FuseGate:基于置信度门控的词元级并联协作机制

田淑娟[1,2],罗宇航[1,2],项淑桓[1,2],李艳春[1,2*],代星霞[1,2],申冬苏[1,2]
(1. 湘潭大学计算机学院,湖南湘潭 411105;2. 湖南省智慧网络国际科技创新合作基地,湖南湘潭 411105)

摘 要: 在模型互联网场景中,部署于不同边缘节点的异构模型可通过协作推理实现能力互补,从而提升系统在复杂任务上的推理能力。词元级并联协作能够在自回归解码过程中融合多个模型的预测信息,是实现细粒度模型协同推理的重要方式。现有词元级并联协作方法多采用逐词元同步的全程协作策略,即在每个解码步均调用多个模型参与候选词元生成与结果聚合。该策略虽有助于提升词元生成质量,但多模型协作并非在所有解码位置均能产生有效收益。在基模型具有较高预测置信度的词元位置继续执行多模型协作,往往会引入额外的计算开销与同步等待,从而增加单词元生成平均时延。针对上述问题,本文提出一种基于置信度门控的词元级并联协作机制FuseGate,可作为轻量级模块集成至现有词元级并联协作框架中。FuseGate无需额外预训练或模型参数更新,也不改变原有的自回归解码范式。该机制以基模型当前解码步Top-1与Top-2候选词元的logits差值表征预测置信度,并利用滑动窗口记录近期置信度变化,根据历史置信度分布动态调整协作判定阈值。在此基础上,FuseGate判断当前解码位置是否需要触发多模型协作,从而将协作计算优先分配至低置信度位置,减少全程协作产生的冗余计算与同步开销。本文构建多组异构模型组合,并在计算推理、阅读理解、多步数学推理和跨领域知识理解等多类任务上开展实验。实验结果表明,FuseGate能够适配不同类型的词元级并联协作框架,在整体推理准确率与全程协作基本相当的情况下,将单词元生成平均时延降低约46%。进一步分析表明,FuseGate对具有不同能力差距的模型组合均表现出较好的适用性,能够在减少无效协作开销的同时,保留有效协作带来的推理性能增益。研究结果表明,基于置信度的选择性协作能够在推理准确率与生成效率之间取得更优折中,为模型互联网场景下异构模型的高效词元级协同推理提供一种轻量化且具有良好框架兼容性的协作机制。

关键词: 大语言模型;模型互联网;词元级并联协作;置信度门控;轻量级集成;选择性协作

基金项目: 国家自然科学基金(No.U24A20247,No.62172349,No.62372396,No.6250073741);湖南省重大标志性创新示范工程项目(No.2022XK2301)

中图分类号: TN92 **文献标识码:** A **文章编号:** 0372-2112(XXXX)XX-0001-15

电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20260419

FuseGate: A Confidence-Gated Mechanism for Token-Level Parallel Collaboration

TIAN Shujuan[1,2], LUO Yuhang[1,2], XIANG Shuhuan[1,2], LI Yanchun[1,2*], DAI Xingxia[1,2], SHEN Dongsu[1,2]
(1. *School of Computer Science*, *Xiangtan University*, *Xiangtan*, *Hunan* 411105, *China*;
2. *Hunan International Scientific and Technological Cooperation Base of Intelligent network*, *Xiangtan University*, *Xiangtan*, *Hunan* 411105, *China*)

Abstract: In the AI-Model Network, heterogeneous models deployed at different edge nodes can complement one another through collaborative inference, thereby enhancing the system's reasoning capability on complex tasks. Token-level parallel collaboration enables fine-grained model cooperation by combining predictions from multiple models during autoregressive decoding. Existing methods commonly adopt full token-wise synchronous collaboration, in which multiple models are invoked at every decoding step for candidate token generation and result aggregation. Although this strategy can improve token generation quality, multi-model collaboration does not yield meaningful benefits at every decoding position. When the base model already has high confidence in its prediction, continuing to involve additional models introduces unnecessary computation and synchronization delays, increasing the average per-token generation latency. To address this problem, we propose FuseGate, a confidence-gated mechanism for token-level parallel collaboration that can be integrated into existing collaboration frameworks as a lightweight module. FuseGate requires neither additional pretraining nor model parameter updates and does not alter the original autoregressive decoding paradigm. At each decoding step, it measures the prediction confidence of the base model using the logit difference between its Top-1 and Top-2 candidate tokens, records recent confidence variations through a sliding window, and dynamically adjusts the collaboration threshold according to the

收稿日期:2026-03-31;录用日期:2026-07-02;责任编辑:张通
*通信作者:李艳春

historical confidence distribution. Based on this adaptive threshold, FuseGate determines whether multi-model collaboration should be triggered at the current decoding position, thereby concentrating collaborative computation on low-confidence positions and reducing the redundant computation and synchronization overhead caused by full collaboration. We evaluate FuseGate using multiple heterogeneous model combinations on tasks including computational reasoning, reading comprehension, multi-step mathematical reasoning, and cross-domain knowledge understanding. Experimental results show that FuseGate is compatible with different token-level parallel collaboration frameworks and reduces the average per-token generation latency by approximately 46% while maintaining overall inference accuracy comparable to that of full collaboration. Further analysis shows that FuseGate remains effective across model combinations with different capability gaps, reducing ineffective collaboration while preserving the reasoning gains brought by beneficial model cooperation. These results demonstrate that confidence-based selective collaboration achieves a more favorable trade-off between inference accuracy and generation efficiency, providing a lightweight and framework-compatible mechanism for efficient token-level collaborative inference among heterogeneous models in the AI-Model Network.

Keywords: large language models; AI-Model Network; token-level parallel collaboration; confidence gating; lightweight integration; selective collaboration

Foundation Item(s): National Natural Science Foundation of China (No.U24A20247, No.62172349, No.62372396, No.6250073741); Major Landmark Innovation Demonstration Project of Hunan Province (No.2022XK2301)

0 引言

大语言模型在自然语言生成与复杂推理任务中的表现逐年提升。大语言模型的计算开销与部署成本高,使单模型范式在边缘环境和资源受限场景下难以发挥作用。为突破上述限制,文献[1]提出模型互联网(AI-Model Network)的概念。模型互联网将部署于不同网络节点的模型组织为协同体,使得协同体中各个节点模型在成本可控的前提下,获得接近模型网络中强模型的推理性能。在协同体中,任务通常先由基模型接收并完成初步处理,随后一个或多个协作模型再参与补充推理。基模型与协作模型依托协作机制进行配合,从而提升协同体在复杂任务场景下的推理能力。因此,如何在模型互联网场景下设计更加高效的协作机制,已成为当前研究中的重要问题[2-5]。

在模型互联网背景下,多模型并联协作逐渐发展为一种重要的协作范式[6]。多模型并联协作让多个模型在相同的输入文本下同时参与答案生成,整合各个模型生成的输出结果得到最终答案。现有并联协作方法按照协作粒度的不同可分为全回复级协作和词元级协作:全回复级协作是令多个模型独立生成完整回答,经过投票或重排序后给出最终的输出结果[7-8]。全回复级协作实现简单,但需要对多个模型的候选结果进行额外评审,导致该方法推理成本较高;词元级协作能够在生成过程中对多个模型的预测结果逐词元融合。词元级方法能对输出结果进行细粒度控制,逐渐成为模型互联网场景下的重要研究方向。

目前来看,词元级协作已成为多模型协作推理的一种重要实现方式。单个模型在获得当前上下文后,会对下一位置可能生成的候选词元进行评分,这些评分共同组成相应的logits向量。在这些候选词元中,得分最高的候选记为Top-1,次高者记为Top-2,模型会根据评分结果来进行下一个词元的选择[9]。词元级协作的关键在于,在相同上下文下,把多个模型对下一词元的logits向量进行融合。现有词元级协作方法可分为两类:一类通过对齐不同模型在单个词元位置上的词表及其logits向量进行过滤、加权等操作,构造协同体整体的候选词元与评分结果;另一类方法则是在并联解码的每一步中,对各个模型提出的候选词元进行逐一评估,再根据评估结果确定当前解码步最终要生成的内容。

现有词元级并联协作采用全程同步集成策略。在每一个解码步,协同体都调用多个模型参与结果生成。多模型参与结果生成能有效提升生成质量,但也带来了两个问题:首先,并非基模型的所有解码位置都需要多模型协作;其次,同步等待多个模型完成当前解码步会引入额外的时间开销,这会导致单词元生成平均时延的增加。实际上,基模型在不同词元位置的生成质量并不相同。在部分词元位置,基模型已经给出可靠的预测,如果继续在这一词元位置执行并联协作,就会造成资源浪费。因此,如何根据基模型在当前词元位置的生成质量选择性地触发多模型协作,已经成为提升词元级并联协作效率的重要问题。

选择合适的词元级置信度特征是实现选择性协作的前提。对基模型而言,置信度特征应能够反映当前生成词元位置的可靠性。同时,置信度特征本身应该计算轻量,不会引入额外的训练。本文在GSM8K数据集[10]上开展实验,分析不同置信度特征与模型推理准确率之间的关系。对于数据集中的每一道题,实验不仅记录模型最终回答是否正确,还保存了模型

