

思考面具内：扩散大型语言模型中的原位提示

Xiangqi Jin¹, Yuxuan Wang², Yifeng Gao^{1,3}, Zichen Wen^{1,3},
Biqing Qi³, Dongrui Liu³, Linfeng Zhang^{1*}

¹School of Artificial Intelligence, Shanghai Jiao Tong University

²Xidian University

³Shanghai AI Laboratory

Abstract

尽管大语言模型 (LLMs) 在许多任务中取得了显著的成功，它们的仅前缀提示范式和顺序生成过程对双向信息提供了有限的灵活性。扩散大语言模型 (dLLMs) 通过其双向注意机制和迭代优化过程提供了新的机会，使得能进行更灵活的就地提示策略。我们介绍了一种新框架 ICE (I - 位思维链提示与 E 早退出)，专门为 dLLMs 设计，将仅前缀提示转化为就地提示。ICE 在迭代优化期间将就地提示直接集成到掩码标记位置中，并采用一个自信感知的早退出机制，以显著减少计算开销。大量实验表明，ICE 的有效性，在 GSM8K 上实现了高达 17.29 % 的准确性提高和 $4.12 \times$ 的加速，并在 MMLU 上实现了高达 $276.67 \times$ 的加速，同时保持了竞争性的性能。我们的代码将在 Github 上发布。

介绍

大型语言模型 (LLMs) (Zhao et al. 2023) 革命性地改变了自然语言处理，其中自回归 (AR) 模型通过其顺序的、从左到右的词元生成范式主导了这一领域 (Brown et al. 2020; Touvron et al. 2023)。AR 模型天生受到仅前缀提示和顺序生成的限制。扩散大型语言模型 (dLLMs) (Ye et al. 2025; Nie et al. 2025; Zhu et al. 2025; Yang et al. 2025) 通过迭代的掩码词元优化 (Austin et al. 2021a; Lou, Meng, and Ermon 2023) 提供了非自回归的替代方案。至关重要的是，dLLMs 通过其双向注意力机制和迭代优化过程呈现了新的机会，能够更灵活地实现现场提示策略，可以将信息直接嵌入到掩码词元位置中 (Wen et al. 2025)。LLaDA 达到了 GPT-3.5 的性能 (Nie et al. 2025)，而 Mercury 实现了每秒 1000+ 个词元 (Liu et al. 2025a)，但 dLLMs 的推理能力仍未被充分探索。

虽然链式思维 (CoT) 提示通过将复杂问题分解为中间推理步骤来证明对自回归 (AR) 模型非常有效 (Wei et al. 2022)，但 dLLM 的双向和迭代特性使得从根本上可以采用不同的方法。与将推理视为顺序前缀条件的 AR 模型不同，dLLM 可以在生成过程中直接嵌入推理，并采用就地提示 (图 1)。此外，AR 模型表现出顺序答案显现，答案在生成顺序完成之前是不可访问的，而 dLLM 通过其双向上下文建模使得答案并发可访问，允许在迭代优化过程中中间可见答案内容。这种架构上的区别为新的信心感知优化策略创造了机会，能够在生成过程中监控答案。

*Corresponding author (zhanglinfeng@sjtu.edu.cn).

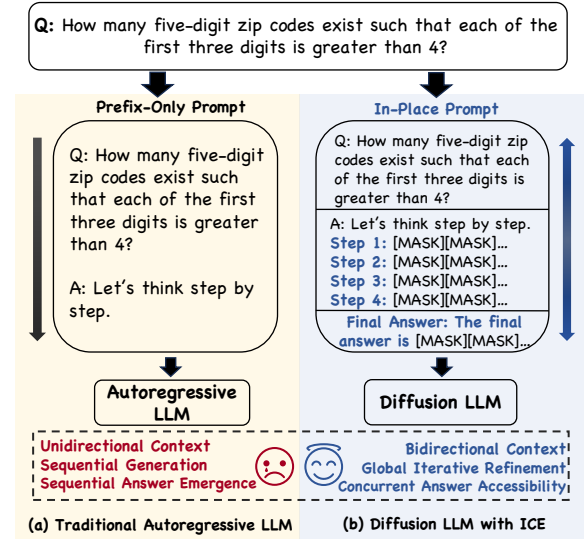


Figure 1: 在 (a) 自回归大型语言模型与 (b) 扩散大型语言模型中的提示构建。自回归大型语言模型采用单向上下文和仅前缀提示，而 dLLMs 利用双向上下文建模，使得就地提示和并行答案获取成为可能。

为了解决这些机会，我们提出了 ICE (I n-Place C hain-of-Thought Prompting with E arly Exit)，这是一种增强 dLLMs 推理能力和推理效率的新框架 (图 3)。我们的核心见解是，dLLMs 的迭代生成过程提供了一个独特的机会，可以在生成过程中直接嵌入推理步骤，将推理从外部预处理转变为生成机制的一个整体组件。ICE 引入了两个关键创新：

就地链式思维提示：这种方法在迭代优化过程中直接将推理步骤整合到被掩盖的标记位置中。通过将生成序列构建为思维和答案的不同部分，并在思维部分嵌入明确的逐步推理模板，它利用了双向 LLMs 的特性。这在保持并行生成优势的同时，能够增强推理性能。

两阶段解码与提前退出机制：受到一个重要的经验观察启发，我们设计了一种信心感知推理策略，该策略利用推理和答案部分之间不同的细化模式。通过系统分析迭代细化的动态，我们发现了 dLLM 的一个独特行为模式：模型对答案标记的信心迅速收敛到较高水平并保持稳定，而推理部分则继续进行细化 (图 2)。这一观察

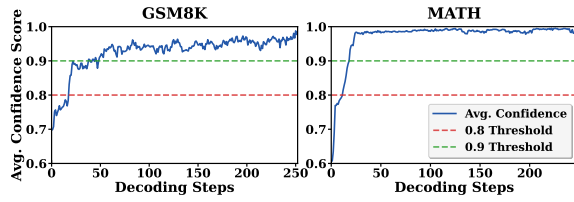


Figure 2: 在生成过程中，GSM8K 和 MATH 测试中的答案部分的平均置信度。模型在答案部分的置信度迅速收敛至高水平，并在后续迭代中保持稳定，表明模型在过程初期内部确定了正确答案，同时继续完善推理轨迹。

