

Article

Uncertainty-Aware Expert Allocation for Efficient Multitask Fine-Tuning of Large Language Models

Maab Elhassan ¹, Minhee Jun ² and Hanseok Ko ^{2,*}¹ Electrical and Computer Engineering, Korea University, Seoul 02841, Republic of Korea; maab98@korea.ac.kr² Electrical and Computer Engineering, Catholic University of America, Washington, DC 20064, USA; junm@cua.edu

* Correspondence: koha@cua.edu

Abstract

Large Language Models increasingly rely on Mixture-of-Experts architectures to scale model capacity while controlling computational cost. However, most MoE systems employ static expert routing strategies that allocate identical computational resources to all tokens, regardless of their uncertainty or difficulty. This paper proposes Confidence-Based Dynamic Routing (CBDR), an uncertainty-aware expert allocation mechanism for multitask fine-tuning of large language models. CBDR estimates token-level confidence from the entropy of the gating distribution and dynamically adjusts the number of activated experts accordingly. Tokens with high confidence are processed with fewer experts, while uncertain tokens receive additional expert capacity. To mitigate expert imbalance, we introduce Confidence-Based Importance Balancing (CBIB), a training objective that encourages balanced expert utilization. Experiments on multiple biomedical question-answering datasets demonstrate that CBDR improves model performance across multiple biomedical QA datasets, achieving absolute improvements of up to 3% in F1 score on the BioASQ benchmark while maintaining efficient expert utilization. These results suggest that uncertainty-aware routing provides a practical mechanism for adaptive compute allocation in sparse large language models.

Dataset: BioASQ (<http://bioasq.org>), PubMedQA (<https://pubmedqa.github.io>), RealMedQA (<https://huggingface.co/datasets/k2141255/RealMedQA>) (access on 7 September 2024).

Dataset License: no license required (publicly available)

Keywords: large language models; multitask fine-tuning; mixture-of-experts; low-rank adaptation; expert routing



Academic Editor: Joaquin Torres
Sospedra

Received: 12 March 2026

Revised: 19 June 2026

Accepted: 25 June 2026

Published: 10 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Large language models (LLMs) have transformed natural language processing by enabling powerful representation learning across a wide range of tasks including question answering, text generation, machine translation, and conversational AI. The success of these models is largely attributed to the transformer architecture introduced by Vaswani et al. [1] and subsequent scaling of model parameters and training datasets. Recent models such as GPT-3 [2], GPT-4 [3], and PaLM demonstrate that increasing model scale leads to substantial improvements in performance across many tasks.

Despite their success, large language models present significant challenges in computational efficiency and scalability. Training and fine-tuning large models require substantial computational resources, which limits their accessibility and applicability in many real-world settings. In particular, adapting LLMs to domain-specific tasks often requires updating billions of parameters, which can be computationally prohibitive.

To address these challenges, two complementary research directions have emerged: parameter-efficient fine-tuning (PEFT) and conditional computation architectures.

Parameter-efficient fine-tuning methods such as adapters [4], prefix tuning [5], and Low-Rank Adaptation (LoRA) [6] significantly reduce the number of trainable parameters required to adapt large models to downstream tasks. Among these approaches, LoRA has gained widespread adoption because it introduces low-rank updates to weight matrices while keeping the original pretrained parameters frozen.

At the same time, Mixture-of-Experts (MoE) architectures provide an effective mechanism for scaling neural networks through conditional computation. Instead of activating the entire model for every input, MoE models consist of multiple expert networks, and a gating mechanism dynamically selects a subset of experts for each token [7]. This design enables models to increase capacity while maintaining manageable computational cost.

Recent large-scale models such as Switch Transformer [8] and GShard [9] demonstrate the effectiveness of MoE architectures for scaling neural networks. However, most MoE models rely on static routing strategies, typically implemented using a fixed top-k expert selection rule. Under this approach, the same number of experts is activated for every token regardless of input complexity.

In practice, tokens in natural language sequences vary significantly in difficulty and uncertainty. Simple tokens may require only minimal computation, while more complex tokens may benefit from additional expert processing. Static routing therefore allocates identical computational resources to all tokens, potentially leading to inefficient computation and suboptimal performance. More importantly, static routing prevents MoE models from allocating computational resources according to token uncertainty. In practice, different tokens may require different levels of reasoning and model capacity. An adaptive routing mechanism that allocates computation based on token uncertainty can therefore improve both efficiency and performance in large-scale language models.

To address this limitation, we propose a Confidence-Based Dynamic Routing (CBDR) mechanism that allocates computational resources according to token-level uncertainty. The key idea is to estimate the confidence of expert assignments using the entropy of the gating probability distribution. Tokens with high confidence are processed using fewer experts, while tokens with lower confidence receive additional expert capacity.

In addition, we introduce a Confidence-Based Importance Balancing (CBIB) loss to mitigate expert imbalance during training. This loss encourages balanced expert utilization while preserving the advantages of dynamic routing.

The main contributions of this work are summarized as follows:

1. We introduce an uncertainty-aware compute allocation framework for MoE-based large language models that dynamically adjusts computational resources according to token-level confidence.
2. We propose a Confidence-Based Dynamic Routing (CBDR) mechanism that dynamically determines the number of activated experts based on the entropy of the gating distribution.
3. We introduce a Confidence-Based Importance Balancing (CBIB) loss to mitigate expert imbalance and improve routing stability.

4. Extensive experiments on three biomedical question-answering datasets demonstrate that the proposed method consistently improves performance compared with static expert routing.

The remainder of this paper is organized as follows. Section 2 reviews related work on large language models, parameter-efficient fine-tuning, and MoE architectures. Section 3 introduces the proposed CBDR framework and the CBIB loss. Section 4 presents the experimental setup, evaluation results, and analysis of expert utilization. Finally, Section 5 concludes the paper and discusses potential directions for future work.

2. Related Work

2.1. Large Language Models and Transformer Architectures

LLMs have become the dominant paradigm in modern natural language processing. Their success is primarily driven by the transformer architecture introduced by Vaswani et al. [1], which replaces recurrent structures with self-attention mechanisms that allow efficient modeling of long-range dependencies in text sequences, as shown in Figure 1. The transformer architecture enables parallel computation over input tokens and significantly improves scalability compared with recurrent neural networks.

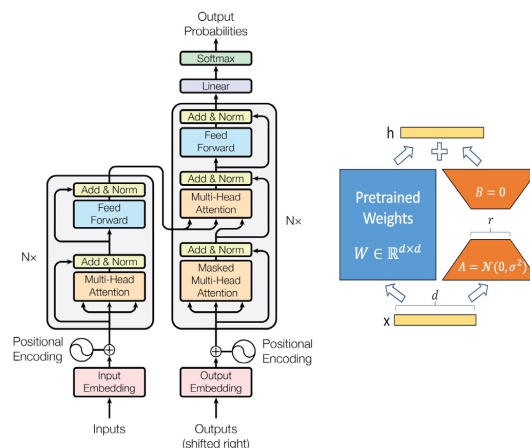


Figure 1. Transformer architecture [1] (the left) and LoRA adaptation [6] (the right).

Early transformer-based models such as BERT [10] demonstrated the effectiveness of bidirectional contextual representations for natural language understanding tasks. Subsequent generative models such as GPT [11], GPT-2 [12], GPT-3 [2], and GPT-4 [3] further demonstrated that scaling transformer models to billions or trillions of parameters leads to substantial improvements in performance across diverse tasks including language modeling, summarization, translation, and question answering.

The rapid progress in LLMs has also been supported by empirical scaling laws. Kaplan et al. [13] showed that model performance improves predictably with increased model size, dataset size, and computational resources. These findings motivated the development of increasingly large language models that rely on massive training datasets and computational infrastructure.