在生成过程中各词元位置对应的 logits 向量,并在此基础上提取相关的置信度特征。采用 Top-1 与 Top-2 的 logits 之差,即 Margin,作为置信度特征的结果,如图 1 所示。

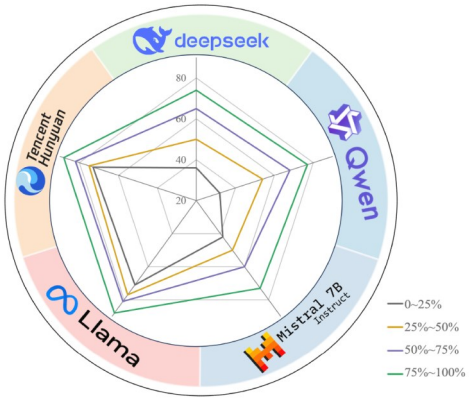

图 1 Margin 分位与推理准确率的关系

Figure 1 Relationship between margin quantiles and inference accuracy

图 1 展示了 5 个大语言模型按照 Margin 分位进行分桶后的统计结果,具体实验设置见第 3.1 节。由图可知,Margin 与推理准确率整体呈正相关。随着 Margin 分位区间升高,模型回答正确的样本比例也相应增加。在部分模型中,这一差异表现得较为明显,最高四分位样本的准确率约为最低四分位样本的 2 倍。上述结果说明,Margin 作为一种无需额外训练的轻量化信号,能够较好地刻画生成过程中的词元置信度。后续消融实验将进一步检验该特征在选择性协作机制中的作用。

本文将 Margin 作为选择性协作的置信度特征。利用置信度特征识别低置信度词元,优先在基模型低置信度位置引入多模型协作,在维持协同体推理准确率的同时控制协作开销。进一步,本文提出了一种基于置信度门控的词元级协作机制 FuseGate。FuseGate 作为外接门控模块可被集成到现有的词元级并联协作框架中。FuseGate 不需要进行预训练,也不需要改动基模型原有的解码流程。现有的词元级并联协作框架在集成 FuseGate 后,对于参与协作的协同体,在每一个解码步中,当基模型的词元置信度较高时,协同体直接采用基模型输出;当基模型词元置信度较低时,则会调用协作模型与基模型一同在词元置信度低的解码步执行多模型词元集成。通过这种选择性协作的方式,FuseGate 能够把有限的协作预算优先分配到更需要协作的解码步,从而实现生成质量与推理效率的更优权衡。

本文的主要贡献概括如下:

(1)提出一种基于置信度门控的词元级协作机制 FuseGate。该机制作为外接门控模块集成到现有并联协作框架中,在不改变原框架主解码流程的前提下,实现面向分位点参数约束的选择性协作调用。

(2)设计了一种基于词元级置信度特征的协作触发机制,动态判断是否触发多模型协作,从而将额外计算集中于低置信度位置。实验结果表明,该机制采用的置信度特征的效果优于其他常见置信度特征。

(3)在多组异构模型组合和多类基准任务上的实验结果表明,相较于全程并联协作基线,引入 FuseGate 后的并联框架在大幅降低单词元生成平均时延的同时能够基本维持准确率。其中,在 DuetNet[6] 框架上的集成实验中,在维持全并联协作推理准确率基本不变的情况下,单词元生成平均时延降低约 46%。

1 相关工作

单一模型范式在实际应用中逐渐显露不足。不同任务对模型能力的要求并不一致:有的任务需要较强的长文本理解能力,有的任务依赖复杂推理过程,有的任务则更看重模型的响应速度。为解决这一问题,AI-Model Network 范式逐渐受到关注。该范式通过统一接口连接多个具有不同能力侧重的模型,并将其组织成一个可调度的多模型系统。在模型互联网中,模型之间的协作并不局限于单一形式,而是可以发生在不同粒度上。模型协作按协作粒度划分为文本级协作和词元级协作。文本级协作通常面向完整输入或完整输出进行决策,依据任务类型、输入特征或模型能力差异,选择更适合当前任务的模型。也可以采用级联调用机制,默认使用基模型独立输出,在基模型难以给出可靠结果时,再引入能力更强、成本更高的模型参与处理[11-15]。

词元级协作则更深入地嵌入到自回归解码过程中。词元级协作会在每一个解码步记录多个模型在相同上下文下输出的对下一词元的预测信号,由集成或决策机制确定当前解码步的最终输出。与文本级协作相比,由于不同模型在同一解码位置上的预测结果可以直接对齐和比较,系统能够在生成过程中持续利用多模型信息,模型之间的互补性得到了更好的利用。

依据最终输出形成机制的差异,现有词元级协作方法通常可分为概率分布集成与一致性协同。概率分布集成方法在单步解码阶段对多个模型的预测分布进行直接融合。但在异构模型协作场景中,不同模型通常使用不同的分词器和词表,这导致不同模型在同一上下文下生成的候选词元并不对应。早期研究主要关注如何高效地把不同词表空间中的预测结果进行对齐:Yu 等人[16]提出通过学习投影矩阵,将不

同模型的输出分布映射到统一空间，以实现细粒度融合。但是，这种方法需要额外训练，有一定的附加开销。为降低对齐成本，后续研究开始尝试更加轻量的候选筛选机制和分布截断机制；Yao等人[17]提出仅保留每个解码步中概率较高的部分候选词元参与融合，这样可以降低在全词表范围内的对齐与聚合开销；进一步地，Wang等人[6]采用Top-K与Top-P联合截断策略，对低概率候选词元进行过滤，在减少聚合噪声的同时，也提升了推理准确率。

与概率分布集成思路不同，一致性协同并不急于把多个模型的下一词元概率分布合并到一起，它关注的是模型之间在当前解码位置上是否形成了相近判断。这种一致或分歧本身就可以作为一种信号：当一致性较强时，说明当前候选结果更可靠；分歧明显时，则提示该位置存在较高不确定性。围绕这一思路，已有方法给出了不同实现路径：Hao等人[18]提出投票式协作框架，将多个模型对下一词元的输出视作一种集体投票，并利用高得分候选来反映模型间的强一致性；Xu等人[19]在束搜索过程中引入集体困惑度，用它衡量候选路径是否可靠。如果多个协作模型都认为某一路径的困惑度偏高，该路径便会被提前剪枝，从而避免低质量候选继续扩展；Huang等人[20]则将研究重点进一步转向模型内部交互，尝试通过共享特征层表示和同步中间状态，提升异构模型之间的协作能力。

现有的词元级协作，无论是采用概率分布集成方法，还是采用一致性协同方法，主流做法仍然是逐词元同步的全程协作，即在生成过程中的每一个解码步，多个模型都需要持续参与推理和协同。这样的方式虽然能够在一定程度上提升生成准确性，但在实际部署时，往往也会带来高计算开销和通信开销。原因在于：一方面，多个模型需要贯穿整个解码过程持续工作，这会使计算资源被长时间占用；另一方面，逐词元同步的协作机制，也会限制系统根据生成难度动态调整协作预算的能力，使其难以实现更加灵活的选择性协作和高效调度。

因此，在预算约束和时延约束下，如何设计一种轻量可泛化的触发信号，并据此对协作时机和协作预算进行动态控制，从而在生成质量和推理效率之间取得更好的平衡，仍然是模型互联网中词元级协同推理亟待解决的关键问题。基于这一考虑，本文提出了一种基于置信度门控的词元级协作机制FuseGate。

2 置信度门控的词元级协作机制

本节详细介绍FuseGate的整体框架及其门控策略。首先给出基于置信度门控的词元级协作推理流程，然后介绍基于滑动窗口的动态置信度门控机制。

2.1 机制概述

传统词元级并联协作推理采用全程并联策略，需在每一个解码步等待参与协作的模型都生成下一词元，这带来显著的单词元生成平均时延。为减少并联协作中需要调用协作模型的解码步数，以提升单词元生成效率，本文引入基于置信度选择性协作机制FuseGate。FuseGate以基模型为中心将推理过程划分为两个阶段：独立推理阶段与并联推理阶段。在默认情况下，下一词元由基模型独立预测，整个解码过程保持较低的计算开销。随着生成推进，若某一位置的置信度信号表明基模型判断不够稳定，FuseGate将把该位置标记为需要协作的解码步，并调用协作模型参与logits聚合。

假设一共有N个模型参与推理，其中，M_1表示基模型，$M_n, n\in\{2,3,\cdots,N\}$表示协作模型。图2展示了FuseGate的词元级协作推理流程。在每个解码步中，系统首先由基模型生成候选词元，并计算置信度；随后再根据门控结果决定是否触发协作模型参与聚合；最后完成聚合采样、序列更新与终止。整个过程可以进一步划分为5个子步骤，具体说明如下：