揭示了模型通常在明确的推理痕迹完成之前就已经在内部确定了正确答案。基于这一见解，ICE 实现了一种两阶段的解码方法，允许所有答案标记的并行解码，同时有效减少冗余计算。

广泛的实验验证证明了 ICE 在多样化推理基准测试中的有效性。在数学推理任务中，ICE 在 GSM8K 上实现了最高达 17.29 % 的准确率提升，并且加速了 $4.12 \times$ ，同时在 MATH 上也获得了持续的提升。对于知识密集型任务，ICE 在 MMLU 上提供了高达 $276.67 \times$ 的加速，同时在 GPQA 上实现了超过 $40 \times$ 的加速，同时保持了具有竞争力的准确性。值得注意的是，ICE 与现有的加速技术兼容，如 dLLM-Cache (Liu et al. 2025c)，在结合使用时可实现累积效益。

我们的贡献有三个方面：

- 我们提出了第一个用于动态大语言模型 (dLLMs) 的原地提示框架，将提示直接嵌入到被掩盖的标记中，以提高准确性和效率。
- 我们开发了一种具有提前退出机制的两阶段解码策略，这种策略显著降低了推理延迟，同时保持了生成质量。
- 我们提供了详尽的实证证据，证明 ICE 的有效性，并确定推理模式与生成机制之间的架构对齐可以产生协同效益。

相关工作

扩散大型语言模型。最近的扩散大型语言模型 (dLLMs) (Nie et al. 2025; Austin et al. 2021b; Lou, Meng, and Ermon 2023; Shi et al. 2024; Liu et al. 2025b)，特别是 LLaDA，代表了一种从自回归生成转向的一种范式转变，其采用双向注意力机制和迭代的优化过程。与从左到右顺序生成标记的自回归模型不同，dLLMs 利用有完整序列上下文意识的掩码标记预测，使它们能够克服基本的限制，比如逆转咒语 (Berglund et al. 2023)。LLaDA 是一个从头开始训练的拥有 80 亿参数的模型，与 LLaMA3 8B 的性能相媲美。这种掩码扩散过程本质上通过其非自回归的架构支持上下文学习和指令跟随，开辟了序列中直接嵌入结构化推理的独特机会，而不是仅依赖于自回归模型中的前缀提示。

连锁思维推理。连锁思维 (CoT) 提示 (Wei et al. 2022; Nakamura and et al. 2021) 的引入通过促进系统的逐步问题分解，显著提高了大型语言模型 (LLMs) 的推理准确性。然而，传统的 CoT 方法固有其对自回归生成和提示级指导的依赖的约束，这限制了它们与基

于扩散的架构的有效整合。最近的进展主要针对自回归模型 (Kojima et al. 2022; Gao and et al. 2023)，因此未能利用与扩散型大型语言模型 (dLLMs) 固有的双向上下文建模能力的独特结合。为解决这一差距，我们引入了原地 CoT 提示，将推理步骤直接嵌入到序列中进行迭代优化，从而在生成过程中实现细粒度的控制。dLLMs 的高效推理。扩散 LLMs 由于其迭代生成过程固有地面临高计算开销。当前对 dLLMs 的加速方法主要集中在算法层面的计算优化策略 (Wu et al. 2025; Ma et al. 2025; Hu et al. 2025; Luxembourg, Permuter, and Nachmani 2025)。dLLM-Cache (Liu et al. 2025c) 通过一个无需训练的自适应缓存框架解决这一挑战，该框架会根据策略重复利用稳定的提示计算，并采用相似性引导的部分响应更新。SlowFast Sampling (Wei et al. 2025) 通过解决不确定性的探索性‘慢’阶段和根据确定性、收敛性和位置原则进行自信生成的激进‘快’阶段之间动态交替来进一步提高推理速度。与这些算法方法不同，我们的方法引入了一个内容感知的加速框架，该框架利用 dLLMs 特有的双向性进行基于置信度的提前退出。

方法论

预备知识

掩码扩散大型语言模型。掩码扩散大型语言模型 (dLLMs) 实现一个前向过程，通过引入一个特殊的 [MASK] 标记，逐步破坏输入序列 x_0 。这个过程由一个连续的时间参数 $t \in [0, 1]$ 控制。在每个时间步 t ，生成的序列 x_t 被部分掩码化，每个标记独立地以 $1 - \alpha_t$ 的概率被替换为 [MASK]，或者以 α_t 的概率保留。噪声计划 α_t 单调递减于 t ，决定了损坏率。在 $t = 1$ 时，序列 x_1 完全被掩盖，仅由 [MASK] 标记组成。

训练一个 masked dLLM 涉及到一个双向预测器 f_θ ，该预测器从其损坏的对应序列 x_t 重建原始序列 x_0 。在每次训练步骤中，均匀地采样一个时间步 $t \in [0, 1]$ ，并根据 α_t 定义的前向过程对标记进行掩码。目标是 minimized 负证据下界 (NELBO)，这是数据负对数似然的上界。对于 masked dLLMs，NELBO 简化为加权对数似然损失，其权重源自 α_t (Sahoo et al. 2024)。流行的 LLaDA 模型使用线性噪声调度 $\alpha_t = 1 - t$ ，其中产生的 NELBO 为：

$$-\mathbb{E} \left[\frac{1}{t} \sum_{k=1}^{|x_t|} \mathbb{I}[x_t^k = [\text{MASK}]] \log f_\theta(x_0^k | x_t) \right], \quad (1)$$

，其中 $t \sim \mathcal{U}[0, 1]$ ， $x_0 \sim p_{\text{data}}$ ， $x_t \sim q_{t|0}(x_t | x_0)$ ， $|x_t|$ 表示序列长度， x_t^k 表示 x_t 的第 k 个标记，而 $\mathbb{I}[\cdot]$ 是一个指示函数，将损失限制在被掩码的标记上。与依赖于静态掩码比例和单步标记预测的 BERT (Devlin 2018) 相反，masked dLLMs 采用动态掩码概率并支持从完全掩码状态的迭代解码，从而实现生成建模。