Despite these advances, the rapid growth of model size introduces significant challenges. Training and fine-tuning models with billions of parameters require extensive computational resources and memory. In many applications, particularly domain-specific tasks such as biomedical text analysis, adapting large models to specialized datasets becomes computationally expensive and difficult to deploy. These challenges have motivated research in efficient model adaptation methods and scalable neural architectures that can maintain strong performance while reducing computational cost.

2.2. Parameter-Efficient Fine-Tuning

Fine-tuning pretrained language models on downstream tasks is a widely used approach for adapting general-purpose models to specific applications. However, traditional fine-tuning methods require updating all model parameters, which can be computationally expensive for large models.

Parameter-efficient fine-tuning (PEFT) methods aim to address this limitation by reducing the number of trainable parameters required for adaptation. Instead of modifying the entire network, PEFT techniques introduce small task-specific modules while keeping the original model parameters frozen. One widely used PEFT approach is the adapter module introduced by Houlsby et al. [4], which inserts lightweight neural layers between transformer blocks. These adapters enable task-specific adaptation while maintaining most pretrained parameters unchanged. Another approach is **prefix tuning**, proposed by Li and Liang [5], which introduces learnable prefix tokens that modify the attention mechanism during training. Prefix tuning has been shown to achieve competitive performance with minimal parameter updates.

Among PEFT techniques, Low-Rank Adaptation (LoRA) [6] has gained significant attention due to its efficiency and simplicity. LoRA decomposes weight updates into low-rank matrices and injects them into existing linear layers of the transformer architecture. This approach dramatically reduces the number of trainable parameters while preserving the expressive capacity of the pretrained model. Subsequent extensions such as QLoRA [14] combine LoRA with quantization techniques to further reduce memory consumption and enable fine-tuning of extremely large models on limited hardware. Parameter-efficient fine-tuning methods are particularly useful for multitask learning scenarios, where a single model must be adapted to multiple tasks simultaneously. By reducing the number of trainable parameters, PEFT techniques enable efficient training and deployment of large models in resource-constrained environments.

2.3. Multitask Learning in LLM

Multitask learning aims to train a single model to perform multiple tasks simultaneously by sharing representations across tasks. The concept of multitask learning was originally introduced by Caruana [15], who showed that learning multiple related tasks can improve generalization by exploiting shared structure in the data. In the context of large language models, multitask learning enables a single model to perform a variety of tasks such as question answering, text classification, summarization, and dialogue generation. Recent work has demonstrated that multitask fine-tuning can improve performance by leveraging shared knowledge across tasks and reducing the need for separate task-specific models [16,17].

Multitask learning also offers advantages in terms of computational efficiency and model deployment. Instead of maintaining multiple specialized models for different tasks, a single multitask model can be deployed across multiple applications. However, multitask learning introduces new challenges related to resource allocation and model capacity. Different tasks may require different levels of model complexity and computational effort. Static allocation of computational resources may therefore lead to inefficient utilization of model capacity. Designing mechanisms that dynamically allocate model capacity according to task complexity or input difficulty remains an important research challenge in multitask learning systems.

2.4. Mixture-of-Experts (MoE) Architectures

MoE architectures provide an effective mechanism for scaling neural networks while maintaining computational efficiency. The original concept of MoE was introduced by

Kaplan et al. [13], who proposed a framework in which multiple expert networks are combined through a gating mechanism that selects experts based on input data.

In modern deep learning architectures, MoE layers typically consist of multiple feed-forward expert networks and a gating network that determines which experts should process each input token. Instead of activating the entire model for every input, only a small subset of experts is activated, resulting in conditional computation. Shazeer et al. [7] introduced sparsely gated MoE layers for large neural networks and demonstrated that conditional computation can dramatically increase model capacity without proportionally increasing computational cost. More recent large-scale models such as Switch Transformer [8] and GShard [9] successfully applied MoE architectures to transformer models with billions or trillions of parameters. These models demonstrated that sparse expert activation enables efficient scaling of neural networks to unprecedented sizes.

MoE architectures are particularly well suited for multitask learning scenarios because different experts can specialize in different types of inputs or tasks. This specialization allows the model to capture diverse patterns in the data while maintaining efficient computation. However, MoE architectures also introduce new challenges. One of the most well-known issues is expert imbalance, where a small number of experts receive the majority of routing assignments while others remain underutilized. Various load-balancing techniques have been proposed to address this problem [8]. Another limitation is the use of static expert routing strategies, which assign a fixed number of experts to each token regardless of input complexity.

2.5. Static Expert Routing and Its Limitations

In most MoE implementations, expert selection is performed using a top-k routing strategy, where the gating network assigns probabilities to experts and the top-k experts with the highest probabilities are selected. Although this approach is computationally efficient, it has several limitations.

First, static routing assigns the same number of experts to all tokens regardless of their difficulty or uncertainty. In natural language processing tasks, tokens may vary significantly in complexity. Some tokens can be processed easily using minimal computation, while others may require more complex reasoning or contextual understanding.

Second, static routing may lead to inefficient resource allocation. Simple tokens may receive unnecessary computational resources, while complex tokens may not receive sufficient processing capacity.

Third, static routing can contribute to expert imbalance. Because the same experts are frequently selected across tokens, certain experts may dominate routing assignments, leading to reduced diversity of learned representations.

These limitations suggest that more adaptive routing strategies could improve both computational efficiency and model performance.

2.6. Dynamic Expert Routing and Adaptive Computation

Adaptive computation has been studied extensively in neural networks as a mechanism for allocating computational resources according to input difficulty.

One of the earliest approaches is Adaptive Computation Time (ACT) proposed by Graves [18], which allows recurrent networks to dynamically determine the number of computation steps required for each input.

More recent work has explored adaptive routing mechanisms in Mixture-of-Experts models. Huang et al. [19] proposed allocating experts according to task difficulty, demonstrating that more complex tasks benefit from increased expert capacity. Similarly,

Yun et al. [20] investigated routing strategies that optimize inference efficiency by dynamically selecting experts based on input characteristics.

While these approaches demonstrate the potential benefits of adaptive routing, many existing methods rely on task-level or sequence-level difficulty estimates. These estimates may not capture fine-grained variations in token-level complexity.

While adaptive routing approaches such as ACT and dynamic MoE have been explored in prior work, these methods typically involve architectural modifications or additional training procedures that make direct comparison with the MoE-LoRA framework non-trivial. In particular, these approaches often introduce different routing objectives or computational mechanisms, which makes it difficult to isolate the effect of the proposed routing strategy under a consistent experimental setting. Unlike recent dynamic routing approaches such as Huang et al. [19], which primarily allocate experts based on task difficulty or externally estimated routing complexity, the proposed CBDR framework derives uncertainty directly from the entropy of the gating probability distribution produced within the MoE routing process itself. This design enables uncertainty estimation without requiring additional routing predictors or auxiliary difficulty estimation modules. Furthermore, the proposed CBIB objective explicitly integrates confidence-aware balancing into expert utilization, providing a unified framework for adaptive routing and expert balancing within the MoE-LoRA fine-tuning setting.

Therefore, in this work, we focus on comparing against standard MoE and MoE-LoRA baselines to provide a controlled evaluation and isolate the contribution of the proposed confidence-based dynamic routing mechanism.

2.7. Addressing Uncertainty-Based Routing by the Proposed Method

Uncertainty estimation has become an important topic in machine learning. In Bayesian deep learning, predictive uncertainty is often measured using entropy or variance of predictive distributions. Entropy-based uncertainty measures have been widely used for tasks such as active learning, ensemble prediction, and model calibration. In the context of Mixture-of-Experts architectures, the entropy of the gating distribution naturally reflects the uncertainty of expert assignment decisions. When one expert receives a dominant probability, the routing decision is confident. Conversely, when multiple experts receive similar probabilities, the gating network exhibits uncertainty. Despite this intuitive interpretation, relatively few studies have explored using gating entropy as a signal for dynamic expert allocation.