步骤1：输入处理。在推理开始前，基模型先将原始文本转化为词元序列，并映射为对应的词元标识符（Token Identifier，ID），从而得到模型可处理的数值化输入序列I_0。

步骤2：基模型候选生成。在解码步t开始时，系统会把提示词和已经生成的上下文进行拼接，形成当前输入序列I_t。该序列会作为本轮推理的输入，用来生成下一词元的预测分布。

在解码步t的开始阶段，系统只调用基模型M_1进行计算，不会立即引入协作模型。基模型在给定输入I_t条件下，输出的未归一化logits向量记为

$$\mathbf{logits}_{M_1}=\mathrm{G}_{M_1}(I_t) \tag{1}$$

其中，G_{M_1}表示基模型对应的生成函数，它的输出结果是基模型在词表中对各个词元给出的未归一化得分向量。$\mathbf{logits}_{M_1}$的维度与基模型词表大小$|V_1|$相同。对任意词元$u\in V_1$，其对应得分$[\mathbf{logits}_{M_1}]_u$可以反映基模型在当前解码步中对该词元的置信度，得分越高，说明该词元越有可能被基模型选为当前输出候选。

为了过滤掉得分较低、贡献相对有限的候选词元，并控制后续计算过程中可能产生的额外开销，现有的并联方法会对$\mathbf{logits}_{M_1}$进行截断。具体来说，现有的并联方法会采用基于Top-K与Top-P的截断策略，先保留得分最高的K个候选词元，再在这些候选词元中选取累计概率达到阈值P的最小集合。联合

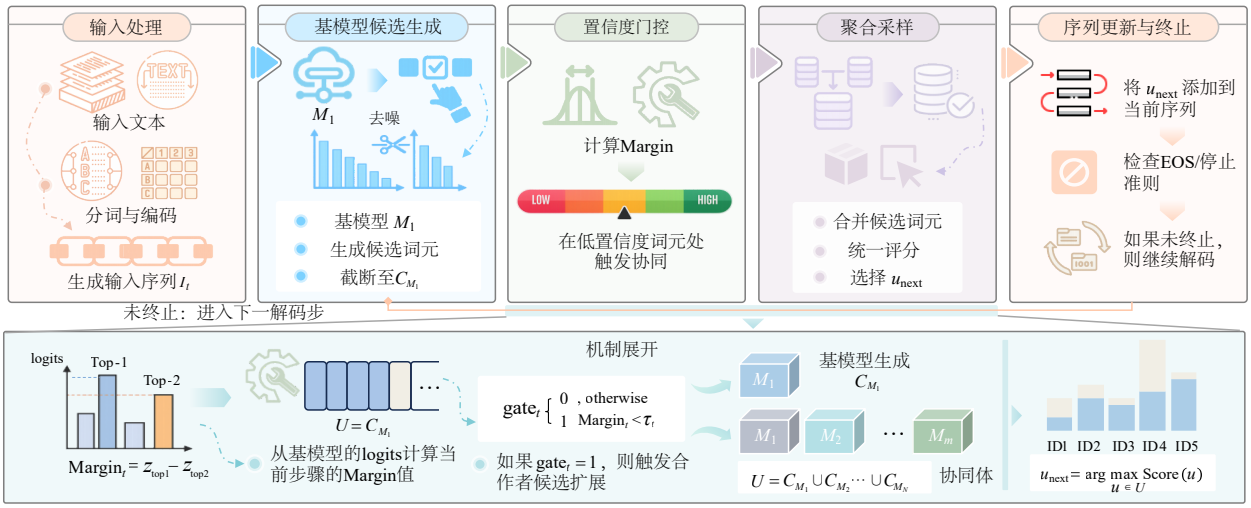

图2 基于置信度门控的词元级协作机制

Figure 2 Confidence-gated mechanism for token-level parallel collaboration

截断后的候选词元集合记为

$$C_{M_1} = F(\mathbf{logits}_{M_1}, K, P) \tag{2}$$

式(2)中的 $F(\cdot)$ 表示联合截断算子。步骤1和步骤2默认计算基模型候选词元集合与对应logits向量用于后续的协作判断，仅当判断触发协作时，才进一步计算协作模型的候选词元集合和对应logits向量。

步骤3：置信度门控。在完成基模型词元候选集合 C_{M_1} 的生成后，FuseGate会进一步利用基模型logits向量对当前解码步的置信度进行判定。记 $\mathbf{logits}_{M_1}$ 中最大值与次大值分别为

$$z_{top1} = \max(\mathbf{logits}_{M_1}) \tag{3}$$

$$z_{top2} = 2\text{nd}\max(\mathbf{logits}_{M_1}) \tag{4}$$

其中，$\max(\cdot)$ 表示取最大元素，$2\text{nd}\max(\cdot)$ 表示取次大元素。定义解码步 t 的Margin为

$$\text{Margin}_t = z_{top1} - z_{top2} \tag{5}$$

本文采用 z_{top1} 与 z_{top2} 的logits差值Margin来描述候选词元的置信度。Margin越大表示最优候选的相对优势越明显，基模型对当前词元的预测分布越确信；Margin越小表示基模型生成的该词元置信度较低。

FuseGate是一个轻量级门控模块，它会把基模型的Margin作为置信度特征，并据此判断当前解码步是否需要进行协作。对于每一个基模型，FuseGate会维护一个对应的FuseGate实例。FuseGate会在每个解码步计算置信度信号，并维护长度为 W 的滑动窗口 H 记录最近若干步的置信度特征，在分位点参数 q 的约束下，输出二值的协作门控变量 gate_t：

$$\text{gate}_t = \text{FuseGate}(\text{Margin}_t, q, W), \text{gate}_t \in \{0, 1\} \tag{6}$$

在判断当前解码步是否需要调用协作模型时，FuseGate会一同保存基模型在该解码步的Margin值。该记录会为后续词元的选择性协作提供参照。

步骤4：聚合采样。在门控判定完成后，系统根据 gate_t 对基模型和协作模型进行候选生成与聚合：

$$\text{聚合采样} = \begin{cases} \text{基模型独立生成 } \text{gate}_t = 0 \\ \text{并联协作生成 } \text{gate}_t = 1 \end{cases} \tag{7}$$

当 $\text{gate}_t = 0$ 时，系统会由基模型进行独立推理，此时不会触发协作模型的计算过程，系统仅保留基模型的 $\mathbf{logits}_{M_1}$ 与 C_{M_1}。当 $\text{gate}_t = 1$ 时，系统进入并联推理阶段。此时，对每个协作模型 $M_n, n \in \{2, 3, \cdots, N\}$，系统都会基于相同输入文本 I_t 计算对应的logits向量，并执行与基模型一致的联合截断：

$$\mathbf{logits}_{M_n} = G_{M_n}(I_t), n \in \{2, 3, \cdots, N\} \tag{8}$$

$$C_{M_n} = F(\mathbf{logits}_{M_n}, K, P), n \in \{2, 3, \cdots, N\} \tag{9}$$

随后，将当前解码步中实际参与聚合的各模型的候选词元集合取并集构成该解码步词元的统一候选空间 U：

$$U = C_{M_1} \cup C_{M_2} \cdots \cup C_{M_N} \tag{10}$$

对统一候选集合 U 中的每个候选词元 u，将其在各模型中的logits进行逐词元累加，得到该词元的累计得分，记为 $\text{Score}(u)$：

$$\text{Score}(u) = \sum_{n=1}^{N} [\mathbf{logits}_{M_n}]_u, u \in U \tag{11}$$

FuseGate的门控作用体现在求和的参与项是否包含协作模型。当 $\text{gate}_t = 0$ 时，系统只计算单模型得分；当 $\text{gate}_t = 1$ 时，协作模型logits才被计算并纳入累加。在得到聚合后的词元候选空间 U 以及对应的累计得分后，系统会从词元候选空间 U 中选取下一个词

元u_{next}。词元的聚合得分越大，表明该词元在多个模型中的综合支持度越高，也越有可能成为当前解码步的一致输出。下一个词元可以表示为

$$u_{\text{next}}=\text{argmax}_{u\in U}\text{Score}(u) \tag{12}$$

步骤5：序列更新与终止。在确定u_{next}后，系统会将该词元追加到当前输入序列中，并同步更新所有模型的上下文状态。随后，系统会检查当前生成过程是否已经满足终止条件。如果生成词元为终止符（End of Sequence，EOS），或者生成长度已达到预设的最大值，则结束本轮解码；否则，则会把更新后的序列作为新的输入，并返回步骤2继续执行。最后，将u_{next}追加到输出序列中，并更新输出结果：

$$I_{t+1}=I_t\otimes u_{\text{next}} \tag{13}$$

$$A_{t+1}=A_t\otimes u_{\text{next}} \tag{14}$$

本文将算法流程形式化为算法1。

2.2 基于滑动窗口的置信度门控

为在词元级推理过程中更精准地触发协作，本文提出了一种基于滑动窗口的置信度门控模块FuseGate。在每一个解码步中，FuseGate接收基模型输出的logits向量，计算当前解码步的置信度信号。FuseGate会参考最近W步的置信度信号变化，结合分位点控制参数q，判断是否引入协作模型参与推理。