掩码 dLLMs 的推理过程。推理过程通过迭代细化逆转前向腐蚀，逐步将完全掩码的序列转换为连贯的输出。给定一个推理序列 $y = (y_{\text{prompt}}, y_{\text{gen}})$ ，其中 y_{prompt} 表示初始提示， y_{gen} 表示要生成的序列，模型在跨越 N 个离散细化步骤中保持一个中间状态 $y^{(k)} = (y_{\text{prompt}}, y_{\text{gen}}^{(k)}) \in \mathcal{T}^L$ ，从 $k = N$ 进展到 $k = 0$ 。其中， $\mathcal{T}$ 表示标记词汇表， L 表示总序列长度。该过

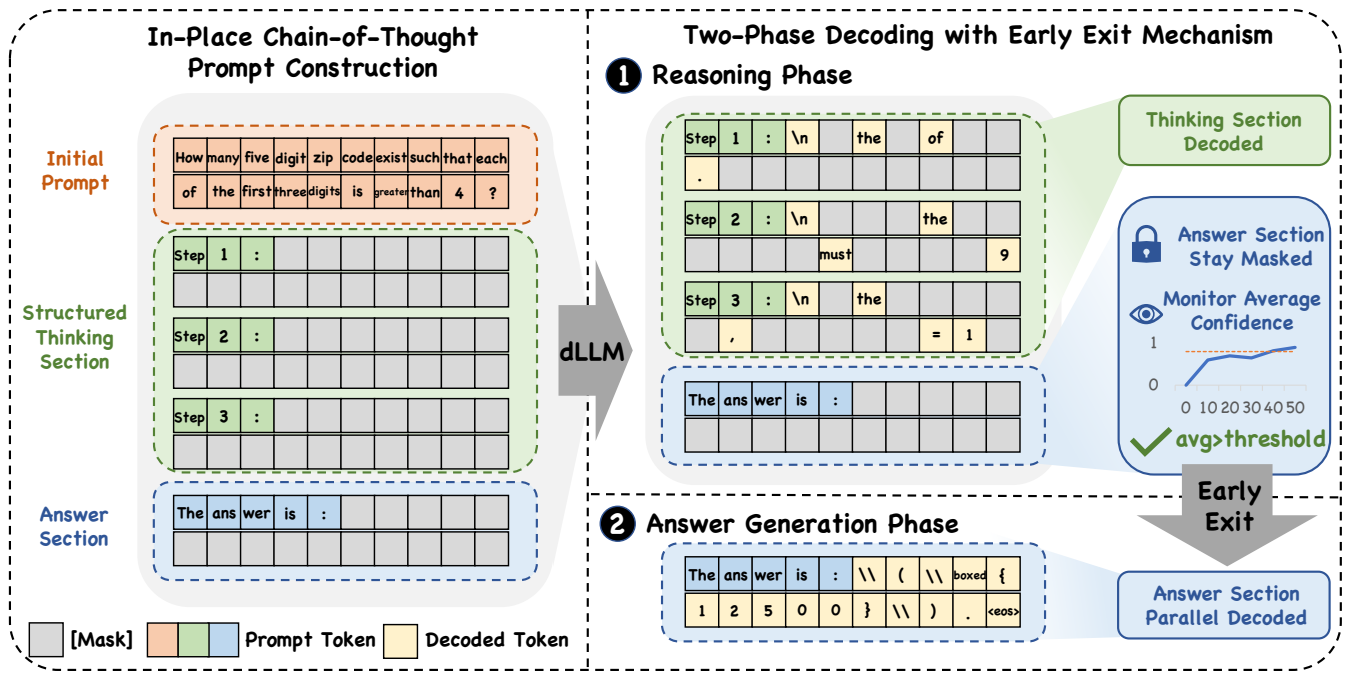


Figure 3: ICE 框架概述。ICE 集成了两个关键组成部分：(1) 原地思维链提示法，将结构化的逐步推理模板直接嵌入到提示中，以及 (2) 具有信心感知提前退出机制的两阶段解码。在推理阶段，模型通过监测被掩盖答案部分的平均信心来迭代解码思维部分。当达到信心阈值时，框架转换到答案生成阶段，并行解码所有答案标记以生成最终响应。

程以一个完全掩码的生成序列初始化：

$$\mathbf{y}^{(N)} = (\mathbf{y}_{\text{prompt}}, \underbrace{[\text{MASK}], \dots, [\text{MASK}]}_{L_{\text{gen}} \text{ times}}) \quad (2)$$

在每一步 $k \in \{N, N-1, \dots, 1\}$ ，双向预测器 f_{θ} 从当前噪声状态 $\mathbf{y}^{(k)}$ 估计原始序列 $\mathbf{y}_0$ ：

$$f_{\theta}(\mathbf{y}_0 | \mathbf{y}^{(k)}) \quad (3)$$

模型在步骤 k 通过贪心解码获得干净序列的估计，记为 $\hat{\mathbf{y}}_0^{(k)}$ ：

$$\hat{\mathbf{y}}_{0,i}^{(k)} = \arg \max_{v \in \mathcal{V}} f_{\theta}(\mathbf{y}_{0,i} = v | \mathbf{y}^{(k)}) \quad \forall i \in \{1, \dots, L\} \quad (4)$$

随后，一个转移函数 S 通过基于当前估计 $\hat{\mathbf{y}}_0^{(k)}$ 选择性地更新 $\mathbf{y}^{(k)}$ 中的标记来生成下一状态 $\mathbf{y}^{(k-1)}$ ：

$$\mathbf{y}^{(k-1)} = S(\hat{\mathbf{y}}_0^{(k)}, \mathbf{y}^{(k)}, k) \quad (5)$$

S 的实现通常采用诸如基于置信度的解掩或随机解掩等策略。

就地链式思维提示

dLLM 的迭代、非自回归生成范式，加上其固有的双向注意机制，使其能从自回归模型采用的传统仅前缀提示策略中根本性地脱离出来。尽管自回归模型受到顺序的、从左到右的生成过程的限制，dLLM 具备同时考虑整个序列上下文并实现并发答案可访问性的能力。由于能够在生成过程中始终意识到答案区域，同时对推理步骤进行同步优化，这种架构优势解锁了新的提示范式。