The proposed Confidence-Based Dynamic Routing (CBDR) method differs from existing approaches in several key aspects. First, CBDR estimates token-level uncertainty directly from the gating probability distribution, eliminating the need for additional difficulty estimation modules. Second, CBDR dynamically adjusts the number of activated experts per token, enabling fine-grained allocation of computational resources. Third, the proposed Confidence-Based Importance Balancing (CBIB) loss incorporates confidence information into expert balancing objectives, improving expert utilization stability.

These contributions enable adaptive compute allocation at the token level while maintaining the efficiency advantages of MoE architectures. Table 1 summarizes the differences between CBDR and existing adaptive routing approaches.

Table 1. Existing Adaptive Routing Approaches.

Method	Routing Level	Difficulty Measure	Compute Adaptation
Switch Transformer	Token	None	Fixed Top-K
Dynamic MoE	Task/Sequence	Heuristic	Dynamic
ACT	Step	Halting Probability	Dynamic
CBDR (Ours)	Token	Gating Entropy	Dynamic Expert Count

3. Proposed Method

3.1. Overview of the CBDR Framework

This section introduces the proposed Confidence-Based Dynamic Routing (CBDR) framework for adaptive expert allocation in Mixture-of-Experts (MoE) transformer architectures. The objective of the proposed method is to allocate computational resources according to token-level uncertainty during multitask fine-tuning of large language models.

Modern MoE architectures employ conditional computation by activating only a subset of expert networks for each token [7,8]. In conventional implementations, the number of experts activated for each token is fixed and determined by a predefined top-k routing rule. Although this strategy maintains computational efficiency, it fails to account for variability in token difficulty and uncertainty.

In practical natural language processing tasks, tokens vary significantly in semantic complexity. Some tokens correspond to common patterns that the model can process with high confidence, while others represent ambiguous or rare patterns requiring additional computation. Static expert routing therefore allocates identical computational resources to tokens regardless of their difficulty.

To address this limitation, the proposed CBDR framework dynamically adjusts the number of activated experts according to token-level confidence derived from the entropy of the gating distribution.

The CBDR framework consists of three main components:

1. Gating network and entropy-based confidence estimation;
2. Dynamic expert allocation mechanism;
3. Confidence-Based Importance Balancing (CBIB) loss.

The first component estimates token-level routing uncertainty using the entropy of the gating probability distribution. The second component dynamically determines the number of experts assigned to each token based on this confidence score. The third component introduces a training objective that encourages balanced expert utilization during training. This design enables the model to allocate additional computational capacity to uncertain tokens while avoiding unnecessary computation for tokens that can be processed confidently. The core idea is illustrated in Figure 2.

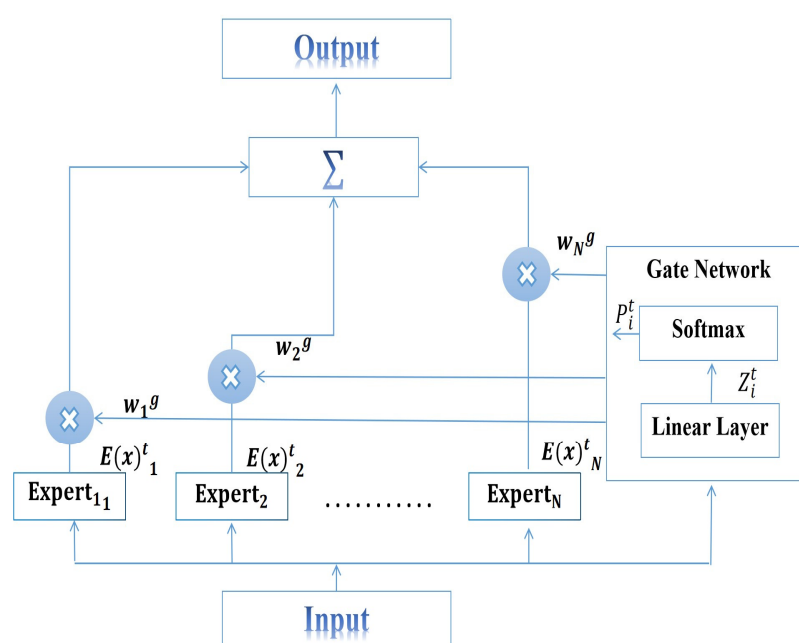


Figure 2. Gating Network.

3.2. MoE Transformer Architecture

We consider a transformer architecture consisting of alternating multi-head self-attention layers and feed-forward layers [1]. In the proposed framework, the standard feed-forward layer is replaced by a Mixture-of-Experts layer.

Let $x_t \in \mathbb{R}^d$ denote the representation of token t produced by the attention layer.

The MoE layer contains K expert networks, $E_1, E_2, \dots, E_K$, and each expert is implemented as a feed-forward network with parameters $E_k(\cdot; \theta_k)$.

A gating network determines which experts should process each token. For a given token t , the gating network produces a set of logits Z_{it} corresponding to each expert i , $G(x_t)_i = g_i \cdot x_t = Z_{it}$. These logits are converted into routing probabilities through a softmax function $P_{it} = \text{softmax}(Z_{it})$. The softmax operation produces a probability distribution over all experts, $\text{softmax}(Z_{it}) = \frac{e^{Z_{it}}}{\sum_{j=1}^K e^{Z_{jt}}}$, where K denotes the number of experts.

3.3. Entropy-Based Confidence Estimation and Dynamic Routing

In conventional MoE models, a fixed number of experts (top-K) are selected according to these probabilities. As shown by Algorithm 1, the output of the MoE layer is then computed as

$$Y = \sum_{j \in \text{top}K} P_{jt} \cdot E(x_t)_j, \quad (1)$$

where $E(x_t)_j$ denotes the output of expert j .

Algorithm 1: Confidence-Based Dynamic Routing (CBDR)

Input: token representation x_t

Output: MoE output y_t

1: Compute gating logits $z_i = g(x_t)$

2: Compute routing probabilities $p_i = \text{softmax}(z_i)$

3: Compute entropy

$$H(x_t) = -\sum_{i=1}^K p_i \log p_i$$

4: Compute confidence score

$$C(x_t) = 1 - \frac{H(x_t)}{\log K}$$

5: Determine number of experts

$$k_t = k_{\min} + (1 - C(x_t))(k_{\max} - k_{\min})$$

6: Select top k_t experts

7: Compute expert outputs

$$y_t = \sum_{i \in \text{Top}(k_t)} p_i E(x_t)_i$$

Return y_t

However, static expert routing assigns the same number of experts to all tokens regardless of their uncertainty or difficulty. To address this limitation, the proposed approach introduces a *token-level confidence score* derived from the entropy of the gating distribution. The entropy of the probability distribution is computed as $H(P_t) = -\sum_{j=1}^K P_{jt} \log P_{jt}$. This entropy measures the uncertainty of the expert assignment. Higher entropy indicates that the gating network is uncertain about which expert should process the token.

We define the token-level uncertainty score U_t based on the normalized entropy of the gating probability distribution as $U_t = H(P_t)/\log K$, where $\log$ denotes the natural logarithm. The corresponding confidence score is defined as $C_t = 1 - U_t$. Higher entropy corresponds to higher uncertainty, indicating less confident expert assignment. This formulation normalizes the confidence score to the range $[0,1]$, where higher values indicate greater certainty in expert selection. Entropy serves as a natural measure of uncertainty in probability distributions. When the gating distribution has low entropy, the model strongly prefers a specific expert, indicating high confidence in expert selection. Conversely, high entropy indicates uncertainty in expert assignment, suggesting that additional experts may be beneficial for processing the token.