固定阈值门控假设各个解码位置可以共享同一判断标准。然而，在长文本生成场景中，这一假设并不适用。随着大模型不断地进行自回归解码，模型对不同词元的置信度会持续变化。而FuseGate采用滑动窗口机制保存近W步的Margin信息，并将其形成局部置信度分布作为门控判断的依据。

为此，FuseGate会在推理过程中维护一个容量为W的滑动窗口H，用于记录最近W个解码步的Margin值。在解码步t时，记窗口状态为

$$H=\{\text{Margin}_{t-k+1},\text{Margin}_{t-k+2},\cdots,\text{Margin}_t\},k=\min\{W,t\} \tag{15}$$

为刻画近期生成状态，本文使用容量为W的先进先出队列维护历史窗口H。在每个解码步结束后，当前得到的Margin会被加入该队列。若队列中的样本数量超过窗口容量，则按照进入顺序删除最早的样本，保证窗口内仅保留最近W个解码步的Margin信息。

在推理开始阶段，窗口内样本数量较少，历史置信度信息并不充分。如果这时直接计算动态阈值，容易造成门控判断不稳定。因此，当历史窗口H还没被填满时，FuseGate会采用保守策略，默认触发协作，并将当前Margin_t写入窗口，以积累后续门控所需的置信度样本。当窗口填满后，FuseGate才会依据历史窗口H与分位点参数q，计算当前解码步对应的动态阈值τ_t：

$$\tau_t=\text{Quantile}(H,q) \tag{16}$$

Quantile(H,q)用于计算历史窗口H的q分位点，其中，参数q决定动态阈值在历史置信度分布中的位置。随后，FuseGate将当前解码步的Margin_t与阈值进行比较，并据此得到二值门控变量：

算法1 基于置信度门控的词元级协作推理算法

输入：初始输入I_0；窗口长度W；截断参数K,P；分位点参数q，协作模型$M_n,n\in\{2,3,\cdots,N\}$，基模型为M_1，门控函数FuseGate，最大输出长度$T_{\max}$

输出：生成序列A

1. 初始化$I_t\leftarrow I_0,A\leftarrow\varnothing,t\leftarrow 0,u_{\text{next}}\leftarrow\varnothing$
2. WHILE（$u_{\text{next}}\neq$EOS 且 $t<T_{\max}$）：
   //若终止条件尚未满足，解码过程将继续进行
3. $\mathbf{logits}_{M_1}=G_{M_1}(I_t)$
   //基于当前上下文计算下一词元的logits向量
4. $C_{M_1}=F(\mathbf{logits}_{M_1},K,P)$
   //基于Top-K与Top-P策略筛选候选词元
5. $z_{\text{top1}}=\max(\mathbf{logits}_{M_1})$
6. $z_{\text{top2}}=2\text{nd}\max(\mathbf{logits}_{M_1})$//计算当前解码步的置信度
7. $\text{Margin}_t=z_{\text{top1}}-z_{\text{top2}}$
8. $\text{gate}_t=\text{FuseGate}(\text{Margin}_t,q,W)$
   //生成二值协作门控信号gate_t，控制当前步是否触发并联协作
9. IF $\text{gate}_t=1$ THEN：
   //若当前Margin_t低于动态置信度阈值，则触发并联协作
10. FOR n=2,3,$\cdots$,N DO：
11. $\mathbf{logits}_{M_n}=G_{M_n}$
12. $C_{M_n}=F(\mathbf{logits}_{M_n},K,P)$
13. END FOR
14. $U=C_{M_1}\cup C_{M_2}\cdots\cup C_{M_N}$
    //构造协作模式下的统一候选空间
15. $\text{Score}(u)=\sum_{n=1}^{N}[\mathbf{logits}_{M_n}]_u,u\in U$
    //聚合各模型对候选词元的支持得分
16. ELSE：
17. $U=C_{M_1}$
18. $\text{Score}(u)=[\mathbf{logits}_{M_1}]_u,u\in U$
    //聚合基模型对候选词元的支持得分
19. END IF
20. $u_{\text{next}}=\text{argmax}_{u\in U}\text{Score}(u)$
    //在当前候选空间中选择综合得分最高的词元
21. $I_{t+1}=I_t\otimes u_{\text{next}}$
22. $A_{t+1}=A_t\otimes u_{\text{next}}$
    //将当前词元追加到输入上下文和生成序列中
23. $t=t+1$
24. END WHILE
25. RETURN A

$$\text{gate}_t=\begin{cases}1 & \text{Margin}_t<\tau_t\\ 0 & \text{Margin}_t\geqslant\tau_t\end{cases} \tag{17}$$

$\text{gate}_t=1$ 表示当前解码步的置信度低于动态阈值，此时系统会触发协作模型参与并联推理；$\text{gate}_t=0$ 表示当前解码步的置信度不低于阈值。在此情况下，无需引入协作模型，解码过程退化为只由基模型参与的独立推理。

通过上述设计，FuseGate具有近似比例控制的特点。由于 τ_t 是根据历史窗口 H 计算得到的第 q 分位点，因此参数 q 可以作为统一的控制参数，用来直接调节协作触发频率。较大的 q 会提高协作被触发的概率，使更多低置信度位置能够引入协作模型；而较小的 q 则会减少协作模型的调用次数，进而减少单词元生成平均时延。

总体来看，FuseGate会将当前解码步的Margin与历史窗口 H 中的动态分位点阈值进行比较，从而自适应地决定在当前解码位置是否触发多模型选择性协作。整个判断过程只依赖基模型当前解码步的logits向量及长度为 W 的历史窗口。因此，该机制引入的额外开销较低，具备较好的框架兼容性。

本文将算法流程形式化为算法2。

算法2　基于滑动窗口的置信度门控

输入：当前解码步置信度 Margin_t；窗口长度 W；分位点参数 q

输出：当前解码步协作门控信号 gate_t

维护历史滑动窗口 H，窗口最大长度为 W
IF $|H|<W$ THEN：
　$\text{gate}_t=1$
　//历史滑动窗口内的样本没有填满窗口时默认触发协作
ELSE：
　$\tau_t=\text{Quantile}(H,q)$
　//当历史滑动窗口的样本填满窗口时，计算历史滑动窗口的第 q 分位点作为动态阈值
　IF $\text{Margin}_t<\tau_t$ THEN：
　　$\text{gate}_t=1$
　ELSE：
　　$\text{gate}_t=0$
　END IF
END IF
$H\leftarrow\text{Append}(H,\text{Margin}_t)$
　//将当前 Margin_t 写入历史滑动窗口
IF $|H|>W$ THEN：
　$H\leftarrow\text{Dequeue}(H)$
　//移除历史滑动窗口中最早进入的一个Margin
END IF
RETURN gate_t

3　实验分析

3.1　实验设置

为验证FuseGate在异构模型并联协作场景中的适用性，本文选取5个参数规模相近，但在模型架构、训练数据来源和分词方式上存在差异的开源大语言模型，以两个模型为一组构建异构模型组合。相关模型信息如表1所示。

表1　本文所用模型

Table 1　Models used in the experiments

简称	模型名称	发布者
Ds	deepseek-llm-7b-chat[21]	深度求索
Q1	qwen1.5-7B-Chat[22]	阿里巴巴
M3	mistral-7b-instruct-v0.3[23]	Mistral AI
L3	Llama-3.1-8B-Instruct[24]	Meta
Hy	Hunyuan-7B-Instruct[25]	腾讯

本节所有实验均在配备2 × 80 GB A100的服务器上完成，各个模型分别独立部署在单张GPU上。为了保证实验比较结果的可靠性，除协作策略存在差异外，各方法在提示词模板、输出格式约束和截断策略等设置上保持一致。同时，实验过程也采用统一的实验流程和评测标准。由此，不同方法之间的性能差异，主要来自协作机制本身，而不是由解码配置或实现细节的不同所造成。

表1列出了本文实验所采用的5个开源大语言模型。为考察FuseGate在异构并联协作场景中的适用性，实验选取了参数规模相近但模型架构、训练数据来源和分词方式存在差异的模型，组成异构模型集合。

本文采用表2所列的4类数据集对FuseGate进行评估。其中，SimpleMath[26]主要考察规则驱动场景下的计算稳定性；BoolQ[27]用于评估段落条件下的阅读理解与事实一致性判断能力；GSM8K[10]用于检验多步数学推理中的长链路计算；MMLU[28]则用于考察跨学科知识理解与综合判别能力。上述设置涵盖了从低语义确定性任务到高复杂度推理任务，可以对FuseGate在不同认知维度下的触发有效性、协作收益与泛化能力进行较为全面的分析。

结合预实验结果以及第3.2节中的参数敏感性分析，本文为后续实验设定统一的实验配置，以保证不同方法之间的公平比较。所有对比方法均采用相同的硬件部署方式、模型组合、提示词模板和评测脚本，解码过程均采用贪心解码策略，不引入其他生成参数。实验采用统一的超参数设置：最大生成长度 $T_{\max}=512$，窗口大小设为 $W=1\,024$，分位点参数设为 $q=0.5$，用于控制选择性协作的平均触发强度。同