我们的方法通过将生成序列 $\mathbf{y}_{\text{gen}}$ 结构化为两个语义上不同的部分来利用这种独特的能力：一个思考部分 $\mathbf{y}_{\text{thinking}}$ 和一个答案部分 $\mathbf{y}_{\text{answer}}$ 。这种结构划分是由 dLLM 的双向特性独特实现的：与自回归模型不同，在自回归模型中，推理必须在任何答案内容可用之前顺序生成，而 dLLMs 可以在迭代优化过程中同时考虑推理和答案上下文。形式上，我们定义生成序列为：

$$\mathbf{y}_{\text{gen}} = (\mathbf{y}_{\text{thinking}}, \mathbf{y}_{\text{answer}}) \quad (6)$$

这导致完整的输入序列被结构化为：

$$\mathbf{y} = (\mathbf{y}_{\text{prompt}}, \mathbf{y}_{\text{thinking}}, \mathbf{y}_{\text{answer}}) \quad (7)$$

为了在这些功能部分之间建立清晰的界限，我们引入了一个任务特定的答案指示器，该指示器位于思考部分和答案部分之间。这个指示器作为一个明确的信号，使模型从推理详述过渡到最终回答的制定。

在此基础上，我们通过将思维部分 $\mathbf{y}_{\text{thinking}}$ 分解为多个明确的推理步骤 $T = (T_1, T_2, \dots, T_{N_t})$ 来进一步引导模型进行系统化的推理分解。这是通过在被掩盖的词汇中嵌入明确的逐步推理模板来实现的。思维部分被初始化为：

$$\mathbf{y}_{\text{thinking}}^{(N)} = (\underbrace{T_1, T_2, \dots, T_{N_t}}_{N_t \text{ reasoning steps}}) \quad (8)$$

将结构推理线索直接嵌入生成空间的方法有效地引导 dLLM 在形成答案之前产生明确、可追溯的推理序列，从而提高推理过程的透明度和准确性。重要的是，这种就地的方法利用了 dLLM 的并发答案可访问性，使得模型能够在整个迭代优化过程中保持对推理发展和答案形成的整体感知。

两阶段解码与提前退出机制

虽然我们的方法中嵌入式的链式思考提示显著增强了推理能力，但 dLLM 推断的迭代特性引入了相当大的计算负担。为了解决这一挑战，我们利用 dLLM 的并发答案可访问性进行置信优化。我们观察到在迭代细化过程中置信动态的一个关键模式：模型在答案标记上的置信度快速收敛到高水平，并在随后的迭代中保持显著稳定性，而思考部分则持续进行细化（图 2）。这种不对称的收敛模式表明，在早期置信度稳定化之后继续细化会在答案质量上产生收益递减，同时导致不必要的计算成本。

这一现象表明，模型往往在显式推理步骤完成之前就已经在内部大致收敛到正确答案。在这一信心收敛点之后继续进行迭代细化，会在答案质量上带来收益递减，同时会产生不必要的计算成本，因为后续的迭代主要是用于阐述和细化推理步骤，而不是提高最终答案的准确性。

利用这一洞察，我们引入了一种高效的推理策略，包括一个具有置信度基础提前退出机制的两阶段解码过程。核心创新在于使用置信度阈值 τ 来动态控制阶段之间的转换，使得能够根据模型的内部状态进行自适应计算。

基于置信度的阶段切换。在整个解码过程中，我们持续监控被遮蔽的答案标记的平均置信度。我们首先计算每个单独答案标记的置信度得分：

$$\text{confidence}_i^{(k)} = \max_{v \in \mathcal{T}} f_{\theta}(\mathbf{y}_{0,i} = v \mid \mathbf{y}^{(k)}) \quad (9)$$

然后，我们计算所有被遮蔽答案标记的平均置信度：

$$\text{avg_confidence}_{\text{answer}}^{(k)} = \frac{1}{L_{\text{answer}}} \sum_{i \in \text{answer indices}} \text{confidence}_i^{(k)} \quad (10)$$

此置信度阈值 τ 作为阶段转换的决定性标准：当 $\text{avg_confidence}_{\text{answer}}^{(k)} \geq \tau$ 时，我们触发从推理阶段提前退出，并立即过渡到答案生成。

阶段 1：推理阶段。在这个初始阶段，我们专注于生成思维轨迹（ $\mathbf{y}_{\text{thinking}}$ ），同时保持所有答案标记处于其遮蔽状态。转换函数 S （方程 5）仅在 $\mathbf{y}_{\text{thinking}}$ 内解锁标记，指导依据模型的置信度分数。至关重要，我们同时监控答案置信度以进行早期退出检测。当置信度阈值被达到时，该阶段会立刻终止。

第二阶段：答案生成阶段。此阶段仅在 $\text{avg_confidence}_{\text{answer}}^{(k)} \geq \tau$ 时触发。一旦转换，我们执行一个单步解码操作以揭示完整的答案序列 $\mathbf{y}_{\text{answer}}$ 。这种动态的阶段切换消除了冗余计算，同时保持准确性，因为模型已经在最终答案上表现出了足够的信心。在附录中提供了 ICE 框架的详细算法概述。

实验

实验设置

我们在不同的基准测试下评估 ICE：GSM8K (Cobbe et al. 2021) 和 MATH (Hendrycks et al. 2021) 用于数学推理，MMLU (Hendrycks et al. 2020) 和 GPQA (Rein et al. 2023) 用于知识密集型推理。我们使用两种代表性的 dLLMs 进行综合评估：LLaDA-8B-Instruct (Nie et al. 2025) 和 LLaDA-1.5 (Zhu et al. 2025)，以及在

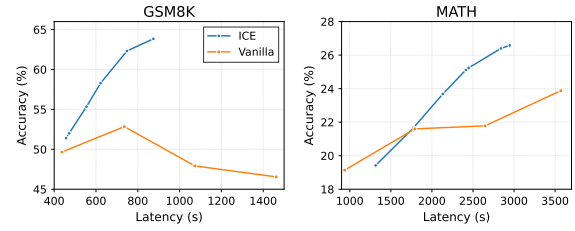


Figure 4: ICE 与原始基准之间的延迟-精度权衡对比。ICE 在 GSM8K 和 MATH 数据集上展示了更优越的帕累托前沿。

8 个 NVIDIA H100 GPU 上的语言模型评估框架 (Gao et al. 2024)。我们将 ICE 与两种基线方法进行比较：(1) Vanilla：标准的自回归生成方法，和 (2) Prefix CoT：传统的作为输入前缀的思维链提示。ICE 提供两种操作模式：ICE-SP（速度优先）用于最大加速和 ICE-PP（性能优先）用于卓越准确性，这两种模式由超参数配置。