Using this confidence score, the number of experts assigned to each token is dynamically adjusted, $E_t = E_{min} + C_t (E_{max} - E_{min})$, where E_{min} and E_{max} represent the minimum and maximum number of experts that can be activated for a token.

Let k_{min} denote the minimum number of experts that must be activated for each token, and k_{max} denote the maximum number of experts allowed. The number of activated experts for token t is computed as

$$k_t = k_{min} + U_t (k_{max} - k_{min}), \quad (2)$$

where U_t is the normalized uncertainty derived from the entropy of the gating distribution. Since the number of activated experts must be an integer, the computed k_t value is rounded to the nearest integer before expert selection is performed.

Equation (2) has the following properties: first, when uncertainty is low ($U_t \approx 0$), the number of experts approaches k_{min} ; second, when uncertainty is high ($U_t \approx 1$), the number of experts approaches k_{max} . This formulation aligns with the intuition that more uncertain tokens require additional expert capacity for accurate processing. This formulation reflects the intuition that more uncertain tokens require additional expert capacity for accurate processing. Since confidence is defined as $C_t = 1 - U_t$, higher confidence corresponds to fewer activated experts, establishing an inverse relationship between confidence and expert allocation.

As shown by Figure 3, the routing algorithm then selects the top k_t experts according to the routing probabilities $p_i(x_t)$.

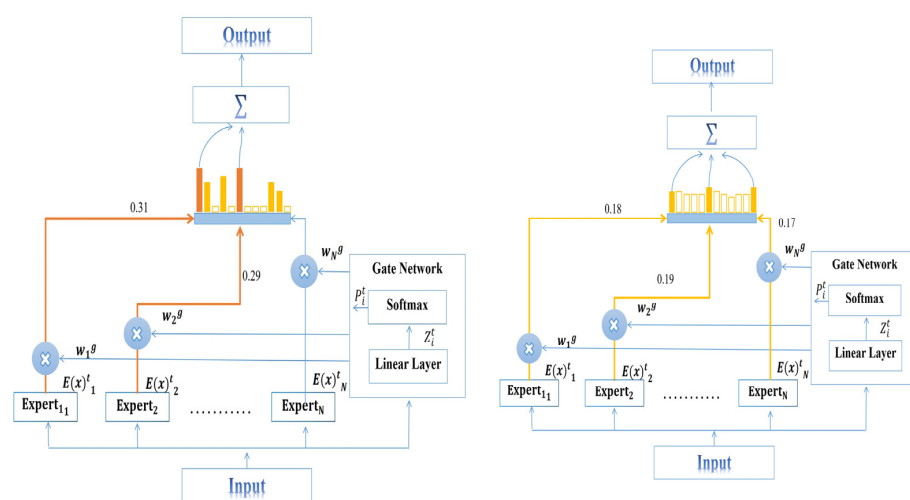


Figure 3. High confidence dynamic routing activates fewer experts (left), while low-confidence tokens activate additional experts (right).

This adaptive strategy allows the model to allocate computational resources according to token difficulty.

Under this routing strategy, high-confidence tokens are processed with fewer experts, and low-confidence tokens are assigned additional experts. This dynamic allocation allows the model to devote more computational resources to difficult tokens while avoiding unnecessary computation for easier ones.

In terms of computational overhead for dynamic routing methods, let K = total number of experts, and k_t = number of activated experts for token t .

In standard MoE routing with fixed top-k selection, the computational complexity per token is $O(kd^2)$, where d is the hidden dimension. In CBDR, the complexity becomes $O(k_t d^2)$, where k_t varies across tokens. However, the additional cost of computing entropy is $O(K)$, which is negligible compared with expert computation. Therefore, the overall complexity becomes: $O(k_t d^2 + K)$, since k_t remains small relative to K , CBDR maintains computational efficiency.

In our experiments, the number of experts K is set to six. This further demonstrates that the overhead of entropy computation is negligible compared to expert computation.

3.4. Confidence-Based Importance Balancing (CBIB) Loss

In MoE architectures, a common issue is expert imbalance, where certain experts are activated significantly more frequently than others. This imbalance can lead to inefficient expert utilization and degraded model generalization. To address this problem, we introduce a CBIB loss that encourages more uniform expert participation during training.

The importance M of each expert is computed by aggregating the product of the gating probability and token confidence across all tokens in a batch,

$$M(X) = \sum_{x \in X} G(x) \cdot C(x), \quad (3)$$

where $G(x)$ denotes the gating probability and $C(x)$ represents the token-level confidence, defined as $C(x) = 1 - U(x)$, where $U(x)$ is the normalized uncertainty.

To measure imbalance across experts, we compute the coefficient of variation (CV) of the importance values, $CV(M(X)) = \sigma(M(X)) / (\mu(M(X)) + \epsilon)$, where σ and μ represent the standard deviation and mean of expert importance values, respectively.

We use CV because it provides a normalized measure of dispersion that is independent of the scale of importance values, enabling consistent comparison across training batches. The weighting coefficient w_{CBIB} is selected based on validation performance to balance task accuracy and expert utilization.

The proposed importance balancing loss is defined as

$$L_{CBIB} = w_{CBIB} (CV(M(X)))^2, \quad (4)$$

where w_{CBIB} is a weighting coefficient. This loss penalizes imbalanced expert utilization and encourages the model to distribute workload more evenly across experts. The load balancing loss follows the auxiliary loss formulation used in Switch Transformer, which encourages uniform expert assignment. The final training objective is defined as $L_{total} = L_{load\ balancing} + L_{CBIB}$, where L_{CBIB} complements the standard load balancing objective by incorporating token-level confidence into the expert utilization process. The gradients from L_{CBIB} are applied to the gating network parameters, as the objective is to influence expert selection behavior rather than modifying expert representations directly. By incorporating confidence information into the balancing objective, the proposed method promotes stable expert utilization while preserving the benefits of adaptive expert routing.

This formulation can be interpreted as an uncertainty-aware reweighting of expert importance, where tokens with higher uncertainty contribute more strongly to the balancing objective. Unlike standard load-balancing losses that treat all tokens equally, the proposed

CBIB loss adapts the balancing process based on token difficulty, which helps stabilize expert specialization while reducing the influence of noisy or ambiguous routing decisions.

3.5. Integration with LoRA Fine-Tuning

The proposed framework is designed to work with **Low-Rank Adaptation (LoRA)** [6]. LoRA reduces the number of trainable parameters by introducing low-rank updates to weight matrices. Given a pretrained weight matrix W , LoRA replaces the update with

$$W' = W + BA, \quad (5)$$

where $A \in \mathbb{R}^{r \times d}$, $B \in \mathbb{R}^{d \times r}$ and r is the rank of the decomposition.

During training, pretrained weights W remain frozen, and only A and B are updated.

In the proposed framework, LoRA modules are inserted into (i) attention projection matrices, (ii) feed-forward layers, and (iii) expert networks.

This design allows efficient adaptation of the model to new tasks while preserving the benefits of dynamic routing.

The proposed CBDR framework offers several advantages. Computational resources are allocated according to token difficulty. It improves expert utilization as CBIB prevents expert imbalance. It provides compatibility with PEFT methods since CBDR integrates naturally with LoRA fine-tuning. In addition, it presents minimal computational overhead since entropy estimation adds negligible cost.

These properties make CBDR a practical approach for improving efficiency and scalability in MoE-based large language models.

4. Experiments

This section describes the experimental setup, datasets, and evaluation results used to assess the effectiveness of the proposed CBDR method for MoE-LoRA models. The experiments focus on evaluating whether the proposed routing mechanism improves model performance and expert utilization in multitask medical question-answering tasks.

4.1. Experimental Setup

The proposed method is implemented within a Mixture-of-Experts transformer architecture combined with Low-Rank Adaptation (LoRA) for parameter-efficient fine-tuning. All experiments are conducted using pretrained large language models augmented with MoE layers in the feed-forward components of the transformer architecture.

