拡散言語モデルのテキスト生成順序の最適化

浅野輝 [1,2]　**小津野将** [3]　**齋藤邦章** [3]　**馬場雪乃** [1]
[1] 東京大学　[2]RIKEN AIP　[3]OMRON SINIC X
asano-hikaru19, yukino-baba@g.ecc.u-tokyo.ac.jp,
tadashi.kozuno, kuniaki.saito@sinicx.com

概要

マスク拡散言語モデル（Masked Diffusion Language Model; MDLM）は、任意順序でのテキスト生成を可能にする次世代の大規模言語モデルとして注目されている。その柔軟性により、推論能力やデータ効率の面で自己回帰モデルを上回る性能が報告されている一方、推論時に「どの位置を先に unmask するか（where-to-unmask）」という順序選択が生成経路を規定し、性能および安定性を大きく左右する。しかし、MDLM の標準的な学習は主としてトークン復元（what-to-unmask）を最適化しており、順序の学習はヒューリスティックに委ねられてきた。本研究では順序の影響を明示的に分析するため、訓練時に利用可能な正解系列を参照した順序オラクル（Gt-Prob / Gt-Margin）を新たに導入し、順序選択の理想的な上限性能を評価する。LLaDA 8B を用いた GSM8K・MATH・AQuA・StrategyQA のベンチマーク実験を通じて、Gt-Margin に基づく順序選択が既存の margin 指標を大きく上回り、where-to-unmask が最終性能を決定づける重要な要因になり得ることを確認した。

1　はじめに

マスク拡散言語モデル（Masked Diffusion Language Model; MDLM）[1] は、マスクトークン（<mask>）を用いた拡散過程により、テキスト生成に新たな自由度をもたらす手法である。左から右への単方向生成に縛られる自己回帰（AR）モデルとは異なり、MDLM は任意の順序でトークンを復元できるため、文章の補完や編集などの非順序的なタスクに本質的な適性を持つ。加えて、プログラミングや数学などの論理推論タスクにおいて同規模の AR モデルを上回る性能が示されており、その構造的な優位性から次世代の基盤モデル候補として期待されている [2, 3, 4]。

MDLM によるテキスト生成は二つの決定を含んでいる。すなわち、「次にどの位置を unmask するか（where-to-unmask）」と、「その位置にどのトークンを生成するか（what-to-unmask）」である。what-to-unmask が単純な語彙分布の予測であるのに対し、where-to-unmask は生成経路そのものを決定するため、初期の選択ミスが後続の復元を制約し、誤りが全体へ波及する。したがって、順序選択は生成品質を左右する重要な設計要素となる。

しかし、標準的な MDLM の学習は、ランダムにマスクされた入力から正解トークンを復元する穴埋め学習として実装され、損失関数は主として what-to-unmask の精度向上を目的としている [5, 6]。そのため、where-to-unmask の順序選択は明示的には最適化されず、**推論時の順序決定はモデルの不確実性を考慮したヒューリスティックに委ねられてきた**。近年では、順序そのものを最適化するために推論ループを環境として扱い、下流タスクの正解率などを報酬として where-to-unmask を強化学習で獲得する試みもあるが、多数のロールアウトを必要とするため、学習コストが非常に高いという問題がある。このように、訓練時に利用可能な正解系列（ground-truth）を最大限活用し、where-to-unmask を低コストで最適化する手法は、依然として確立されていない。

本稿では、**順序選択を最適化対象として明示的に扱い**、正解系列を用いた順序オラクルによって理想的な性能上限を特定し、このオラクル順序を模倣する効率的な学習枠組みを提案する。実験の結果、提案手法は GSM8K や MATH といった論理推論タスクにおいて既存手法を大幅に凌駕する性能を示した（表 1）。この結果は、where-to-unmask の最適化が性能向上における支配的要因であることを実証するとともに、オラクル模倣の精度向上により推論性能をさらに改善できる余地があることを示唆している。