我们的主要结果总结在表 1 中。

数学推理任务。对于复杂的数学推理问题，ICE-PP 在 GSM8K 上实现了显著的准确性提升：+17.29 % 和 +13.03 %，提高了速度 $1.67 \times$ 和 $1.52 \times$ ；在 MATH 上实现了 +3.00 % 和 +2.54 % 的提升，同时保持了计算效率。ICE-SP 提供了卓越的效率（在 GSM8K 上为 $4.12 \times$ 和 $3.21 \times$ ，在 MATH 上为 $1.67 \times$ 和 $1.70 \times$ ），几乎没有准确性损失，展示了多样化的优化能力。

知识密集型任务。对于知识密集型推理任务，ICE 同时实现了准确性的提升和显著的效率提升。GPQA 表现出明显的改进，分别提高了 +4.91 % 和 +5.57 %，同时实现了 $19.24 \times$ 和 $42.08 \times$ 的超凡加速，表明在需要深厚领域专业知识任务上具有有效性。MMLU 实现了 +13.10 % 和 +0.74 % 的提升，并且速度提高显著，达到 $133.08 \times$ 和 $276.67 \times$ ，更大的准确性提升对于 LLaDA 指令表明了对于非偏好优化模型的特别益处。这些结果表明，可以通过早期层的表示有效地解决多样的推理查询，而不需要进行明确的速度与准确性的权衡。

延迟-准确性权衡比较。图 4 将 ICE 与通过固定输出长度并调整生成步骤来实现不同权衡点的普通基准进行了比较。ICE 在这两个数据集上均建立了优越的帕累托前沿：在 GSM8K 上将延迟减少了 $2-4 \times$ 的同时保持了准确性，并在 MATH 上实现了更低的延迟和更高的准确性。

与 dLLM-Cache 的兼容性。为了验证 ICE 与现有优化技术的兼容性，我们评估了其 dLLM-Cache 的集成。表 2 展示了 ICE 在与缓存机制结合时能够保持其有效性，既能实现显著的额外加速，又能保持准确性。对于 ICE-SP，缓存加速在 GSM8K 上将加速从 $4.12 \times$ 提升到 $6.10 \times$ ，在 MATH 上则从 $1.67 \times$ 提升到 $2.02 \times$ 。ICE-PP 同样受益于缓存集成，在 GSM8K 上实现了 $2.47 \times$ 的加速，在 MATH 上实现了 $1.33 \times$ 的加速。缓存引入的准确性降低是极小的（通常为 $< 2\%$ ），这证明我们的结构化推理方法在结合补充优化策略时依然具有鲁棒性。

Task	Method	LLaDA-8B-Instruct			LLaDA-1.5		
		Accuracy $\uparrow$	Latency (s) $\downarrow$	Speedup $\uparrow$	Accuracy $\uparrow$	Latency (s) $\downarrow$	Speedup $\uparrow$
GSM8K	Vanilla	46.55	1461.83	1.00 $\times$	45.19	3376.21	1.00 $\times$
	+ Prefix CoT	40.49 -6.06	1443.02 -18.81	1.01 $\times$	34.95 -10.24	3555.10 $+178.89$	0.95 $\times$
	ICE-SP	46.01 -0.54	354.86 -1106.97	4.12 $\times$	45.56 $+0.37$	1050.48 -2325.73	3.21 $\times$
	ICE-PP	63.84 $+17.29$	874.53 -587.30	1.67 $\times$	58.22 $+13.03$	2221.24 -1154.97	1.52 $\times$
MATH	Vanilla	23.88	3570.13	1.00 $\times$	22.74	10018.06	1.00 $\times$
	+ Prefix CoT	22.26 -1.62	3517.61 -52.52	1.01 $\times$	19.76 -2.98	10034.73 $+16.67$	1.00 $\times$
	ICE-SP	23.68 -0.20	2132.79 -1437.34	1.67 $\times$	22.74 0.00	5885.87 -4132.19	1.70 $\times$
	ICE-PP	26.88 $+3.00$	3155.99 -414.14	1.13 $\times$	25.28 $+2.54$	8774.40 -1243.66	1.14 $\times$
GPQA	Vanilla	27.46	970.84	1.00 $\times$	28.13	2156.62	1.00 $\times$
	+ Prefix CoT	27.68 $+0.22$	1043.77 $+72.93$	0.93 $\times$	27.90 -0.23	2234.75 $+78.13$	0.97 $\times$
	ICE	32.37 $+4.91$	50.45 -920.39	19.24 $\times$	33.70 $+5.57$	51.24 -2105.38	42.08 $\times$
MMLU	Vanilla	49.67	19396.79	1.00 $\times$	60.35	47795.97	1.00 $\times$
	+ Prefix CoT	51.22 $+1.55$	19469.64 $+72.85$	1.00 $\times$	60.11 -0.24	48298.11 $+502.14$	0.99 $\times$
	ICE	62.77 $+13.10$	145.75 -19251.04	133.08 $\times$	61.09 $+0.74$	172.79 -47623.18	276.67 $\times$

Table 1: ICE 在不同任务上的性能比较。实验是使用 LLaDA-8B-Instruct（长度为 256）和 LLaDA-1.5（长度为 512）进行的，块长度和生成步骤与生成长度匹配。前缀 CoT 指的是带有 CoT 提示作为前缀的 vanilla。ICE 在推理加速时以 SP 模式运行，而在提高准确性时以 PP 模式运行。对于 MMLU 和 GPQA，统一配置同时实现了卓越的准确性提升和效率增益。

Task	Method	Acc.	Latency	Speedup
GSM8K	Vanilla	46.55	1461.83	1.00 $\times$
	ICE-SP	46.01	354.86	4.12 $\times$
	+ Cache	46.17	239.83	6.10 $\times$
	ICE-PP	63.84	874.53	1.67 $\times$
	+ Cache	61.56	592.73	2.47 $\times$
MATH	Vanilla	23.88	3570.13	1.00 $\times$
	ICE-SP	23.68	2132.79	1.67 $\times$
	+ Cache	23.72	1771.15	2.02 $\times$
	ICE-PP	26.88	3155.99	1.13 $\times$
	+ Cache	25.94	2684.77	1.33 $\times$