The primary base model used in this study is Mistral-7B-Instruct [21], a transformer-based language model with approximately seven billion parameters. This choice enables a controlled evaluation of the proposed routing mechanism within a consistent experimental setting.

The models are implemented using the PyTorch v2.4.1 deep learning framework and the HuggingFace Transformers library, which provides optimized implementations of transformer architectures and training utilities. Training is performed on a workstation equipped with an NVIDIA TITAN RTX GPU with 24 GB of memory was manufactured by NVIDIA in Santa Clara, California, United States., an Intel Xeon CPU, and 64 GB of system memory. Mixed-precision training is employed to reduce memory consumption and accelerate training.

The transformer backbone is augmented with Mixture-of-Experts layers replacing the standard feed-forward network blocks. Each MoE layer contains a set of expert networks and a gating network that determines which experts are activated for each token. In the proposed framework, the number of activated experts is dynamically determined based on token-level confidence derived from the entropy of the gating probability distribution.

To evaluate scalability, experiments are conducted using three different expert configurations consisting of 4, 8, and 16 experts, respectively. The CBDR mechanism dynamically adjusts the number of activated experts for each token within a predefined range. In the experiments, the minimum number of experts per token is set to two, while the maximum number of experts is set to six. This configuration allows the model to allocate additional computational resources to uncertain tokens while maintaining computational efficiency for high-confidence tokens.

The weighting coefficient w_{CBIB} is empirically selected based on validation performance to balance task accuracy and expert utilization.

4.2. LoRA Fine-Tuning Configuration

To enable efficient training of large models, the proposed framework incorporates Low-Rank Adaptation (LoRA) [6], which significantly reduces the number of trainable parameters during fine-tuning. Instead of updating the full weight matrices of the transformer architecture, LoRA introduces low-rank update matrices that modify the original pretrained weights while keeping the majority of parameters frozen.

In the experiments, LoRA modules are inserted into the attention projection matrices and feed-forward layers of the transformer architecture. Specifically, LoRA is applied to the query, key, value, and output projection matrices of the self-attention mechanism as well as to the linear transformations in the feed-forward network layers. The rank of the LoRA decomposition is set to 16, and the scaling factor is also set to 16, which is consistent with configurations commonly used in parameter-efficient fine-tuning studies.

During training, the parameters of the pretrained backbone model remain frozen, and only the LoRA parameters and gating network parameters are updated. This approach significantly reduces the number of trainable parameters and allows efficient fine-tuning even for large models.

The models are trained using the AdamW optimizer, which has been widely adopted for transformer-based architectures due to its stability and strong empirical performance. The learning rate is set to 2×10^{-4} , and the batch size is set to 8. Each model is trained for three epochs, which was found to be sufficient for convergence in the biomedical question-answering tasks considered in this work.

Due to the small size of the datasets, we use a limited number of training epochs to avoid overfitting. In preliminary experiments, we observed that performance saturates quickly, and additional training epochs do not yield significant improvements.

4.3. Datasets

The proposed method is evaluated on three biomedical question-answering datasets (Table 2) that represent different types of domain-specific reasoning tasks. Biomedical question answering is a particularly challenging application for large language models because it requires understanding specialized terminology and reasoning over scientific knowledge.

Table 2. Dataset Statistics [22–26].

Dataset	# Total	# Train	# Test	Task
BioASQ [22–24]	810	670	140	Yes/No
PubMedQA [25]	1000	900	100	Yes/No/Maybe
RealMedQA [26]	1159	1000	159	Q-A

Note: # denotes the number of samples.

These datasets represent different biomedical QA scenarios, including binary classification and open-ended question answering. Biomedical QA tasks often involve limited training data and domain-specific reasoning, making them suitable for evaluating the effec-

tiveness of dynamic expert allocation. The first dataset used in the experiments is **BioASQ**, a biomedical question-answering benchmark derived from PubMed articles [22–24]. BioASQ questions are created by biomedical experts and cover a wide range of topics including gene-disease relationships, biological processes, and drug interactions. In this study, we focus on the yes/no question subset of BioASQ, which requires determining whether a biomedical statement is correct. The dataset contains 670 training samples and 140 test samples. The second dataset is **PubMedQA** [25], which consists of biomedical questions derived from PubMed abstracts. Each question is accompanied by supporting text from biomedical literature, and the task requires the model to determine whether the answer is “yes,” “no,” or “maybe.” PubMedQA emphasizes reasoning over scientific evidence rather than simple knowledge retrieval. The dataset used in the experiments contains 900 training samples and 100 test samples. The third dataset is **RealMedQA** [26], which consists of real-world clinical questions collected from healthcare environments. These questions often involve diagnostic reasoning, treatment decisions, and interpretation of medical evidence. Compared with BioASQ and PubMedQA, RealMedQA focuses more strongly on clinical reasoning tasks. The dataset contains 1000 training samples and 159 test samples.

Together, these datasets provide complementary evaluation scenarios covering biomedical knowledge retrieval, scientific reasoning, and clinical decision-making.

4.4. Baseline Models

To evaluate the effectiveness of the proposed CBDR mechanism, several baseline models are considered. The first baseline is the pretrained language model without task-specific fine-tuning. This model serves as a reference point to evaluate the benefits of parameter-efficient fine-tuning and expert specialization.

The second baseline is a **LoRA fine-tuned model without expert routing**. This model evaluates the benefits of parameter-efficient fine-tuning alone and provides a comparison for the proposed MoE-based architecture.

The third baseline is a **MoE-LoRA model with static expert routing**, which represents the standard approach used in most existing MoE architectures. In this configuration, a fixed top-k routing strategy is used to select experts for each token. Specifically, two experts are selected for each token according to the highest routing probabilities produced by the gating network. These baselines allow the experiments to isolate the contributions of dynamic routing and expert balancing introduced in the proposed method.

4.5. Main Experimental Results

Table 3 summarizes the performance of the evaluated models across the three biomedical question-answering datasets. The results demonstrate that the proposed CBDR mechanism consistently improves performance compared with both the LoRA-only model and the static MoE-LoRA baseline.

The results show that the proposed CBDR mechanism consistently improves performance across all datasets. On BioASQ, CBDR achieves the highest F1 score of 0.88 and MRR (Mean Reciprocal Rank) of 0.83, outperforming both Single LoRA and the static MoE-LoRA baseline. On PubMedQA, the combination of CBDR with the importance balancing loss achieves the best results, reaching an F1 score of 0.53 and accuracy of 0.78. On RealMedQA, CBDR improves cosine similarity from 0.71 to 0.74, indicating improved semantic alignment between predicted and reference answers. These results demonstrate that dynamic expert routing based on token confidence improves both model performance and expert utilization.

Table 3. Main Results of Confidence-Based Dynamic Routing (CBDR) across datasets. All results are obtained using the Mistral-7B-Instruct base model unless otherwise noted. Cosine Similarity is used as the evaluation metric for RealMedQA.

Dataset	BioASQ [22–24]		PubMedQA [25]		RealMedQA [26]
Metric	F1	MRR	F1	Accuracy	Cosine Similarity
Pretrained	0.70	0.66	0.15	0.17	0.70
Single LoRA	0.84	0.71	0.36	0.45	0.84
Baseline (MoE-LoRA)	0.85	0.79	0.48	0.73	0.85
MoE-LoRA + CBDR	0.88	0.83	0.51	0.76	0.88
MoE-LoRA + CBDR + $L_{importance}$	0.87	0.82	0.53	0.78	0.87

Note: Bold values indicate the best results.

Although CBDR dynamically increases the number of experts for uncertain tokens, the overall computation remains efficient because a large portion of tokens are processed with fewer experts. In practice, the cost of computing entropy from the gating distribution is negligible compared with the computational cost of activating additional experts. As a result, CBDR introduces minimal overhead while enabling more efficient allocation of expert computation.