2 準備

本稿では、語彙集合を $\mathcal{V}$、生成対象の長さを L とし、プロンプトを $\mathbf{c}$、生成部分を $\mathbf{x} = (x_1, \ldots, x_L) \in \mathcal{V}^L$ と表す。プロンプト $\mathbf{c}$ は常に条件として与えられる。マスクトークン m = <MASK> を含む拡張語彙を $\bar{\mathcal{V}} = \mathcal{V} \cup \{m\}$ とし、時刻 $t \in [0,1]$ における部分観測系列を $\mathbf{x}_t \in \bar{\mathcal{V}}^L$ とする。位置 i が未確定であることを $\mathbf{x}_{t,i} = m$ で表す。また、δ_a をデルタ分布として、$\delta_a(z) = \mathbb{1}[z = a]$ と定義する。

順過程 MDLM は、クリーンデータ $\mathbf{x}_0 = (x_1, \ldots, x_L)$ を順過程で段階的にマスクし、その逆過程で復元することで生成を行う。時刻 $t = 1$ は全マスク状態 $\mathbf{x}_1 = (m, \ldots, m)$、$t = 0$ は完全にクリーンなデータ $\mathbf{x}_0$ に対応する。

具体的には、マスキングスケジュール $\alpha_t \in [0,1]$（時刻 t においてトークンが**非マスク**である確率）を用い、$\alpha_0 = 1, \alpha_1 = 0$ とする。各位置を独立に確率 $1 - \alpha_t$ でマスクすることで、順過程の遷移確率は以下のように定義される。

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \prod_{i=1}^{L} \Big(\alpha_t \, \delta_{x_{0,i}}(\mathbf{x}_{t,i}) + (1 - \alpha_t) \, \delta_m(\mathbf{x}_{t,i}) \Big) \quad (1)$$

逆向き過程（生成） 生成過程 $t : 1 \to 0$ では、ニューラルネットワーク $\mu_\theta(\mathbf{x}_t, \mathbf{c})$ を用いてマスク位置を段階的に復元する。任意の時刻 $0 \le s < t \le 1$ に対するステップ $t \to s$ において、各マスク位置は確率 $(\alpha_s - \alpha_t)/(1 - \alpha_t)$ で非マスク化対象として選択され、復元対象集合 $U_{t \to s}$ が決定される（where-to-unmask）。続いて、各 $i \in U_{t \to s}$ に対して μ_θ が予測する分布 $\mu_{\theta,i}(\cdot \mid \mathbf{x}_t, \mathbf{c})$ からトークンをサンプリングする（what-to-unmask）。各位置の遷移確率は以下で与えられる。

$$p_\theta(x_{s,i} \mid \mathbf{x}_t, \mathbf{c}) = \begin{cases} \mu_{\theta,i}(x_{s,i} \mid \mathbf{x}_t, \mathbf{c}) & (\mathbf{x}_{t,i} = m,\ i \in U_{t \to s}), \\ \delta_{\mathbf{x}_{t,i}}(x_{s,i}) & (\text{otherwise}). \end{cases} \quad (2)$$

ここで、otherwise のケースは既知トークンの維持 $(\mathbf{x}_{t,i} \neq m)$ および未選択マスクの残留 $(\mathbf{x}_{t,i} = m,\ i \notin U_{t \to s})$ を表す。

学習 MDLM の学習は、ランダムなマスク位置の復元（穴埋め）として定式化される [5, 6]。時刻 t を区間 $[0,1]$ 上の一様分布 $\mathrm{Unif}(0,1)$ からサンプリングし、その t に対してマスク状態 $\mathbf{x}_t \sim q(\cdot \mid \mathbf{x}_0)$ を生成する。そして、以下の重み付き負対数尤度を最小化する。

$$\mathcal{L}(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \mathbf{x}_t} \left[\frac{\alpha_t'}{1 - \alpha_t} \sum_{i : \mathbf{x}_{t,i} = m} - \log \mu_{\theta,i}(x_{0,i} \mid \mathbf{x}_t, \mathbf{c}) \right]. \quad (3)$$

ここで、α_t' は α_t の時間微分である。この目的関数は、マスク位置のトークン予測（what-to-unmask）を直接最適化する。一方で、生成時の復元順序（where-to-unmask）は明示的に学習されず、モデルの予測確信度等から暗黙的に決定されるか、推論アルゴリズムとして外生的に与えられる。

