自己修正学習と UCB デコーディングによる離散拡散テキスト生成

浅田 真生 [1]　三輪 誠 [2,1]
[1] 産業技術総合研究所　[2] 豊田工業大学
masaki.asada@aist.go.jp　makoto-miwa@toyota-ti.ac.jp

概要

画像合成分野で成功を収めている拡散モデルは，従来の左から右への逐次生成に代わる，柔軟かつ反復的なテキスト生成手法として注目されている．しかし，訓練時と推論時におけるモデルの振る舞いの乖離が，性能向上の障害となっている．これは，訓練時には正解テキストにノイズを付加した系列からの復元を学習する一方で，推論時にはモデル自身の出力に基づく反復的な修正が求められるためである．本研究では，この課題を解決するため，**自己修正学習**と **UCB デコーディング**を統合した拡散モデルを提案する．自己修正学習は，モデル自身の中間出力を用いて推論時の自己修正挙動を学習させる訓練手法であり，UCB デコーディングは，再マスキング対象の選択を多腕バンディット問題として定式化し，探索と活用のバランスを制御する推論アルゴリズムである．実験より，要約生成，質問生成，および算術推論タスクにおいて提案手法はベースラインモデルを一貫して上回る性能を示した．

1　はじめに

自然言語生成の分野では，自己回帰（AutoRegressive; AR）モデル [1, 2] が標準的な枠組みとして広く用いられてきた．AR モデルは高い生成性能を示す一方で，生成が左から右へと固定された順序で進むため，生成途中における既存トークンの修正や，文脈全体を踏まえた再考が困難であるという制約を有する．近年，画像合成などの連続値領域で顕著な成功を収めている拡散モデル [3] に着想を得て，反復的な洗練過程に基づくテキスト生成手法が注目を集めている．拡散モデルでは，ランダムな初期状態から系列全体を段階的にデノイズすることで生成を行うため，単方向の生成順序に依存せず，文全体にわたるトークン間の依存関係の考慮が可能となる．

拡散モデルを離散的なテキスト生成へ適用するにあたり，解決すべき課題の一つとして，訓練時と推論時の設定の不一致が挙げられる．訓練時，離散拡散モデルは正解トークンにマスク付加した系列を復元するというタスクを通じて学習を行う．一方，推論時には，モデル自身が生成した出力をもとに，再マスキングを伴う反復的な生成・修正を行う必要がある．このような訓練時と推論時の乖離により，モデルは推論過程で生じる自己生成誤りからの修復を十分に学習できず，誤りの蓄積や事実性を欠いた局所解への収束が生じやすくなる [4]．この問題は，AR モデルにおける露出バイアス [5] と類似した性質を持つが，拡散モデルでは過去のトークンに限らず，未来のトークンも自己生成結果に依存するため，より広範に露出バイアスが生じうる．

本研究では，この問題に対処するため，訓練時と推論時を一貫して設計した離散拡散モデルを提案する．訓練時には，モデル自身の中間予測を再帰的に修正させる**自己修正学習**を導入し，推論時に必要となる自己修正能力を学習させる．さらに推論時には自己修正学習によって推論過程に即して得られる各トークン予測のモデル確信度を用い，Upper Confidence Bound (UCB) アルゴリズム [6] を利用した再マスキング（**UCB デコーディング**）によってテキスト生成を行う．これにより，離散拡散モデルにおける生成品質の向上を目指す．[1)]

2　関連研究

拡散モデルによるテキスト生成は，大きく連続拡散モデルと離散拡散モデルの二つに分類される．連続拡散モデルはトークン埋め込みの潜在空間を用い，ランダムなガウスノイズを段階的に付加・除去することで，ターゲットトークンの埋め込み表現へと近づけ，最終的に埋め込み表現をトークン列へと

1）本論文は，EACL2026 にて発表する研究成果 [7] に基づくものである．

変換することでテキスト生成を行う [8]．一方，離散拡散モデルはマスクトークンをノイズとして扱い，マスク穴埋めを複数ステップにわたって反復的に進めることでテキスト生成を行う [9]．離散拡散モデルは，Llama 3 [2] のような事前学習済みテキスト生成モデルを活用できるという利点を持つ一方で，その研究は未だ十分に進んでいるとは言えず，本分野にはさらなる検討の余地がある．そこで本研究では，離散拡散モデルに基づくテキスト生成の品質向上を目的とする．本節では以降，デコーダオンリーの AR モデルを基盤とした離散拡散テキスト生成モデル [10] の概要について述べる．

ターゲット系列 Y_0 に対する順方向プロセス q を次のように定義する．まず，時間ステップ $t = 0$ において，Y_0 はマスクトークンを全く含まない系列である．そこから各ステップにおいて，一定割合のトークンをランダムにマスクすることでノイズを付加し，ステップ T において全トークンがマスクされた系列 Y_T に至るまで，系列を徐々に劣化させていく．一方，逆方向プロセス p_θ では，AR モデルのパラメータを基盤としたデノイジングモデル f_θ を用いる．具体的には，通常の AR モデルで用いられる因果マスクを，未来トークンも参照可能なマスクへと置き換え，さらに Transformer の最終層において系列中のすべてのトークンを非自己回帰的に一括予測する．このモデルは，ソース文脈 X と任意のノイズ状態 Y_t を入力として，元の系列 Y_0 における各トークンの生成確率分布 $\hat{P}_0 = f_\theta(Y_t, X)$ を予測する．

訓練時には，ランダムに選択した時間ステップ $t \in \{1, \ldots, T\}$ におけるモデルの予測と正解系列との誤差を計算し，その期待値として定義される訓練損失 L を最小化する：

$$L = \mathbb{E}_{t\sim\mathcal{U}(1,T),Y_0\sim\mathcal{D},Y_t\sim q(Y_t|Y_0)}.$$