4.6. Ablation Study

To further analyze the contribution of different components, several CBDR configurations were evaluated. The results reported in Tables 4 and 5 correspond to ablation configurations under different experimental settings, including variations in LoRA application and routing strategies. Therefore, absolute values may differ from the main results reported in Table 3, which are obtained under the primary experimental configuration.

The results in Table 4 indicate that CBDR consistently improves performance compared to static routing, while importance balancing enhances expert utilization stability. Among the evaluated configurations, combining CBDR with a fixed top-k expert subset achieves the best overall performance, suggesting a balanced trade-off between routing flexibility and stable expert specialization. While CBDR dynamically allocates experts according to token uncertainty, maintaining a small fixed expert subset helps stabilize the routing process and encourages consistent expert specialization during training.

Table 4. Performance of Different CBDR Configurations with MoE-LoRA.

Dataset	BioASQ		PubMedQA		RealMedQA
Metric	F1	MRR	F1	Accuracy	Cosine Similarity
MoE-LoRA + CBDR	0.85	0.78	0.42	0.61	0.71
MoE-LoRA + CBDR + $L_{importance}$	0.87	0.81	0.40	0.61	0.75
MoE-LoRA + CBDR + Top-K	0.89	0.84	0.42	0.60	0.78
MoE-LoRA + CBDR + Adaptive Threshold	0.88	0.83	0.41	0.60	0.57

Note: Bold values indicate the best results.

Table 5. Ablation study evaluating the contribution of CBDR and CBIB expert balancing across biomedical QA datasets. CBDR + Fixed Top-K ensures inclusion of a fixed subset of top-k experts, while CBDR + Adaptive Threshold uses an uncertainty threshold to control expert allocation.

Model Configuration	CBDR	CBIB Loss	Routing Strategy	BioASQ F1	PubMedQA Accuracy	RealMedQA Similarity
Static MoE-LoRA (Baseline)	X	X	Fixed Top-2	0.85	0.73	0.71
CBDR Only	✓	X	Dynamic Experts	0.87	0.75	0.73
CBDR + CBIB (Proposed)	✓	✓	Dynamic Experts	0.88	0.76	0.74
CBDR + Fixed Top-K	✓	X	Top-2 + Dynamic Scaling	0.87	0.74	0.73
CBDR + Adaptive Threshold	✓	X	Entropy Threshold	0.86	0.74	0.72

Note: Bold values indicate the best results. The symbol ✓ indicates that the corresponding component is used, whereas X indicates that the corresponding component is not used.

Table 5 presents the ablation study evaluating the individual contributions of the proposed CBDR mechanism and the CBIB importance balancing loss. To better understand the contributions of different components of the proposed framework, we conduct an ablation study that evaluates several model configurations. The first configuration uses CBDR without the CBIB loss. This experiment evaluates the impact of dynamic routing alone. The second configuration includes both CBDR and CBIB, representing the full proposed model. Additional configurations explore variations in routing strategies, such as combining CBDR with fixed top-k selection and adaptive threshold routing. The ablation results indicate that CBDR alone provides a significant performance improvement compared with static routing. Adding the CBIB loss further improves model stability and expert utilization by encouraging balanced expert participation during training.

Table 6 analyzes expert utilization patterns during inference to evaluate whether the proposed routing mechanism improves expert diversity. In conventional static MoE routing, a fixed number of experts are selected for each token, often leading to routing imbalance where a small subset of experts receives the majority of routing assignments. This phenomenon is reflected in the baseline results, where the maximum expert load reaches nearly 39% while some experts receive fewer than 5% of assignments. Introducing CBDR reduces this imbalance by dynamically allocating experts according to token uncertainty. As shown in the table, the standard deviation of expert utilization decreases from 0.41 to 0.28, indicating a more balanced distribution of routing assignments across experts. When the CBIB loss is added, expert utilization becomes significantly more balanced. The maximum expert load decreases to approximately 25%, while the minimum expert load increases to over 12%. These results confirm that the proposed CBIB loss effectively mitigates routing collapse and promotes balanced expert participation during training. Balanced expert utilization is important because it allows the model to fully exploit the representational capacity of the expert networks. By distributing routing assignments more evenly, the proposed method increases the diversity of learned representations and improves model generalization.

Table 6. Expert utilization statistics comparing static routing and the proposed CBDR mechanism.

Model	Experts	Avg Experts per Token	Utilization Std Dev ↓	Max Expert Load (%) ↓	Min Expert Load (%) ↑
Static MoE-LoRA	8	2.0	0.41	38.7	4.3
CBDR (no CBIB)	8	2.3	0.28	31.2	8.5
CBDR + CBIB	8	2.4	0.17	24.6	12.3

Note: Bold values indicate the best performance among all methods. The symbols ↑ and ↓ indicate that higher and lower values are better, respectively.

Table 7 evaluates the sensitivity of the proposed routing mechanism to the minimum and maximum expert allocation parameters. These parameters control the range of experts that may be activated for each token depending on the confidence score derived from the entropy of the gating distribution.

Table 7. Sensitivity analysis of routing hyperparameters controlling the minimum and maximum number of activated experts.

k_{min}	k_{max}	Avg Experts per Token	BioASQ F1	PubMedQA Accuracy
1	4	1.8	0.86	0.74
2	4	2.1	0.87	0.75
2	6	2.4	0.88	0.76
3	6	2.8	0.88	0.76
3	8	3.1	0.88	0.76

Note: Bold values indicate the best performance among all methods.

The results show that very small values of k_{min} slightly degrade model performance because some tokens receive insufficient expert capacity. Increasing k_{min} to two improves performance across all datasets by ensuring that each token receives at least a minimal level of expert processing.

The best overall performance is achieved when $k_{min} = 2$ and $k_{max} = 6$. In this configuration, the model maintains efficient computation for high-confidence tokens while allowing uncertain tokens to activate additional experts. Further increasing k_{max} produces only marginal improvements in performance while increasing computational cost. This result suggests that dynamic routing provides the greatest benefit when the expert allocation range is moderate. Overall, the sensitivity analysis demonstrates that the proposed CBDR mechanism is robust to reasonable variations in routing hyperparameters and does not require careful tuning to achieve strong performance.

To directly test the core hypothesis underlying the proposed CBDR framework, we conducted an entropy–performance analysis in which tokens were grouped according to the entropy of the gating probability distribution $H(P_t)$, where $H(P_t)$ represents the entropy of the expert selection probabilities produced by the gating network. This grouping enables us to analyze how model performance varies across different levels of token uncertainty. The purpose of this experiment was to determine whether the performance benefits of dynamic routing are concentrated on tokens with higher uncertainty, as predicted by the proposed method. As shown in Table 8, the performance gain of CBDR over static MoE routing is strongly correlated with gating entropy. Here, the performance gain refers to the improvement of CBDR over static routing within each entropy bin. In the low-entropy regime, where the gating network exhibits high confidence in expert assignment, the performance of CBDR is nearly identical to that of static routing. This result is expected because such tokens are relatively easy to process and do not require additional expert

capacity. In this regime, CBDR typically activates the minimum number of experts, thereby preserving computational efficiency.

Table 8. Entropy-based performance analysis—tokens are partitioned into entropy bins according to the gating distribution. Performance gain is measured as the improvement of CBDR over static MoE routing within each entropy range.