Table 2: ICE 与 dLLM-Cache 在 LLaDA-8B-Instruct 上跨 GSM8K 和 MATH 的兼容性评估。

消融研究

为了更深入地了解 ICE 的设计选择，我们进行了消融研究，重点关注推动性能的关键超参数和架构决策。关键组件的影响。表格 3 展示了 ICE 架构组件的渐进贡献。最初的思维/答案分段机制（+ Segment）在所有基准上提供了一致的改进，对于 LLaDA-8B-Instruct 和 LLaDA-1.5，分别在 GSM8K 上提升了 +9.40 % 和 +7.35 %，验证了在离散掩码语言模型中显式推理结构的基本重要性。结构化思维细分的引入进一步提升了性能，在 GSM8K 上贡献了额外的 +8.42 % 和 +2.50 %，表明细粒度的推理过程分解能够更有效地利用掩码生成范式。基于置信度的提前退出机制（ICE）表现出任务依赖行为：在 GSM8K 上虽表现出轻微的准确性权衡（对于 LLaDA-8B-Instruct 为 -0.53 %），但在知识密集型任务（如 GPQA）上提供了显著的改进（+0.67 % 和 +3.57 %），说明提前退出策略在跨任务推理复杂性

Task	Vanilla	+ Segment (T/A)	+ Structured Thinking	+ Early Exit (ICE)
LLaDA-8B-Instruct				
GSM8K	46.55	55.95	64.37	63.84
MATH	23.88	24.18	26.88	26.88
GPQA	27.68	29.91	31.70	32.37
LLaDA-1.5				
GSM8K	45.19	52.54	55.04	58.22
MATH	22.74	23.74	24.64	25.28
GPQA	27.90	29.91	30.13	33.70

Table 3: ICE 核心组件的消融研究。

显著变化时特别有效。

推理步骤的效果 (N_t)。图 5 展示了推理步骤的粒度和模型性能之间的关键关系。我们的系统分析揭示了不同任务领域的最佳运行点：GSM8K 在 $N_t = 3$ 处达到约 58-60 % 准确率的峰值表现，而 MATH 在 $N_t = 4$ 处以 25-26 % 准确率展现出最佳结果。这一发现表明数学推理任务受益于中等的分解深度，这在推理粒度与计算开销之间取得了平衡。值得注意的是，过度细分 ($N_t > 6$) 在所有基准测试上都一致降低了性能，表明过于细粒度的推理步骤可能引入噪声和计算效率低下。对于像 GPQA 这样知识密集型的任务，该框架在较广泛的思考数范围内（28-31 % 准确率）保持稳定性能，显示出对超参数变化的鲁棒性。

掩码令牌分配策略。我们评估了三种不同的策略来分配掩码令牌在推理步骤中的分布：均匀分配为每个思维步骤保持相等的令牌预算，前重分配优先考虑初始推理步骤，而后重分配则将计算资源集中在最后的推理阶段。实验结果表明，后重和前重策略始终优于均匀分配，尤

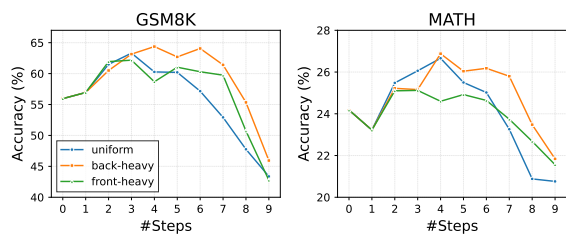


Figure 5: 关于推理步骤的消融研究 (N_t)。

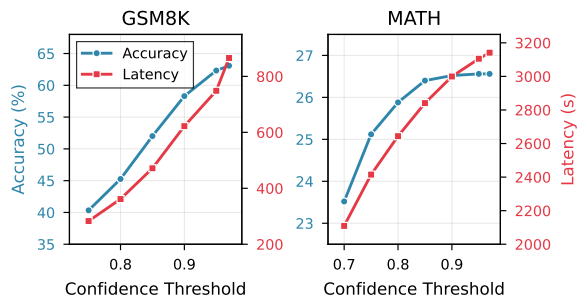


Figure 6: 关于置信阈值 τ 的消融研究。

其在较低的思维数量情况下 ($N_t \leq 4$)。这一发现表明，相较于资源的平均分配，战略性的令牌集中能够产生更卓越的性能，这凸显了在结构化推理框架中，自适应计算分配的重要性。

阈值分析。这一置信度阈值 τ 作为重要的超参数控制我们早退出机制中的速度与准确性权衡。图 6 系统地展示了在数学推理基准 (GSM8K 和 MATH) 上的这一关系。我们的分析揭示了一个明显的模式：较低的阈值在牺牲准确性的情况下强烈优先考虑计算效率，而较高的阈值则强调准确性的保持，速度提升的益处较少。关键是，适中的阈值实现了最佳平衡，在显著提高准确性的同时保持明显的计算效率增益，从而验证了我们基于置信度的早退出策略的有效性。对于知识密集型任务 (GPQA 和 MMLU)，我们观察到过高的置信度阈值收益递减，准确性趋于平稳而不是进一步提高。这一发现强调了任务自适应阈值选择的重要性，其中推理复杂性应指导最佳的置信度校准以实现最大效益。

讨论

就地提示：dLLMs 的新范式

我们的工作将链式思维从顺序的预计算重新定位为 dLLM 迭代细化过程中的一个动态、共生的组件。通过直接在生成画布中嵌入提示，我们突破了自回归模型仅限前缀的限制。这种整合促进了一种共同细化的动态，其中推理轨迹和最终答案在生成步骤中并行演化，互相启发。

这一范式转变是 dLLMs 独有的。与生成式模型中生成的推理步骤不可更改不同，我们的方法允许模型根据出现的答案重新审视并完善其整个思维过程。推理模板 ($T_1, \dots, T_{N_t}$) 更像是一个灵活的框架，而不是僵化的指令，指导模型在适应其对问题的不断发展的理解的同时，进行完善过程。这表明结构化问题分解与扩散模型的核心机制之间存在基本的兼容性，将推理从静态的前