表2 本文所用数据集及提示词

Table 2 Datasets and prompting templates used in the experiments

数据集名称	题目类型	数据集介绍	提示词
SimpleMath[26]	简单计算题	计算$a+b\times c+d-e\times f$,其中,a~f为0~30的随机整数	What is the answer to the question? The final answer must be an Arabic numeral.\nQuestion:{question}\nAnswer:
BoolQ[27]	判断题	包含15 K情境推理问题,每个问题都附带文段。大模型需要阅读给定文段并用T或F给出判断。用于评价大模型的阅读理解与情境推理能力	Read the passage and answer the question by replying true or false. The passage is:{passage}.\nThe question is:{question}\nAnswer:
GSM8K[10]	推理计算题	由8.5 K道高质量的小学数学文字题组成。这些问题需要进行多步的数学推理。用于评估大模型的长链路计算的稳定性	Question:{question}\nLet's think step by step.\n
MMLU[28]	选择题	覆盖57个学科知识的单选题,从基础教育到专业考试层面。题目需要大模型回答A、B、C、D中的某一个选项。用于衡量大模型在跨学科知识和推理上的综合能力	Answer the question by replying A,B,C or D.\nQuestion:{question}\nAnswer:

时,为避免词元候选集合构造差异影响实验结论,本文参照DuetNet的设置,将联合截断策略的参数设为$\mathrm{Top}-K=10$,$\mathrm{Top}-P=0.75$[6]。

本文将对比方法分为以下4类:Single-Model代表单模型独立推理方法,用于刻画基模型单独推理的能力;UniTE和DuetNet代表词元级并联协作方法,分别对应概率向量融合和logits向量融合两种并联融合方式;UniTE+FuseGate和DuetNet+FuseGate代表选择性门控的词元级协作方法,用于验证FuseGate在不同词元级并联框架中的适用性;StaticGate、GaCGate和RandGate是消融实验中的对比方法。具体如下:

(1)Single-Model:单模型独立推理方法。该方法仅使用基模型独立解码,不引入任何并联协作机制,作为无协作条件下的对比基线。

(2)UniTE:基于概率向量融合的词元级并联协作方法。该方法在每个解码步保留各模型的Top-K候选词元,并以候选集合的并集作为对齐与融合空间,在此基础上完成跨模型概率融合。

(3)DuetNet:基于logits向量融合的词元级并联协作方法。该方法在每个解码步联合Top-K与Top-P截断构建候选集合,并在候选空间内直接聚合不同模型的logits。

(4)UniTE+FuseGate:概率融合框架下的选择性协作方法。该方法在词元级并联框架UniTE中嵌入FuseGate模块,仅在门控触发的词元位置执行UniTE并联协作,其余位置由基模型独立完成解码,用于比较选择性协作在概率融合框架下的效果。

(5)DuetNet+FuseGate:logits向量融合框架下的选择性协作方法。该方法在词元级并联框架DuetNet中嵌入FuseGate模块,仅在门控触发的词元位置执行DuetNet并联协作,其余位置由基模型独立完成解码,用于比较选择性协作在logits融合框架下的效果。

(6)DuetNet+StaticGate:静态阈值门控方法。该方法在DuetNet框架中采用固定阈值门控策略,以基模型当前解码步Top-1与Top-2的logits差值,即Margin作为置信度信号,当词元置信度信号低于预设阈值时触发并联协作,其余词元由基模型独立完成解码。该方法用于检验FuseGate中滑动窗口动态阈值机制的作用。

(7)DuetNet+GaCGate:基于Top-1 logits置信度信号的门控方法。该方法在DuetNet框架中引入文献[16]所提出的阈值门控思路,本文将该基线记为GaCGate。该方法以每个解码步logits向量的Top-1值作为置信度信号,当置信度低于阈值时触发并联协作。该方法用于比较不同置信度信号对门控效果的影响。

(8)DuetNet+RandGate:随机门控方法。该方法在DuetNet框架中引入随机门控策略,对每个词元以与FuseGate对应的概率独立触发并联协作。该方法用于验证FuseGate的性能差异是否来自协作触发位置的有效选择。

3.2 参数分析

本节探讨关键超参数窗口大小W与分位点参数q对FuseGate性能的影响。

3.2.1 窗口大小(W)

窗口大小W决定FuseGate中滑动窗口机制统计的置信度时所采用的历史范围。为分析窗口大小W的影响,本小节在固定模型组合与提示词设置下,比较不同窗口大小下协同体的推理准确率及协作率。

从表3可以看到,不同W的设置并未表现出显著的性能差异。该结果表明,FuseGate对窗口大小并不

表3 W值分析

Table 3 Effect of window size W

窗口大小	准确率/%	协作率/%
32	60.23	49.05
128	61.58	49.16
512	61.61	48.90
1 024	**62.11**	**49.36**
2 048	62.09	49.14

注:加粗表示各组实验最优结果。

敏感,较大的窗口有助于提高协作触发的稳定性。综合准确率与协作率,本文在后续实验中采用W = 1 024作为窗口大小设置。

3.2.2 分位点参数(q)

FuseGate中的分位点参数q主要用于控制低置信度位置的判定尺度,并间接影响协作模型介入解码的比例。围绕这一参数,本文在多组异构模型组合上开展对比实验,观察不同q设置下推理准确率的变化。相关结果如图3所示。

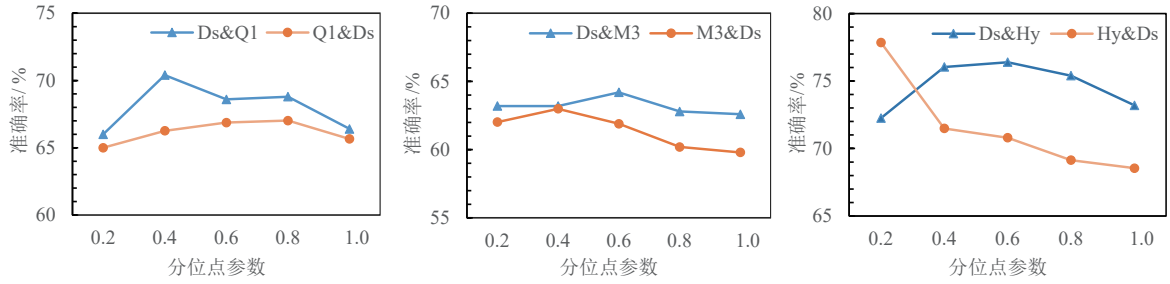

图3 不同分位点参数q下模型组合推理准确率

Figure 3 Inference accuracy of different model combinations under varying quantile parameter q

从图3可以看出,在不同模型组合下,推理准确率随分位点参数q的逐渐变大呈现出非单调变化,部分组合在q较大时准确率会出现回落。这说明分位点参数并非越高越好,而是存在一个与模型能力相关的较优取值区间。

当q较小时,协作触发次数较少,协作模型很难在一些关键的低置信度位置提供有效补充,因此整体带来的收益比较有限;随着q逐渐增大,更多低置信度词元能够得到协作模型的支持,推理准确率随之提升;不过,当q继续增大时,协作模型会频繁地介入到基模型的高置信度位置,这不仅增加了额外的词元生成时延,还可能引入来自协作模型中的噪声,从而在部分模型组合上造成准确率下降。

进一步分析发现,q对推理性能的影响,与基模型和协作模型之间的相对能力差异有关。在基模型和协作模型能力接近的情况下,中等取值的q往往可以带来更均衡的协作触发效果。一方面,系统能够借助协作模型补偿低置信度位置;另一方面,也可以减少协作模型被过度调用时可能产生的干扰。对于基模型弱于协作模型的组合,适当提高q能够增加协作覆盖范围,使更强的协作信号更多参与解码过程,进而提升整体推理性能。相较之下,当基模型强于协作模型时,过高的q会增加协作模型的介入频率,使噪声信号或不稳定预测进入聚合过程,进而导致生成质量下降。

总体来看,分位点参数q越大,协作触发频率越高,单词元生成平均时延也会增加,但准确率提升会逐渐放缓。综合不同模型组合的实验结果,本文将分位点参数设置为q = 0.5。

3.3 准确率分析

为考察FuseGate在不同模型组合下的适用性,本小节根据基模型与协作模型之间的相对能力差异,将实验模型组合划分为3种类型:二者性能相近、基模型性能更强,以及协作模型性能更强。在此基础上,本文进一步在多类测试数据集上比较不同方法的平均推理准确率,以观察FuseGate在不同能力关系下的表现变化,相关结果汇总于表4。表4中,“Ds&Q1”表示一组基模型与协作模型的搭配关系,其中,Ds作为基模型,Q1作为协作模型。指标Y%用于表示相对于基模型单独推理时的准确率变化幅度。当该数值为正时,说明协作方法带来了性能提升;当数值为负时,则表示协作后准确率有所下降。