Normalized Entropy Bin	Confidence Level	Avg Experts Activated (CBDR)	Static MoE Accuracy/F1	CBDR Accuracy/F1	Relative Gain
Low Entropy ([0.0, 0.3))	High confidence	2.0	0.89	0.89	+0.0%
Medium Entropy ([0.3, 0.6))	Moderate confidence	2.6	0.84	0.87	+3.6%
High Entropy ([0.6, 1.0])	Low confidence	3.8	0.76	0.83	+9.2%

In contrast, the gains become substantially larger for tokens with moderate and high gating entropy. For medium-entropy tokens, CBDR improves performance by approximately 3.6%, while for high-entropy tokens the performance gain increases to more than 9%. These results strongly support the central hypothesis of the paper: uncertain tokens benefit disproportionately from additional expert allocation. By activating more experts for tokens where the gating network is uncertain, CBDR enables richer expert aggregation and improved representation quality.

These results suggest that a substantial portion of the overall performance improvement achieved by CBDR originates from the effective handling of a relatively small subset of high-uncertainty tokens. This observation supports the efficiency motivation of the proposed method, where additional expert capacity is selectively allocated only to tokens that require more complex processing.

The corresponding trend is illustrated in Figure 4, where performance gain is plotted as a function of gating entropy. The approximately monotonic relationship between entropy and performance improvement provides direct empirical validation of the proposed uncertainty-aware routing strategy.

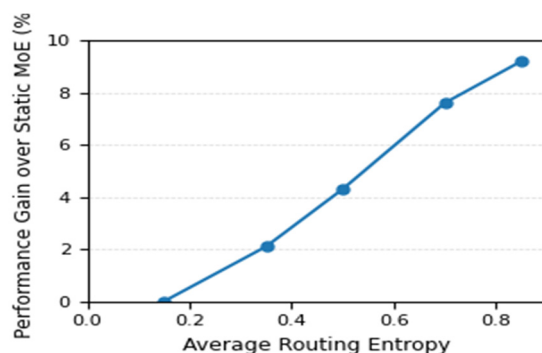


Figure 4. Relationship between gating (or called routing) entropy and CBDR performance gain. Higher entropy tokens receive additional expert allocation.

Figure 4 illustrates the relationship between gating entropy and the relative performance gain of CBDR compared with static MoE routing. The horizontal axis represents the average entropy of the gating distribution, and the vertical axis represents the relative performance improvement of CBDR. The monotonic increase indicates that the benefit of dynamic expert allocation grows with routing uncertainty, supporting the central hypothesis that uncertain tokens benefit most from additional expert capacity. As gating entropy increases, the performance benefit of CBDR increases monotonically. For low-entropy

tokens, where the gating network is already confident in expert assignment, the performance gain is negligible. In contrast, high-entropy tokens exhibit substantially larger gains, indicating that additional expert allocation is most beneficial when routing uncertainty is high. This trend provides direct empirical support for the core hypothesis of the proposed uncertainty-aware routing mechanism.

This is particularly important because it demonstrates that CBDR is not merely a heuristic modification of static routing, but rather a principled mechanism whose benefits are concentrated precisely where additional computation is most needed.

From a broader perspective, this result strengthens the theoretical motivation for using entropy as a routing signal. In probabilistic models, entropy is a standard measure of uncertainty, and the present results show that gating entropy serves as a practical indicator of token-level difficulty in MoE-based language models. The experiment therefore provides both empirical and conceptual support for the use of entropy-based confidence in dynamic expert allocation.

Lastly, in our ablation study, we measured the average number of activated experts per token and estimated the corresponding floating-point operations (FLOPs) to evaluate the computational efficiency of CBDR. Table 9 shows the computational efficiency of CBDR. The reported FLOP values are approximate estimates derived from the average number of activated experts per token and the corresponding MoE layer configuration, rather than direct hardware-level measurements. The additional overhead introduced by the CBIB objective mainly arises from the computation of the coefficient of variation across expert importance values during training.

Table 9. Computational Efficiency of CBDR.

Model	Avg Experts/Token	Relative FLOPs	Performance Gain
Static MoE	2.0	1.00	baseline
CBDR	2.3	1.15	+3–5%
CBDR + CBIB	2.4	1.20	+4–6%

Although CBDR occasionally activates more experts for uncertain tokens, high-confidence tokens frequently require only two experts, reducing overall computing requirements relative to naive dynamic routing strategies. These results indicate that CBDR improves model performance while maintaining computational efficiency.

In practice, the gating entropy distribution indicates that a large proportion of tokens exhibit relatively high confidence. These tokens can therefore be processed with fewer experts, allowing the model to allocate additional computational resources to more uncertain tokens.

While Table 9 reports relative FLOPs to quantify computational cost, we note that wall-clock training and inference time may vary depending on hardware implementation and system-level optimizations. In particular, dynamic expert allocation can introduce irregular computation patterns that affect GPU/TPU utilization.

However, the primary goal of this analysis is to provide a relative comparison of computational cost across routing strategies. Since entropy is computed directly from gating probabilities, the additional overhead remains negligible compared to expert computation.

4.7. Experiment Discussions

The experimental results demonstrate that the proposed CBDR mechanism improves the efficiency and performance of MoE-based language models. By allocating experts dynamically according to token-level confidence, the model can dedicate more computational resources to difficult inputs while avoiding unnecessary computation for simpler

tokens. Although the CBIB loss slightly reduces peak task performance in some cases, it improves expert utilization balance and may enhance training stability in larger-scale models. Overall, the results suggest that uncertainty-aware expert routing provides an effective approach for improving both efficiency and performance in MoE-based LLMs.

To better understand the behavior of the proposed routing mechanism, we analyze the expert utilization patterns observed during the experiments. In MoE architectures, expert imbalance is a common issue, where a small subset of experts receives the majority of routing assignments while others remain underutilized. Such imbalance can reduce computational efficiency and limit the diversity of learned representations.

In the baseline MoE-LoRA model with static top-k routing, the number of activated experts per token is fixed, which often leads to repeated selection of the same experts. In contrast, the proposed CBDR dynamically adjusts the number of activated experts based on token-level confidence. High-confidence tokens are processed with fewer experts, while uncertain tokens activate additional experts. This routing behavior often results in expert concentration, where a small subset of experts receives a large proportion of routing assignments. Such concentration limits the diversity of expert participation and may reduce the overall representational capacity of the MoE architecture.

From the experimental observations, CBDR distributes routing assignments across a broader range of experts compared to static routing. This adaptive routing increases routing diversity and helps mitigate routing collapse, where the gating network repeatedly selects the same experts. Furthermore, the CBIB loss encourages a more uniform distribution of routing probabilities across experts by penalizing high variance in expert importance.

Overall, these results indicate that CBDR improves expert utilization by increasing routing diversity while maintaining balanced expert participation. This behavior supports more efficient and stable MoE training and demonstrates that confidence-based routing provides an effective mechanism for uncertainty-aware compute allocation in MoE-based language models. One possible explanation for this improvement is that dynamic expert allocation allows less-confident tokens to explore a broader set of experts, while the CBIB loss reinforces balanced expert participation across the model. This combination promotes more robust expert specialization and reduces the risk of routing collapse, where a small subset of experts dominates the routing decisions. Furthermore, the additional computation required to estimate the gating entropy is minimal compared with the cost of activating additional experts. Since entropy is computed directly from the gating probabilities that are already available during routing, the overhead introduced by CBDR is negligible relative to the expert computation.

5. Limitations and Future Work

While the proposed CBDR framework demonstrates consistent improvements in performance and expert utilization, several limitations should be noted. We acknowledge that the evaluation datasets are relatively small, which may limit statistical significance. However, consistent performance improvements are observed across all datasets and metrics, suggesting that the proposed method provides a stable performance gain. Future work will include evaluation on larger datasets with statistical significance analysis. First, the method relies on the choice of k_{min} and k_{max} to control the range of expert allocation. Although our experiments show that the model is robust across a reasonable range of these parameters, extreme settings may lead to either under-utilization or excessive activation of experts. Future work could explore adaptive strategies for automatically determining these bounds. Second, the current evaluation focuses on question-answering tasks. While the proposed routing mechanism is general, its effectiveness for other task types, such as open-ended text generation or long-form reasoning, requires further investigation. Third,

although the computational overhead introduced by CBDR is minimal, as entropy is computed directly from the gating probabilities, the cost is incurred at every token and every MoE layer. For very large-scale models with many MoE layers, this overhead may accumulate. Future work could investigate more efficient approximations or selective entropy computation strategies. These limitations suggest several directions for future research to further improve the scalability and general applicability of uncertainty-aware routing in MoE-based language models.