3 手法

本節では、MDLM の推論における where-to-unmask を「部分観測状態 $\mathbf{x}_t$ に対して、各マスク位置へ順序スコアを付与し、その上位から復元位置を選択するランキング問題」として定式化する。まず、推論時に利用可能な代表的順序スコアを整理する。次に、訓練時にのみ利用可能な正解系列 $\mathbf{x}_0$ を参照する**順序オラクル**（Gt-Prob/Gt-Margin）を導入し、where-to-unmask の理想的な順序構造を明示的に定義する。順序オラクルは単なる「評価用の上限」ではなく、各時刻・各状態に対して**復元すべき位置の優先度（ランキング）**を与える。これにより、強化学習のような多数ロールアウトを介さずとも、理想的な順序構造を発見し、模倣学習の教師信号として活用できる。以降、簡潔さのためプロンプト $\mathbf{c}$ は省略する。

3.1 順序に基づくデコード

離散化した時刻列 $1 = t_K > \cdots > t_0 = 0$ に沿って、完全マスク列 $\mathbf{x}_{t_K} = (m, \ldots, m)$ から反復的にマスクを減らす。時刻 t におけるマスク位置集合を

$$M_t = \{ i \in \{1, \ldots, L\} : \mathbf{x}_{t,i} = m \} \quad (4)$$

と定義する。各反復 $k = K, \ldots, 1$ では、各 $i \in M_{t_k}$ に対し順序スコア $u_i(\mathbf{x}_{t_k})$ を計算し、その値に基づいて復元する位置集合 $S_{t_k} \subseteq M_{t_k}$ を選択する（where-to-unmask）。具体的には、$u_i(\mathbf{x}_{t_k})$ の降順で上位から $|S_{t_k}|$ 個を選ぶことで S_{t_k} を定める。選ばれた各 $i \in S_{t_k}$ に対してトークン分布 $\mu_\theta^{(i)}(\mathbf{x}_{t_k})$ により復元を行い、$\mathbf{x}_{t_{k-1}}$ を得る。

本研究では where-to-unmask の寄与を明確に切り分けるため、what-to-unmask は確率的サンプリングではなく最大確率トークンで決定する：

$$\hat{x}_{t_{k-1},i} = \arg\max_{v \in \mathcal{V}} \mu_{\theta,v}^{(i)}(\mathbf{x}_{t_k}) \qquad (i \in S_{t_k}). \quad (5)$$

各反復で確定させる位置数は、準備節で導入したマスキングスケジュール α_t と整合するように、時刻 t_k におけるマスク数の目標値

$$R_k = \lceil (1 - \alpha_{t_k})\, L \rceil \tag{6}$$

を定め、$|S_{t_k}| = R_k - R_{k-1}$ を満たすようにする。これにより、どの指標 u を用いても「各ステップで復元するトークン数」は固定され、差分は純粋に where-to-unmask の選択に帰着する。

3.2 順序スコア

推論時に利用可能な順序スコアとしては、モデルが自身の予測にどの程度の確信を持っているか（不確実性）を表す指標が広く用いられてきた。代表例として、位置 i における最大確率（Top Prob）[7]

$$u_i^{\text{top}}(\mathbf{x}_t) = \max_{v \in \mathcal{V}} \mu_{\theta,v}^{(i)}(\mathbf{x}_t), \tag{7}$$

および上位 2 候補の確率差（Margin）[8]

$$u_i^{\text{margin}}(\mathbf{x}_t) = p_{(1)} - p_{(2)} \tag{8}$$

がある。ここで $p_{(1)} \geq p_{(2)}$ は $\mu_\theta^{(i)}(\mathbf{x}_t)$ の確率上位 2 要素である。これらは推論時に追加情報を要さず計算可能である一方、あくまで**モデル内部の確信度**に依存しており、正解系列に対して「どの位置から埋めるのが安全か／有利か」を直接最適化するものではない。本研究では先行研究に倣い、u^{margin} を代表的な推論時順序スコアとして採用する。

