large language models. *arXiv preprint arXiv:2412.17558*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, and 1 others. 2024. Retrieval-augmented generation for natural language processing: A survey. *arXiv preprint arXiv:2407.13193*.

Xiaoyuan Wu, Weiran Lin, Omer Akgul, and Lujo Bauer. 2025. Estimating llm consistency: A user baseline vs surrogate metrics. *Preprint*, arXiv:2505.23799.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: resolving interference when merging models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 7093 – 7115.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning*, pages 57755 – 57775.

Kepu Zhang, Zhongxiang Sun, Weijie Yu, Xiaoxue Zang, Kai Zheng, Yang Song, Han Li, and Jun Xu. 2025. QE-RAG: A robust retrieval-augmented generation benchmark for query entry errors. *arXiv preprint arXiv:2504.04062*.

Fufangchen Zhao, Guoqiang Jin, Jiaheng Huang, Rui Zhao, and Fei Tan. 2024. Consistency matters: Explore llms consistency from a black-box perspective. *Preprint*, arXiv:2402.17411.

Guido Zuccon, Joao Palotti, and Allan Hanbury. 2016. Query variations and their effect on comparing information retrieval systems. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 691–700.