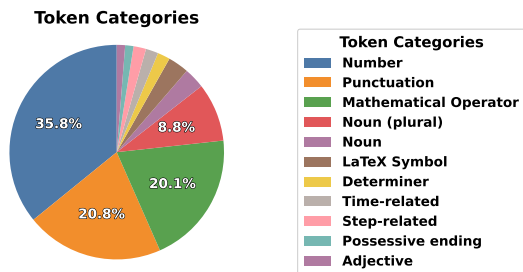


Figure 7: 在 GSM8K 生成过程中信心迅速变化时解码的标记类别统计。

提转变为并行优化的过程。这暗示 dLLMs 可以被设计为具有内部框架，用于其他复杂任务，例如受限生成和规划。

在 dLLMs 中的内部动力学

解耦求解与解释生成。我们的研究表明，dLLMs 有效地将求解过程与解释生成解耦。尽管推理轨迹仍不稳定，但答案部分的快速信心收敛表明模型在完成叙述性解释之前已稳定最终答案。这与 AR 模型中解决方案和解释组成顺序链条的情况形成对比，这使得我们的提前退出机制能够利用 dLLM 的基本特性，而不仅仅是提供计算效率。

逐字级别的置信度动态。逐字级别的分析揭示，置信度的成熟是通过果断的跳跃而非逐步增加来实现的。图 7 展示了在生成 GSM8K 过程中置信度迅速变化时被解码的词类分布。统计数据显示，这些关键性的变化主要由数字词的稳定性推动。模型首先固定定量结果，然后通过调整标点符号和数学运算符来巩固推理语法。这种分层的收敛过程在构建解释性框架之前就锁定了核心结果。有趣的是，这一发现与 SepLLM (Chen et al. 2025) 的观察相呼应，该观察识别出在自回归 LLM 中标点符号词的高注意力分数，暗示结构性词在不同模型架构中扮演了关键角色。

结论

我们引入了 ICE，一个新颖的框架，通过就地 CoT 提示和具信心的早退出机制的两阶段解码，增强了 dLLMs 的推理能力和推断效率。我们的方法利用了 dLLMs 的双向注意力的天然优势，在 GSM8K 上实现了最高 17.29 % 的准确性提升和 $4.12 \times$ 的加速，并在 MMLU 上实现了高达 $276.67 \times$ 的加速。该工作展示了推理模式与生成机制之间的结构对齐可以产生协同效应，将迭代改进从计算负担转化为结构优点，并为非自回归语言模型的高效推断确立了新的范式。

References

- Austin, J.; Johnson, D. D.; Ho, J.; Tarlow, D.; and van den Berg, R. 2021a. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34: 17981–17993.
- Austin, J.; Odena, A.; Nye, M.; Bosma, M.; Michalewski, H.; Dohan, D.; Jiang, E.; Cai, C.; Terry, M.; Le, Q.; et al. 2021b. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.
- Berglund, L.; Tong, M.; Kaufmann, M.; Balesni, M.; Stickland, A. C.; Korbak, T.; and Evans, O. 2023. The Reversal Curse: LLMs trained on “A is B” fail to learn “B is A”. *arXiv preprint arXiv:2309.12288*.
- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.
- Chen, G.; Shi, H.; Li, J.; Gao, Y.; Ren, X.; Chen, Y.; Jiang, X.; Li, Z.; Liu, W.; and Huang, C. 2025. SepLLM: Accelerate Large Language Models by Compressing One Segment into One Separator. In *International Conference on Machine Learning*. Also available at *arXiv:2412.12094*.
- Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Devlin, J. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Gao, L.; Tow, J.; Abbasi, B.; Biderman, S.; Black, S.; DiPofi, A.; Foster, C.; Golding, L.; Hsu, J.; Le Noac’h, A.; Li, H.; McDonell, K.; Muennighoff, N.; Ociepa, C.; Phang, J.; Reynolds, L.; Schoelkopf, H.; Skowron, A.; Sutawika, L.; Tang, E.; Thite, A.; Wang, B.; Wang, K.; and Zou, A. 2024. A framework for few-shot language model evaluation.
- Gao, Y.; and et al. 2023. Synthesizing and Debugging Chain-of-Thought Prompts with Large Language Models. *arXiv*.
- Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.
- Hu, Z.; Meng, J.; Akhauri, Y.; Abdelfattah, M. S.; Seo, J.-s.; Zhang, Z.; and Gupta, U. 2025. Accelerating diffusion language model inference via efficient kv caching and guided diffusion. *arXiv preprint arXiv:2505.21467*.
- Kojima, T.; Hashimoto, D.; Shimbo, T.; and Inui, K. 2022. Large Language Models Are Strong Reasoners. *arXiv*.
- Liu, K.; et al. 2025a. Mercury: Ultra-Fast Language Models Based on Diffusion. *arXiv preprint arXiv:2506.17298*.
- Liu, X.; Liu, Z.; Huang, Z.; Guo, Q.; He, Z.; and Qiu, X. 2025b. LongLLaDA: Unlocking Long Context Capabilities in Diffusion LLMs. *arXiv preprint arXiv:2506.14429*.
- Liu, Z.; Yang, Y.; Zhang, Y.; Chen, J.; Zou, C.; Wei, Q.; Wang, S.; and Zhang, L. 2025c. dllm-cache: Accelerating diffusion large language models with adaptive caching. *arXiv preprint arXiv:2506.06295*.
- Lou, A.; Meng, C.; and Ermon, S. 2023. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*.
- Luxembourg, O.; Permuter, H.; and Nachmani, E. 2025. Plan for Speed-Dilated Scheduling for Masked Diffusion Language Models. *arXiv preprint arXiv:2506.19037*.
- Ma, X.; Yu, R.; Fang, G.; and Wang, X. 2025. dkv-cache: The cache for diffusion language models. *arXiv preprint arXiv:2505.15781*.
- Nakamura, Y.; and et al. 2021. WebGPT: Browser-assisted question-answering with human feedback. In *NeurIPS*.
- Nie, S.; Zhu, F.; You, Z.; Zhang, X.; Ou, J.; Hu, J.; Zhou, J.; Lin, Y.; Wen, J.-R.; and Li, C. 2025. Large Language Diffusion Models. *arXiv preprint arXiv:2502.09992*.
- Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.
- Sahoo, S. S.; Arriola, M.; Schiff, Y.; Gokaslan, A.; Marroquin, E.; Chiu, J. T.; Rush, A.; and Kuleshov, V. 2024. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*.
- Shi, J.; Han, K.; Wang, Z.; Doucet, A.; and Titsias, M. K. 2024. Simplified and Generalized Masked Diffusion for Discrete Data. *arXiv preprint arXiv:2406.04329*.
- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Wei, J.; Chen, X.; Yang, Y.; Klein, D.; Polyak, A.; Chandra, S.; M., S.; and Tan. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv*.
- Wei, Q.; Zhang, Y.; Liu, Z.; Liu, D.; and Zhang, L. 2025. Accelerating Diffusion Large Language Models with SlowFast: The Three Golden Principles. *arXiv preprint arXiv:2506.10848*.
- Wen, Z.; Qu, J.; Liu, D.; Liu, Z.; Wu, R.; Yang, Y.; Jin, X.; Xu, H.; Liu, X.; Li, W.; et al. 2025. The Devil behind the mask: An emergent safety vulnerability of Diffusion LLMs. *arXiv preprint arXiv:2507.11097*.