基于表4的实验结果,主要有以下发现:

(1)当基模型与协作模型性能比较接近时,并联协作优于基模型独立推理。相较独立推理,UniTE与DuetNet的平均准确率分别提升3.40%和5.25%;在各自全程并联的基线之上,引入FuseGate后又分别进一步提升约2.40%和2.15%。这一结果说明,确实存在一定比例的无效协作,而FuseGate门控机制能够减少不必要的并联介入,从而控制推理开销,维持了整体的推理准确率。

(2)当基模型性能优于协作模型时,全程并联协作容易导致准确率降低,而FuseGate在部分模型组合中能够缓解推理准确率下降,甚至在个别组合中可以

表4 FuseGate下并联协作框架平均推理准确率对比

Table 4 Comparison of average inference accuracy across parallel collaboration frameworks with FuseGate

场景	组合	基模型	UniTE		DuetNet	
			UniTE	UniTE+FuseGate	DuetNet	DuetNet+FuseGate
性能相近	Ds&Q1	0.605	0.638(+3.30%)	0.680(+7.50%)	**0.671(+6.60%)**	0.669(+6.40%)
	Q1&Ds	0.603	0.638(+3.50%)	0.644(+4.10%)	0.642(+3.90%)	**0.687(+8.40%)**
	平均	0.604	0.638(+3.40%)	0.662(+5.80%)	0.657(+5.25%)	**0.678(+7.40%)**
基模型的性能优于协作模型	M3&Q1	0.651	0.638(−1.30%)	0.681(+3.00%)	0.645(−0.60%)	**0.691(+4.00%)**
	M3&Ds	0.651	0.666(+1.50%)	**0.692(+4.10%)**	0.574(−7.70%)	0.654(+0.30%)
	L3&M3	**0.724**	0.657(−6.70%)	0.682(−4.20%)	0.567(−15.70%)	0.614(−11.00%)
	L3&Ds	**0.724**	0.636(−8.80%)	0.657(−6.70%)	0.649(−7.50%)	0.562(−16.20%)
	L3&Q1	**0.724**	0.622(−10.20%)	0.682(−4.20%)	0.687(−3.70%)	0.695(−2.90%)
	Hy&Q1	0.697	0.627(−7.00%)	0.689(−0.80%)	**0.716(+1.90%)**	0.694(−0.30%)
	Hy&Ds	**0.697**	0.656(−4.10%)	0.647(−5.00%)	0.673(−2.40%)	0.680(−1.70%)
	Hy&M3	0.691	0.710(+1.90%)	**0.757(+6.60%)**	0.670(−2.10%)	0.702(+1.10%)
	平均	**0.695**	0.652(−4.34%)	0.686(−0.90%)	0.648(−4.73%)	0.662(−3.34%)
基模型的性能弱于协作模型	Ds&M3	0.582	0.665(+8.30%)	**0.676(+9.40%)**	0.641(+5.90%)	0.628(+4.60%)
	Q1&M3	0.581	0.638(+5.70%)	0.648(+6.70%)	**0.685(+10.40%)**	0.633(+5.20%)
	M3&L3	0.653	0.660(+0.70%)	**0.666(+1.30%)**	0.567(−8.60%)	0.598(−5.50%)
	Ds&L3	0.582	0.621(+3.90%)	**0.657(+7.50%)**	**0.657(+7.50%)**	0.621(+3.90%)
	Q1&L3	0.581	0.623(+4.20%)	0.650(+6.90%)	**0.687(+10.60%)**	0.672(+9.10%)
	Q1&Hy	0.581	0.632(+5.10%)	0.676(+9.50%)	**0.716(+13.50%)**	0.673(+9.20%)
	Ds&Hy	0.582	0.694(+11.2%)	0.695(+11.3%)	0.673(+9.10%)	**0.709(+12.70%)**
	M3&Hy	0.653	0.665(+1.20%)	0.675(+2.20%)	0.669(+1.60%)	**0.718(+6.50%)**
	平均	0.599	0.650(+5.04%)	**0.668(+6.85%)**	0.662(+6.25%)	0.657(+5.71%)

注:加粗表示各组实验最优结果。

将性能退化转化为性能提升。以“M3&Q1”组合为例,引入FuseGate后,UniTE与DuetNet在该组合下的表现,都由原来的准确率下降转变为超过3%的准确率提升。在其余模型组合中,FuseGate虽然没有完全把准确率下降转化为性能提升,但整体上仍然能够缩小下降幅度。这说明,按需触发协作能够在一定程度上降低弱协作模型信号对强基模型的干扰。

(3)若协作模型本身具有更强的推理能力,多模型并联往往能够获得较为稳定的收益。强协作模型可以在解码过程中提供质量更高的候选信号,对基模型的局部不确定性形成补偿。引入FuseGate后,这种由强协作模型带来的准确率提升基本没有被削弱。这说明FuseGate会把协作模型的参与更多保留给关键解码位置,使额外计算开销得到压缩。

进一步分析表明,并联协作的效果因模型能力的不同而表现出明显差异。在基模型与协作模型能力接近时,并联协作可获得稳定的推理准确率提升,FuseGate机制则能在此基础上进一步扩大增益。若基模型性能优于协作模型,全程并联协作会因弱协作模型的干扰而出现推理准确率下降,FuseGate通过减少无效并联协作缓解推理准确率的损失。当基模型弱于协作模型时,推理准确率的提升主要源于强协作模型的高置信度词元,FuseGate则能以较低的计算开销保留该收益,并维持推理结果的稳定性。

3.4 单词元生成平均时延分析

与3.3节的准确率分析对应,本小节整理了不同基模型和协作模型的能力关系下的单词元生成平均时延。实验结果如表5所示,表中时延数据的单位为毫秒每词元(millisecond per token, ms/token)。其中,UniTE+FuseGate与DuetNet+FuseGate分别表示在UniTE、DuetNet框架中引入FuseGate门控;“Ds&Q1”等组合表示基模型与协作模型的搭配关系;括号中的(Y%)表示相对于DuetNet的单词元生成平均时延的相对降幅,数值为正表示时延下降,数值为负表示时延增加。

从单词元生成时延来看,FuseGate对单词元生成时延的改善明显。表5中的结果显示,在双模型协作场景下,无论是与UniTE结合,还是与DuetNet结合,引入FuseGate后的平均解码时延均低于对应的全程并联协作方法。以DuetNet为参照,UniTE+FuseGate和DuetNet+FuseGate的平均时延分别下降42.4%和45.1%。

结合表4的准确率结果进一步分析可知,引入

表5 FuseGate下并联协作的单词元生成平均时延对比

Table 5 Comparison of average per-token generation latency for parallel collaboration with FuseGate

场景	组合	DuetNet	UniTE	DuetNet+FuseGate	UniTE+FuseGate
性能相近	Ds&Q1	131.9	116.0(12.03%)	73.5(44.28%)	69.5(47.29%)
	Q1&Ds	128.1	114.8(10.40%)	65.5(48.91%)	69.4(45.83%)
	平均	**130.0**	**115.4(11.23%)**	**69.5(46.56%)**	**69.5(46.57%)**
基模型的性能优于协作模型	M3&Q1	130.3	133.6(−2.52%)	78.7(39.61%)	93.8(28.02%)
	M3&Ds	161.7	133.1(17.69%)	79.7(50.70%)	95.2(41.13%)
	L3&M3	131.0	158.6(−21.04%)	81.3(37.96%)	99.7(23.91%)
	L3&Ds	130.3	107.7(17.34%)	74.3(42.99%)	83.2(36.14%)
	L3&Q1	130.6	111.1(14.96%)	78.7(39.78%)	82.1(37.16%)
	Hy&Q1	153.1	142.5(6.92%)	80.1(47.68%)	91.1(40.50%)
	Hy&Ds	148.5	142.3(4.18%)	79.9(46.20%)	88.9(40.13%)
	Hy&M3	162.8	147.0(9.71%)	100.1(38.51%)	84.1(48.34%)
	平均	**143.5**	**134.5(6.31%)**	**81.6(43.16%)**	**89.8(37.47%)**
基模型的性能弱于协作模型	Ds&M3	161.7	133.0(17.75%)	82.0(49.29%)	85.6(47.07%)
	Q1&M3	130.3	133.6(−2.52%)	73.6(43.56%)	87.3(33.01%)
	M3&L3	131.0	158.6(−21.04%)	75.2(42.64%)	84.6(35.44%)
	Ds&L3	130.3	107.7(17.34%)	71.5(45.14%)	71.9(44.82%)
	Q1&L3	130.6	111.1(14.96%)	68.8(47.31%)	76.3(41.60%)
	Q1&Hy	153.1	142.5(6.92%)	87.9(42.59%)	81.0(47.09%)
	Ds&Hy	148.5	118.4(20.27%)	86.4(41.82%)	77.6(47.74%)
	M3&Hy	162.8	138.7(14.80%)	80.2(50.74%)	87.4(46.31%)
	平均	**143.5**	**130.5(9.13%)**	**78.2(45.53%)**	**81.5(43.25%)**