6. Conclusions

This paper introduced a Confidence-Based Dynamic Routing (CBDR) mechanism for improving expert allocation in Mixture-of-Experts language models. The proposed method was implemented within a MoE-LoRA framework, which combines Mixture-of-Experts architectures with Low-Rank Adaptation (LoRA) for parameter-efficient multitask fine-tuning. By leveraging token-level confidence scores derived from the entropy of the gating distribution, the proposed routing mechanism dynamically adjusts the number of activated experts for each token. Experimental results on three biomedical question-answering datasets—BioASQ, PubMedQA, and RealMedQA—demonstrate that the proposed approach consistently improves model performance compared with conventional static routing strategies. The CBDR mechanism enables more efficient expert allocation by assigning additional computational resources to uncertain tokens while avoiding unnecessary computation for high-confidence inputs. As a result, the model achieves improved performance across multiple evaluation metrics, including F1 score, MRR, accuracy, and cosine similarity. In addition, the proposed Confidence-Based Importance Balancing (CBIB) loss helps mitigate expert load imbalance by encouraging more uniform expert participation during training. Although this balancing objective may introduce minor trade-offs in peak task performance, it improves expert utilization stability and prevents routing collapse, which is particularly important for scalable MoE architectures.

Overall, the findings indicate that uncertainty-aware routing provides an effective approach for improving both performance and computational efficiency in MoE-based large language models. Future work may extend the proposed method to larger multitask benchmarks and investigate its effectiveness in large-scale MoE architectures and other application domains.

These findings suggest that uncertainty-aware routing provides a practical direction for improving the efficiency and adaptability of future large-scale MoE-based language models.

Author Contributions: Conceptualization, M.E. and H.K.; methodology, M.E., M.J. and H.K.; software, M.E., M.J. and H.K.; validation, M.E., M.J. and H.K.; formal analysis, M.E., M.J. and H.K.; investigation, M.E., M.J. and H.K.; resources, M.E. and M.J.; data curation, M.E. and M.J.; writing—original draft preparation, M.E.; writing—review and editing, M.J. and H.K.; visualization, M.E., M.J. and H.K.; supervision, H.K.; project administration, H.K.; funding acquisition, not applicable. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are publicly available and can be accessed from their respective sources: BioASQ at <http://bioasq.org> (accessed on 9 September 2024), PubMedQA at <https://pubmedqa.github.io> (accessed on 10 March 2026), and RealMedQA at <https://huggingface.co/datasets/k2141255/RealMedQA> (accessed on 10 March 2026). No additional license or access permission is required for the use of these publicly available datasets.

Acknowledgments: During the preparation of this work, the authors used ChatGPT (GPT-5.5) for grammar correction. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CBDR	Confidence-Based Dynamic Routing
CBIB	Confidence-Based Importance Balancing
LLMs	Large language models
PEFT	Parameter-Efficient Fine-Tuning
MoE	Mixture-of-Experts
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pre-trained Transformer
LoRA	Low-Rank Adaptation
ACT	Adaptive Computation Time

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017.
2. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. In Proceedings of the NeurIPS, Vancouver, BC, Canada, 6–12 December 2020.
3. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774. [[CrossRef](#)]
4. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-Efficient Transfer Learning for NLP. In Proceedings of the ICML, Long Beach, CA, USA, 10–15 June 2019.
5. Li, X.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the ACL, Bangkok, Thailand, 1–6 August 2021.
6. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of the ICLR, Virtual, 25–29 April 2022.
7. Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; Dean, J. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In Proceedings of the ICLR, Toulon, France, 24–26 April 2017.
8. Fedus, W.; Zoph, B.; Shazeer, N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *J. Mach. Learn. Res.* **2022**, *23*, 1–39.
9. Lepikhin, D.; Lee, H.; Xu, Y.; Chen, D.; Firat, O.; Huang, Y.; Krikun, M.; Shazeer, N.; Chen, Z. GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. In Proceedings of the ICLR, Vienna, Austria, 3–7 May 2021.
10. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019.
11. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. *Improving Language Understanding by Generative Pre-Training*; OpenAI Technical Report; OpenAI: San Francisco, CA, USA, 2018.
12. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. *Language Models Are Unsupervised Multitask Learners*; OpenAI Technical Report; OpenAI: San Francisco, CA, USA, 2019.
13. Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling Laws for Neural Language Models. *arXiv* **2020**, arXiv:2001.08361. [[CrossRef](#)]
14. Dettmers, T.; Pagnoni, A.; Holtzman, A.; Zettlemoyer, L. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv* **2023**, arXiv:2305.14314. [[CrossRef](#)]
15. Caruana, R. Multitask Learning. *Mach. Learn.* **1997**, *28*, 41–75. [[CrossRef](#)]
16. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
17. Zhang, Y.; Yang, Q. A Survey on Multi-Task Learning. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 5586–5609. [[CrossRef](#)]
18. Graves, A. Adaptive Computation Time for Recurrent Neural Networks. *arXiv* **2016**, arXiv:1603.08983. [[CrossRef](#)]
19. Huang, Q.; An, Z.; Zhuang, N.; Tao, M.; Zhang, C.; Jin, Y.; Xu, K.; Xu, L.; Chen, L.; Huang, S.; et al. Harder Tasks Need More Experts: Dynamic Routing in Mixture-of-Experts Models. *arXiv* **2024**, arXiv:2403.07652. [[CrossRef](#)]

20. Yun, L.; Zhuang, Y.; Fu, Y.; Xing, E.P.; Zhang, H. Toward Inference-Optimal Mixture-of-Experts Large Language Models. *arXiv* **2024**, arXiv:2404.02852. [[CrossRef](#)]
21. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. *arXiv* **2023**, arXiv:2310.06825. [[CrossRef](#)]
22. Nentidis, A.; Katsimpras, G.; Krithara, A.; Krallinger, M.; Rodríguez-Ortega, M.; Loukachevitch, N.; Sakhovskiy, A.; Tutubalina, E.; Dimitriadis, D.; Tsoumakas, G.; et al. Overview of BioASQ 2025: The Thirteenth BioASQ Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering. In Proceedings of the CLEF Working Notes, Lucca, Italy, 6–10 April 2025.
23. Tsatsaronis, G.; Balikas, G.; Malakasiotis, P.; Partalas, I.; Zschunke, M.; Alvers, M.R.; Weissenborn, D.; Krithara, A.; Petridis, S.; Polychronopoulos, D.; et al. An Overview of the BioASQ Large-Scale Biomedical Semantic Indexing and Question Answering Competition. *BMC Bioinform.* **2015**, *16*, 138. [[CrossRef](#)] [[PubMed](#)]
24. BioASQ Dataset. Available online: <http://bioasq.org> (accessed on 9 September 2024).
25. Jin, Q.; Dhingra, B.; Liu, Z.; Cohen, W.; Lu, X. PubMedQA: A Dataset for Biomedical Research Question Answering. In Proceedings of the EMNLP, Hong Kong, China, 3–7 November 2019.
26. Kell, G.; Roberts, A.; Umansky, S.; Khare, Y.; Ahmed, N.; Patel, N.; Simela, C.; Coumbe, J.; Rozario, J.; Griffiths, R.R.; et al. RealMedQA: A Pilot Biomedical Question Answering Dataset Containing Realistic Clinical Questions. *arXiv* **2024**, arXiv:2408.08624. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.