3.3 順序オラクル

一方、訓練時には正解系列 $\mathbf{x}_0$ が既知である。この事実を利用すると、推論時には観測不可能であるが、where-to-unmask に関して**理想的な優先度**を与える指標を定義できる。本研究ではこれを**順序オラクル**と呼び、以下の二種類を導入する。

まず、位置 i において正解トークン $x_{0,i}$ を予測できる確率として

$$u_i^{\text{Gt-Prob}}(\mathbf{x}_t) = \mu_{\theta,x_{0,i}}^{(i)}(\mathbf{x}_t) \tag{9}$$

を定義する。さらに「誤ったトークンを選ぶ危険性」まで含めて安全性を測るため、正解トークンが他候補に対してどれだけ優勢かを表す

$$u_i^{\text{Gt-Margin}}(\mathbf{x}_t) = \mu_{\theta,x_{0,i}}^{(i)}(\mathbf{x}_t) - \max_{v \neq x_{0,i}} \mu_{\theta,v}^{(i)}(\mathbf{x}_t) \tag{10}$$

を導入する。

これらの順序オラクルは、各状態 $\mathbf{x}_t$ に対してマスク位置 M_t 上のスコア列 $\{u_i^{\text{oracle}}(\mathbf{x}_t)\}_{i \in M_t}$ を与えるため、

$$\pi_t^* = \text{argsort}_{i \in M_t}\, u_i^{\text{oracle}}(\mathbf{x}_t) \tag{11}$$

として**復元優先度のランキング**を一意に定められる（$u^{\text{oracle}} \in \{u^{\text{Gt-Prob}}, u^{\text{Gt-Margin}}\}$）。このランキングから、各ステップでの最適選択 $S_{t_k}^*$（上位 $R_k - R_{k-1}$ 個）も自然に誘導される。

本研究では、順序オラクルを（i）where-to-unmask の性能上限を与える参照点として用いることで、既存の推論時スコアとのギャップを定量化し、順序選択が最終性能へ与える影響を切り分ける。同時に（ii）式 (11) が与えるランキング／選択 $S_{t_k}^*$ は、各 $\mathbf{x}_t$ に対する**密な教師信号**であり、将来的に planner を $\mathbf{x}_t$ のみから π_t^* を近似するよう学習させるためのデータ（オフライン模倣・蒸留・learning-to-rank）として直接利用できる。すなわち、順序オラクルは「到達可能な理想順序の明確化」と「低コストな学習材料の提供」を同時に満たす基盤として位置づけられる。

4 実験

本節では、MDLM におけるデコード順序（where-to-unmask）が性能に与える影響を検証する。前節（3）で定義した順序スコアを比較し、順序オラクルによる理想化と固定長生成下での挙動を評価する。

4.1 実験設定

データセット　推論における順序の重要性が表面化しやすいベンチマークとして、GSM8K [9]、MATH [10]、AQuA [11]、StrategyQA [12] の 4 データセットを用いる。GSM8K は小学校レベルの算数文章題、MATH は競技数学レベルの問題、AQuA は多肢選択形式の代数問題、StrategyQA は推論手順が明示されない Yes/No 質問である。

モデル　基盤モデルには、8B 規模でスクラッチ学習された拡散言語モデル LLaDA [1] を用いる。デコードは手法節（3）の手順に従い、また、それぞれのデータセットに対して、Supervised Fine-tuning（SFT）[5] を行ったモデルを用い、順序スコアのみを変更して評価を行う。

順序スコアのベースライン　順序選択の比較対象として、Random（ランダム順序）、AR（左から右への自己回帰順序）、Inverse-AR（右から左への逆順序）、Top-Prob（式 (7)）、Margin（式 (8)）を用いる。

評価指標　各データセットの参照解答とモデル出力から抽出した予測解答の完全一致に基づく正解率

表 1　順序オラクルによる上限評価（正解率）

順序スコア	GSM8K	MATH	AQuA	StrategyQA
Random	0.330	0.195	0.520	0.533
AR	0.595	0.365	0.590	0.733
Inverse-AR	0.365	0.160	0.295	0.635
Top-Prob	0.555	0.215	0.465	0.655
Margin	0.605	0.180	0.540	0.685
Gt-Prob	0.770	0.200	0.495	0.660
Gt-Margin	**0.900**	**0.460**	**0.605**	**0.795**