注:加粗表示各组实验最优结果。

FuseGate后,并联协作方法并未因协作触发频率下降而产生显著性能损失。在多数模型组合中,选择性协作仍能够维持相对稳定的推理准确率,表明模型协作带来的有效增益集中于部分解码位置,而非均匀分布于整个生成过程。FuseGate通过置信度门控机制对协作时机进行筛选,在维持推理准确率的同时降低了单词元生成平均时延。

综上,在本文考察的模型组合和任务设置下,FuseGate在多数配置中都能在减少协作模型调用的同时,维持与全程协作相近的推理准确率,在推理准确率和单词元生成平均时延之间取得了更好的折中。

3.5 消融实验

本节在模型组合“Q1&Ds”下,基于DuetNet并联协作框架对FuseGate开展消融实验。为保证结果具有可比性,除被消融对象外,其余实验设置保持不变,测试数据集和提示词模板均参照第3.1节的设置。

3.5.1 滑动窗口机制消融

为验证滑动窗口机制在FuseGate中的作用,本小节对不同门控方式进行了消融实验。在并联协作框架DuetNet下,设置了两种协作触发方式:

(1)DuetNet+StaticGate:静态阈值门控,即采用预设阈值τ直接判断当前解码步是否触发协作。

(2)DuetNet+FuseGate:滑动窗口机制门控,即基于最近若干解码步的Margin历史分布动态确定当前步的触发阈值,并据此自适应决定是否触发协作。

表6结果表明,静态阈值作为门控依据时稳定性不足。不同阈值设置会导致明显的性能波动,且阈值提高后,准确率并没有表现出单调提升的趋势。与之相比,在协作比例接近的情况下,FuseGate取得了更好的推理结果。这表明滑动窗口机制能够为当前解码步提供动态参照,更准确地筛选出需要调用协作模型的词元。

表6 滑动窗口机制消融实验结果

Table 6 Ablation results of the sliding-window mechanism

方法	门控设置	准确率/%	协作率/%
DuetNet	全程协作	64.2	100.0
DuetNet+StaticGate	$\tau=2$	64.4	23.2
DuetNet+StaticGate	$\tau=4$	65.3	39.7
DuetNet+StaticGate	$\tau=6$	66.7	51.0
DuetNet+StaticGate	$\tau=8$	65.9	60.3
DuetNet+FuseGate	滑动窗口	**68.7**	**49.67**

注:加粗表示各组实验最优结果。

3.5.2 门控策略消融

本小节通过门控策略消融实验，分析协作位置选择对模型性能的影响，验证FuseGate是否能够更有效地筛选需要协作的词元位置。实验设置如下：

（1）DuetNet：全程协作，在每个解码步均触发协作。

（2）DuetNet+RandGate：随机门控，即在与FuseGate相近的平均协作率下随机触发协作，进行3次实验取平均值。

（3）DuetNet+FuseGate：动态门控，即依据当前解码步的置信度动态决定是否触发协作。

实验结果如表7所示。可以看出，全程协作虽然在每一个解码步中都引入协作模型参与解码，但并没有取得最优性能，这说明协作并不是越频繁就越有效。相比之下，RandGate能够降低协作率。但是，由于其触发位置由随机策略决定，难以覆盖真正高风险的解码步，因此带来的性能提升相对有限。FuseGate在与RandGate协作率相近的情况下，取得了更高的准确率，说明其优势并不来源于协作次数本身，而在于能够更准确地选择协作时机。

表7 门控策略消融实验结果

Table 7 Ablation results of different gating strategies

方法	门控策略	准确率/%	协作率/%
DuetNet	全程协作	64.2	100
DuetNet+RandGate	随机门控	61.8	50.2
DuetNet+FuseGate	动态门控	**68.7**	**49.67**

注：加粗表示各组实验最优结果。

3.5.3 置信度信号消融

为验证Margin作为置信度信号的有效性，本小节在保持并联协作框架和门控形式不变的条件下，仅替换协作触发所依据的置信度特征，并比较协同体在不同置信度信号下的推理性能。本小节以DuetNet作为统一的并联协作框架，设置以下3种触发方式：

（1）DuetNet+FuseGate：采用本文提出的FuseGate机制，以基模型当前解码步Top-1与Top-2的logits之差（Margin）作为置信度信号，当置信度较低时触发协作。

（2）DuetNet+GaCGate：采用基于Top-1置信度的阈值门控策略，以基模型当前解码步的Top-1 logits值作为触发依据，当其低于阈值时触发协作。

（3）DuetNet+RandGate：随机选择与前两种方法相同数量的词元位置触发协作。重复3次实验取平均值，以消除随机波动带来的影响。

分位点参数q分别设置为0.3、0.5和0.7。表8展示了不同分位点参数下3种门控策略的准确率对比结果。

从表8可以看出，在3种不同分位点参数设置下，DuetNet+FuseGate均取得了最高的推理准确率。相较而言，GaCGate和RandGate的实验结果波动较为明显，整体准确率也低于FuseGate。该结果表明，GaCGate在刻画置信度变化时稳定性不足，而RandGate缺乏明确的置信度选择依据，因此这两种方法难以达到与FuseGate相近的实验效果。该实验结果也从侧面验证了将Margin作为门控信号的合理性。

表8 置信度信号消融实验结果

Table 8 Ablation results of different confidence signals

方法	$q=0.3$	$q=0.5$	$q=0.7$
DuetNet+RandGate	62.2	61.4	62.5
DuetNet+GaCGate	66.0	66.1	65.2
DuetNet+FuseGate	**67.5**	**68.7**	**67.7**

注：加粗表示各组实验最优结果。

3.6 推理结果分布分析

为分析FuseGate相比其他门控策略带来更高准确率的直观原因，本节对推理结果的模式分布进行分析，结果如图4所示。图中“√√×”表示两个模型独立推理的结果均正确，但协作后错误；“√×√”表示两个模型独立推理中一个正确、一个错误，而协作后推理正确，其余情况类推。

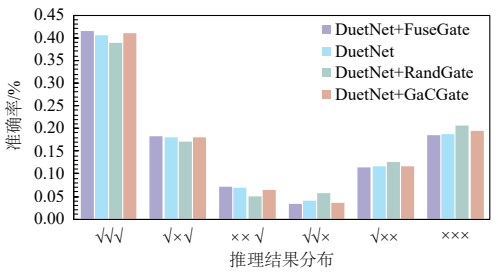

图4 不同门控策略下的推理结果模式

Figure 4 Inference result patterns under different gating strategies

观察图4的结果，可以发现：

当两个单模型均推理正确时，对应的“√√√”占比均处于较高水平，而“√√×”占比相对较低。当两个单模型的推理结果为一错一对时，不同门控策略在结果分布上出现可观察差异：相较于RandGate与GaCGate，FuseGate更倾向于产生协作正确的结果模式，并相对减少协作错误的结果模式。当两个单模型均推理错误时，整体上所有方法的推理结果仍倾向于错误，但仍存在一定比例的“××√”，说明词元级双模型并联具有一定纠正单模型错误的能力。在相同的组合条件下，DuetNet+FuseGate相比DuetNet与其他门控基线呈现更有利的分布结构，其“×××”占比相对更低而“××√”占比相对更高。由此可见，FuseGate维持推理准确率

的主要原因在于其对协作位置的选择更加准确。

4 结束语

本文围绕模型互联网场景下词元级并联协作推理的效率优化问题展开研究。针对现有逐词元同步协作方法在提升生成质量的同时,引入冗余计算与通信同步开销的问题,提出了一种基于置信度门控的选择性协作机制FuseGate。该机制通过面向分位点参数约束的滑动窗口策略动态选择词元序列,并依据词元间logits差异刻画词元置信度,实现选择性触发并联协作,减少词元级并联协作的冗余计算与通信开销。实验结果表明,在多组异构模型组合和多类基准任务上,将FuseGate接入现有词元级并联协作框架后,能够在降低单词元生成平均时延的同时,尽可能维持与全程协作方法相近的推理准确率。本文工作为模型互联网场景下异构模型的高效协同推理提供了一种轻量化、可插拔的优化思路,也为后续面向动态预算分配和更细粒度协作控制的模型协作研究奠定了基础。

本文提出的FuseGate机制能够有效降低词元级并联协作的单词元生成平均时延。在此基础上,后续研究可从以下两个方面进一步拓展:首先,当前机制主要判断当前解码步是否需要触发并联协作,而协作模型的选择仍有进一步研究价值。未来可结合模型能力、任务类型和推理开销等因素,探索词元级协作中的动态模型选择机制;其次,FuseGate在触发协作后直接采用并联融合结果,后续可进一步引入协作收益评估机制,判断并联协作输出相较于基模型输出是否更可靠,从而减少低质量协作信号对词元生成的影响。

参考文献

[1] 李哲涛, 曾曦玉, 王建辉, 等. 模型互联网: 概念、现状和未来[J]. 计算机研究与发展, 2026, 63(5): 1305-1318. DOI: 10.7544/issn1000-1239.202550223.
Li Zhetao, Zeng Xiyu, Wang Jianhui, et al. AI-model network: Concept, current state and future[J]. Journal of Computer Research and Development, 2026, 63(5): 1305-1318. DOI: 10.7544/issn1000-1239.202550223.(in Chinese)

[2] Bommasani R, Liang P, Lee T. Holistic evaluation of language models[J]. Annals of the New York Academy of Sciences, 2023, 1525(1): 140-146. DOI: 10.1111/nyas.15007.