Wu, C.; Zhang, H.; Xue, S.; Liu, Z.; Diao, S.; Zhu, L.; Luo, P.; Han, S.; and Xie, E. 2025. Fast-dLLM: Training-free Acceleration of Diffusion LLM by Enabling KV Cache and Parallel Decoding. arXiv:2505.22618.

Yang, L.; Tian, Y.; Li, B.; Zhang, X.; Shen, K.; Tong, Y.; and Wang, M. 2025. Mmada: Multi-modal large diffusion language models. arXiv preprint arXiv:2505.15809.

Ye, J.; Xie, Z.; Zheng, L.; Gao, J.; Wu, Z.; Jiang, X.; Li, Z.; and Kong, L. 2025. Dream 7B.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. arXiv preprint arXiv:2303.18223.

Zhu, F.; Wang, R.; Nie, S.; Zhang, X.; Wu, C.; Hu, J.; Zhou, J.; Chen, J.; Lin, Y.; Wen, J.-R.; and Li, C. 2025. LLaDA 1.5: Variance-Reduced Preference Optimization for Large Language Diffusion Models. ArXiv, abs/2505.19223.

实现细节

数据集。我们在四个涵盖数学和知识密集型任务的多样化基准上评估 ICE。对于数学推理，我们使用 GSM8K (Cobbe et al. 2021)，一个包含 8500 个需要多步推理的小学数学文字题的数据集，以及 MATH (Hendrycks et al. 2021)，一个跨代数、几何及其他领域的高中竞赛数学难题集合。对于知识密集型推理，我们采用 MMLU (Hendrycks et al. 2020)，一个涵盖从基础数学到高级专业主题的 57 个学科的综合基准，以及 GPQA (Rein et al. 2023)，一个旨在评估生物学、物理学和化学高级推理的研究生级别的 Google 防作弊问答数据集。这些基准综合评估了跨不同领域和难度等级的推理能力的广度和深度。

模型和评估协议。我们使用两个具有代表性的大规模离散语言模型 (dLLMs) 进行全面评估：LLaDA-8B-Instruct，这是一个从零开始训练的具有 8B 参数的掩蔽离散扩散模型，以及 LLaDA-1.5，该模型结合了方差减少偏好优化 (VRPO) 以改善与人类偏好的对齐。为了确保结果的一致性和可重复性，我们在所有基准上采用了广泛使用的语言模型评估框架 (Gao et al. 2024)。

ICE 的详细伪代码显示在算法 1 中。

Algorithm 1: ICE: 原地连续提示与早退出

Require: Prompt $\mathbf{y}_{\text{prompt}}$, thinking template $T = \{T_1, T_2, \dots, T_{N_t}\}$, answer indicator, confidence threshold τ , max steps N

Ensure: Generated sequence $\hat{\mathbf{y}}_{\text{final}}$

- 1: // Phase 1: Initialize structured sequence
- 2: $\mathbf{y}_{\text{thinking}}^{(N)} \leftarrow (T_1, T_2, \dots, T_{N_t})$ {Insert thinking templates}
- 3: $\mathbf{y}_{\text{answer}}^{(N)} \leftarrow ([\text{MASK}], \dots, [\text{MASK}])$ {Mask all answer tokens}
- 4: $\mathbf{y}^{(N)} \leftarrow (\mathbf{y}_{\text{prompt}}, \mathbf{y}_{\text{thinking}}^{(N)}, \mathbf{y}_{\text{answer}}^{(N)})$
- 5: $k \leftarrow N$, phase $\leftarrow$ reasoning
- 6: while $k > 0$ and phase = reasoning do
- 7: // Generate prediction for current step
- 8: $\hat{\mathbf{y}}_0^{(k)} \leftarrow \arg \max_v f_\theta(\mathbf{y}_0 = v \mid \mathbf{y}^{(k)})$ {Eq. 4}
- 9: // Compute confidence scores
- 10: for $i \in \{1, \dots, L\}$ do
- 11: $\text{confidence}_i^{(k)} \leftarrow \max_{v \in \mathcal{T}} f_\theta(\mathbf{y}_{0,i} = v \mid \mathbf{y}^{(k)})$ {Eq. 5}
- 12: end for
- 13: // Check early exit condition
- 14: $\text{avg_conf}_{\text{answer}}^{(k)} \leftarrow \frac{1}{L_{\text{answer}}} \sum_{i \in \text{answer indices}} \text{confidence}_i^{(k)}$ {Eq. 7}
- 15: if $\text{avg_conf}_{\text{answer}}^{(k)} > \tau$ then
- 16: phase $\leftarrow$ answer_generation {Early exit triggered}
- 17: break
- 18: end if
- 19: // Update only thinking section
- 20: $\mathbf{y}^{(k-1)} \leftarrow S_{\text{thinking}}(\hat{\mathbf{y}}_0^{(k)}, \mathbf{y}^{(k)}, k)$ {Selective unmasking}
- 21: $k \leftarrow k - 1$
- 22: end while
- 23: // Phase 2: Answer generation
- 24: if phase = reasoning then
- 25: phase $\leftarrow$ answer_generation {Normal termination}
- 26: end if
- 27: // Single-step answer decoding
- 28: $\hat{\mathbf{y}}_{\text{final}} \leftarrow \arg \max_v f_\theta(\mathbf{y}_{\text{answer}} = v \mid \mathbf{y}_{\text{current}})$
- 29: return $\hat{\mathbf{y}}_{\text{final}}$