を用いる。具体的には、全問題について、抽出した予測解答が参照解答と文字列として完全に一致するかを判定し、一致した問題数の割合を正解率として算出する。抽出はテンプレートに従って行い、最終解答部分のみを比較する。

4.2　実験 1：順序オラクルによる上限評価

本実験では、初期化における生成長の不確実性を排除して順序の効果だけを評価するため、生成対象の長さ L_{gen} には正解列の長さを用いる。すなわち、完全マスク列 $\mathbf{x}_{t_K} = (m, \ldots, m)$ の長さを ground-truth に合わせ、順序スコアだけを切り替える。

表 1 に既存の順序スコアと順序オラクルによる結果を示す。Gt-Margin は全てのデータセットで最良であり、特に GSM8K と MATH で改善幅が顕著である。同一のモデル・アルゴリズム（what-to-unmask）で順序のみを変更してこのような大きな差が生じることは、MDLM において where-to-unmask が最終性能の支配的要因になり得ることを強く示唆する。また、この上限評価は、追加学習を伴わずに推論経路（順序）を改善するだけでも大きな性能利得が得られ得ることを示す。順序最適化はモデル本体を変えずに推論過程を改良するため、計算資源の制約が厳しい場面でも有効な改善手段になり得る。

実験 1 より、順序のみによって到達可能な性能の上限が大きく、順序最適化が MDLM のボトルネックになり得ることが分かる。一方で、推論時に利用可能な順序スコア（Top-Prob や Margin）は Gt-Margin に届いておらず、このギャップは「推論時に観測できる不確実性から、正解に近い順序をいかに近似するか」が今後の主要課題であることを示唆する。特に、Top-Prob は単一候補の確率に依存するため過大信頼な位置を選びやすいのに対し、Margin は上位候補間の差を用いて「局所的に安定している空欄」を優先でき、頑健な順序選択に繋がると解釈できる。

表 2　固定長生成における順序スコアの比較

順序スコア	GSM8K	MATH	AQuA	StrategyQA
Random	0.200	0.135	0.525	0.415
AR	0.505	0.267	0.525	0.635
Inverse-AR	0.266	0.067	0.467	0.635
Top-Prob	0.600	0.235	0.555	0.635
Margin	0.605	0.180	0.540	0.685
Gt-Prob	0.630	0.367	0.505	0.505
Gt-Margin	**0.845**	**0.415**	**0.585**	**0.840**

4.3　実験 2：固定長生成

実験 1 では生成長に ground-truth を用いたが、実運用では出力長を固定して生成することが多い。そこで次に、生成対象の長さ L_{gen} を固定した場合でも、順序最適化の利得が残るかを検証する。

表 2 に固定長設定での結果を示す。固定長生成という現実的な推論設定のもとでも、Gt-Margin による順序最適化は大きな性能向上を維持しており、特に GSM8K や MATH で顕著な改善が観測される。この傾向は、固定長生成が本質的に持つ難しさ（生成長のミスマッチに起因する余剰領域の混入や、停止位置・終端処理の不確実性）に対しても、where-to-unmask の最適化が頑健に働くことを示唆する。また、効果の大きさが GSM8K/MATH で特に顕著であることは、途中で構築される文脈の整合性が推論手順の正しさに直結するタスクほど、順序選択の恩恵が大きいことを示す。

5　結論

本稿では、マスク拡散言語モデル（MDLM）の推論において「次にどの位置を unmask するか（where-to-unmask）」が最終性能を大きく左右する点に着目し、順序選択をランキング問題として定式化した。さらに、訓練時にのみ利用可能な正解系列を参照する順序オラクル（Gt-Prob / Gt-Margin）を導入し、where-to-unmask に関する到達可能な性能上限を明確化した。LLaDA 8B を用いた GSM8K・MATH・AQuA・StrategyQA での評価により、Gt-Margin に基づく順序選択が一貫して最良となることを確認した。

今後の課題として、推論時に利用可能な指標と順序オラクルとのギャップを埋めるため、ランキングを教師信号とした模倣学習による planner の学習、可変長出力での検証、ならびに where-to-unmask と what-to-unmask の共同最適化が挙げられる。

謝辞

本研究は、JST ムーンショット型研究開発事業 JPMJMS2236-8 の支援を受けたものです。

参考文献

[1] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. **arXiv [cs.CL]**, 2025.