[3] Yin Shukang, Fu Chaoyou, Zhao Sirui, et al. Woodpecker: Hallucination correction for multimodal large language models[J]. Science China Information Sciences, 2024, 67(12): 220105. DOI: 10.1007/s11432-024-4251-x.

[4] Chiang W L, Zheng Lianmin, Sheng Ying, et al. Chatbot arena: An open platform for evaluating LLMs by human preference[C/OL]//Proceedings of the 41st International Conference on Machine Learning, 2024: 8359-8388. https://openreview.net/forum?id=3MW8GKNyzI.

[5] Liu Xiao, Yu Hao, Zhang Hanchen, et al. AgentBench: Evaluating LLMs as agents[C/OL]//Proceedings of the Twelfth International Conference on Learning Representations, 2024. https://openreview.net/forum?id=zAdUB0aCTQ.

[6] 王建辉, 李哲涛, 伍涛, 等. Token级多模型并联协作推理[J]. 计算机学报, 2025, 48(11): 2579-2593.
Wang Jianhui, Li Zhetao, Wu Tao, et al. Token-level collaborative reasoning for parallel multi-models[J]. Chinese Journal of Computers, 2025, 48(11): 2579-2593. (in Chinese)

[7] Wang Junlin, Wang Jue, Athiwaratkun B, et al. Mixture-of-agents enhances large language model capabilities[C/OL]//Proceedings of the 13th International Conference on Learning Representations, 2025. https://openreview.net/forum?id=h0ZfDIrj7T.

[8] Sun Qiushi, Yin Zhangyue, Li Xiang, et al. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration[C/OL]//Proceedings of Conference on Language Modeling, 2024. https://openreview.net/forum?id=GDdxmymrwL.

[9] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: Curran Associates Inc., 2017: 6000-6010.

[10] Cobbe K, Kosaraju V, Bavarian M, et al. Training verifiers to solve math word problems[PP/OL]. V2. arXiv (2021-11-18)[2026-07-12]. https://arxiv.org/abs/2110.14168.

[11] Chen Lingjiao, Zaharia M, Zou J. FrugalGPT: How to use large language models while reducing cost and improving performance[PP/OL]. V1. arXiv (2023-05-09) [2026-07-12]. https://arxiv.org/abs/2305.05176.

[12] Lu Keming, Yuan Hongyi, Lin Runji, et al. Routing to the expert: Efficient reward-guided ensemble of large language models[C]//Proceedings of 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New York: Association for Computational Linguistics, 2024: 1964-1974. DOI: 10.18653/v1/2024.naacl-long.109.

[13] Pan Jiabao, Zhang Yan, Zhang Chen, et al. DynaThink: Fast or slow? A dynamic decision-making framework for

large language models[C]//Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. New York: Association for Computational Linguistics, 2024: 14686-14695. DOI: 10.18653/v1/2024.emnlp-main.814.

[14] 刘忠仁, 李哲涛, 王建辉, 等. 模型互联中多模型串并联协作推理[J]. 电子学报, 2025, 53(11): 3817-3835. DOI: 10.12263/DZXB.20250503.
Liu Zhongren, Li Zhetao, Wang Jianhui, et al. Multi-model serial and parallel collaborative inference in AI-ModelNet[J]. Acta Electronica Sinica, 2025, 53(11): 3817-3835. DOI: 10.12263/DZXB.20250503.(in Chinese)

[15] Mohammadshahi A, Shaikh A R, Yazdani M. Routoo: Learning to route to large language models effectively[PP/OL]. V3. arXiv (2024-10-02)[2026-07-12]. https://arxiv.org/abs/2401.13979.

[16] Yu Y C, Kuo C C, Ye Ziqi, et al. Breaking the ceiling of the LLM community by treating token generation as a classification for ensembling[C]//Findings of the Association for Computational Linguistics: EMNLP 2024. New York: Association for Computational Linguistics, 2024: 1826-1839. DOI: 10.18653/v1/2024.findings-emnlp.99.

[17] Yao Yuxuan, Wu Han, Liu Mingyang, et al. Determine-then-ensemble: Necessity of top-k union for large language model ensembling[C/OL]//Proceedings of the 13th International Conference on Learning Representations, 2025. https://openreview.net/forum?id=FDnZFpHmU4.

[18] Hao Chao, Wang Zezheng, Huang Yanhua, et al. Token-by-token election: Improving language model reasoning through token-level multi-model collaboration[EB/OL]. (2024-09-13)[2026-07-12]. https://openreview.net/forum?id=QPZy2XMgzn.

[19] Xu Yangyifan, Ren Shuo, Zhang Jiajun. Collaborative beam search: Enhancing LLM reasoning via collective consensus[C]//Proceedings of 2025 Conference on Empirical Methods in Natural Language Processing. New York: Association for Computational Linguistics, 2025: 11398-11410. DOI: 10.18653/v1/2025.emnlp-main.574.

[20] Huang Yichong, Feng Xiaocheng, Li Baohang, et al. Ensemble learning for heterogeneous large language models with deep parallel collaboration[C]//Proceedings of the 38th International Conference on Neural Information Processing Systems. New York: Curran Associates Inc., 2024: 3808. DOI: 10.52202/079017-3808.

[21] Bi Xiao, Chen Deli, Chen Guanting, et al. DeepSeek LLM: Scaling open-source language models with longtermism[PP/OL]. V1. arXiv (2024-01-05)[2026-07-12]. https://arxiv.org/abs/2401.02954.

[22] QWEN Team. Introducing Qwen1.5[EB/OL]. (2024-02-04)[2026-07-12]. https://qwenlm.github.io/blog/qwen1.5/.

[23] Mistral AI Team. Mistral 7B[EB/OL]. (2023-09-27)[2026-07-12]. https://mistral.ai/news/announcing-mistral-7b/.

[24] Grattafiori A, Dubey A, Jauhri A, et al. The llama 3 herd of models[PP/OL]. V3. arXiv (2024-11-23)[2026-07-12]. https://arxiv.org/abs/2407.21783.

[25] Tencent. Hunyuan-7B-Instruct[EB/OL]. (2025-07-30)[2026-07-12]. https://huggingface.co/tencent/Hunyuan-7B-Instruct/.

[26] Du Yilun, Li Shuang, Torralba A, et al. Improving factuality and reasoning in language models through multiagent debate[C/OL]//Proceedings of the 41st International Conference on Machine Learning, 2024: 467. https://openreview. net/challenge? redirect= % 2Fpdf% 3Fid% 3Dzj7YuTE4t8.

[27] Clark C, Lee K, Chang Mingwei, et al. BoolQ: Exploring the surprising difficulty of natural yes/no questions[C]//Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New York: Association for Computational Linguistics, 2019: 2924-2936. DOI: 10.18653/v1/N19-1300.

[28] Hendrycks D, Burns C, Basart S, et al. Measuring massive multitask language understanding[C/OL]//Proceedings of the 9th International Conference on Learning Representations, 2021. https://openreview. net/forum? id=d7KBjmI3GmQ.

作者简介

田淑娟 女,1982年6月出生于湖南省郴州市。现为湘潭大学计算机学院教授、博士生导师。主要研究方向为模型互联网、网络安全和边缘计算。

E-mail: sjtianwork@xtu.edu.cn

李艳春 女,1981年2月出生于湖南省涟源市。现为湘潭大学计算机学院副教授、硕士生导师。主要研究方向为图像处理、计算机视觉和深度模型安全。

E-mail: ycli@xtu.edu.cn

罗宇航 男,2003年3月出生于湖南省岳阳市。现为湘潭大学计算机学院硕士研究生。主要研究方向为模型互联网、智能体和大小模型协同。

E-mail: 18075120223@163.com

代星霞 女,1996年2月出生于湖南省长沙市。现为湘潭大学计算机学院讲师、硕士生导师。主要研究方向为边缘计算、大小模型协同和智能优化。

E-mail: xingxdai@xtu.edu.cn

项淑桓 男,2001年9月出生于江西省景德镇市。现为湘潭大学计算机学院硕士研究生。主要研究方向为联邦学习、大模型和人工智能安全。

E-mail: 202531630501@smail.xtu.edu.cn

申冬苏 女,1978年11月出生于湖南省邵阳市。现为湘潭大学计算机学院副教授、硕士生导师。主要研究方向为网络安全、密码学和人工智能安全。

E-mail: dsshen@xtu.edu.cn