[2] Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, and Lingpeng Kong. Diffusion of thought: Chain-of-thought reasoning in diffusion language models. In A Globerson, L Mackey, D Belgrave, A Fan, U Paquet, J Tomczak, and C Zhang, editors, **Advances in Neural Information Processing Systems**, Vol. 37, pp. 105345–105374. Curran Associates, Inc., 2024.

[3] Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. In **The Thirty-ninth Annual Conference on Neural Information Processing Systems**, 2025.

[4] Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. DiffuCoder: Understanding and improving masked diffusion models for code generation. **arXiv [cs.CL]**, June 2025.

[5] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K Titsias. Simplified and generalized masked diffusion for discrete data. In **Proceedings of the 38th International Conference on Neural Information Processing Systems**, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc.

[6] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In **Proceedings of the 41st International Conference on Machine Learning**, ICML'24, Vienna, Austria, 2024. JMLR.org.

[7] Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. In **The Thirteenth International Conference on Learning Representations**, 2025.

[8] Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In **Forty-second International Conference on Machine Learning**, 2025.

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

[10] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In J Vanschoren and S Yeung, editors, **Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks**, Vol. 1, 2021.

[11] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In Regina Barzilay and Min-Yen Kan, editors, **Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 158–167, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[12] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. **Transactions of the Association for Computational Linguistics**, Vol. 9, pp. 346–361, 2021.

A　実験の詳細

本付録では、本文の再現性向上のために、固定長生成における生成長の設定、SFT の学習条件、および推論・評価で用いたプロンプトテンプレートを補足する。

A.1　固定長生成における生成長の設定

固定長生成の実験（§4）では、各データセットごとに生成対象長 L_{gen} を固定し、以下の値を用いた：

- GSM8K: $L_{\text{gen}} = 256$
- MATH: $L_{\text{gen}} = 512$
- AQuA: $L_{\text{gen}} = 256$
- StrategyQA: $L_{\text{gen}} = 128$

これらは、各データセットにおける参照解答の典型的な長さ（特に推論文・式展開を含む出力長）を考慮して、必要十分な長さとなるように設定した。すなわち、推論に必要な記述が途中で切断されやすい設定を避けつつ、過度に長い固定長によって余剰領域が増え、最終解答抽出が不安定化する状況を抑えることを意図している。

A.2　SFT (Supervised Fine-tuning) の学習条件

各データセットに対して SFT を行ったモデルを用いて評価した。ベースのモデルには LLaDA-8B-Base[1] を用いた。SFT の主要ハイパーパラメータを表 3 に示す。

表 3　SFT の主要ハイパーパラメータ

ハイパーパラメータ	値
学習エポック数	5
LoRA rank (r)	32
LoRA alpha (lora_alpha)	64
LoRA dropout (lora_dropout)	0.05
バッチサイズ	2
学習率	2e-4
学習率スケジューラ	cosine

なお、上記以外の最適化手法・正則化・学習の細部については、LLM の学習用のライブラリである transformers[2] の標準設定に従った。

```
Respond in the following format:
<reasoning>
...
</reasoning>
<answer>
...
</answer>

{<question>}
```

図 1　推論・評価で用いたプロンプトテンプレート

A.3　プロンプトテンプレート

図 1 に、推論・評価の際に用いたプロンプトテンプレートを示す。本テンプレートは、出力を<reasoning>と<answer>に明確に分離し、最終解答の抽出規則を単純化することを目的としている。

1）https://huggingface.co/GSAI-ML/LLaDA-8B-Base
2）https://huggingface.co/docs/transformers/en/index