推論時には，逆方向プロセス p_θ に基づき，系列全体の生成を複数ステップにわたって反復的に行う．各デノイジングステップでは，現在の系列 Y_t に対してデノイジングモデル f_θ が全トークンについて生成確率分布を予測し，系列全体の推定を行う．その後，予測結果に基づいて一部のトークンを再マスクし，次ステップにおいて再び全トークンの予測を行う．このように，全トークンの予測と部分的な再マスキングを交互に繰り返すことで，生成結果を段階的に洗練させていく．再マスキングは，各ステップにおいて対象となるトークン数を制御しながら行われ，推論の進行に伴って再マスクされるトークンの割合は徐々に減少する．再マスキングの戦略としては，各ステップで一定割合のトークンをランダムに選択する方法に加え，モデルの予測確信度に基づき，確信度の低いトークンを優先的に再マスクする方法が用いられる．

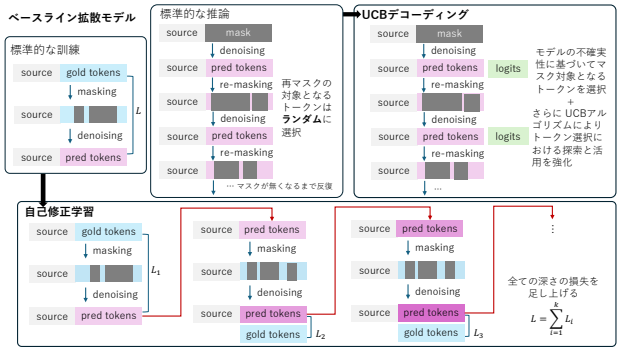


図 1　提案手法の概要

3　提案手法

本研究では，上記の乖離を低減しモデルの自己修正能力を高めるために，訓練時の目的関数の拡張と，UCB アルゴリズムを用いた推論時の再マスキングという二つのアプローチを提案する．提案手法の概要を図 1 に示す．

3.1　自己修正学習

自己修正学習では，推論時に必要とされる多段階の誤り修正プロセスを訓練時に取り込む．具体的には，モデル自身の予測を再度入力としてフィードバックし，繰り返し自己訂正させることで，誤りからの復元能力を向上させる．自己修正学習の手順は以下の通りである:

- ステップ 1: モデルは初期ノイズ状態 Y_t からクリーンな系列を予測し，その予測ロジットと正解系列 Y_0 との間の損失 L_1 を計算する．
- ステップ 2: 次に，モデルが出力した予測 $\hat{Y}_0^{(1)}$ を新たな入力系列とみなし，同一のノイズレベル t まで再度ノイズ付加して $Y_t^{(1)}$ を生成する．この系列を再びモデルでデノイジングし，出力と Y_0 との間の損失 L_2 を計算する．
- ステップ k: この処理を k 回繰り返し，合計損失 $L = \sum_{i=1}^{k} L_i$ を最小化する．

ここで，k はハイパーパラメータである．上記の損失関数の拡張により，通常の 1 ステップ（$k = 1$）の

データセット	手法		R-1	R-2	R-L	Faith.	Cove.	Cohe.
XSum	AR	Transformer [11]	30.6	10.8	24.4	-	-	-
		BART [1]	38.7	16.1	30.6	-	-	-
		ProphetNet [12]	39.8	17.1	32.0	-	-	-
	Diffu.	Diffusion-NAT [13]	38.8	15.3	30.8	-	-	-
		Two-step Diffusion [4]	38.5	14.8	30.9	-	-	-
		ベースライン拡散モデル	39.31	14.67	30.92	27.20	10.91	29.06
		提案手法	**43.25**	**19.25**	**35.26**	**47.51**	**15.04**	**43.18**
		w/o UCB デコーディング	40.71	17.05	33.79	25.69	8.41	21.39
		w/o 自己修正学習	41.69	17.83	33.93	43.60	12.54	37.80
データセット	**手法**		**R-L**	**B-4**	**MT**	**Cons.**	**AnsCons.**	**Cohe.**
SQuAD	AR	Transformer [11]	29.4	4.6	9.8	-	-	-
		BART [1]	42.5	17.0	23.1	-	-	-
		ProphetNet [12]	**48.0**	**19.5**	**23.9**	-	-	-
	Diffu.	Diffusion-NAT [13]	46.6	16.1	21.9	-	-	-
		Two-step Diffusion [4]	43.5	15.4	**23.0**	-	-	-
		ベースライン拡散モデル	40.87	13.39	20.86	63.60	60.25	53.56
		提案手法	44.07	16.06	21.81	**76.92**	**71.89**	**73.31**
		w/o UCB デコーディング	41.37	13.24	19.56	54.86	51.69	44.44
		w/o 自己修正学習	43.64	15.40	21.40	74.98	69.57	71.23

表 1　テキスト要約タスク XSum および質問生成タスク SQuAD における性能評価．R-1/R-2/R-L/B-4/MT はそれぞれ ROUGE-1/ROUGE-2/ROUGE-L/BLEU-4/METEOR を表す．Faith./Cove./Cohe./Cons./AnsCons. は LLM 評価の項目（0–100 に線形変換）を示す．**太字**は各評価指標での最高性能を示す．

手法	Accuracy (%)
ベースライン拡散モデル	46.92
提案手法	**57.31**
w/o UCB デコーディング	52.84
w/o 自己修正学習	53.60

表 2　算術推論タスク GSM8K における性能評価

学習では得られない自己修正能力を多段階の学習で引き出すことを目指す．

3.2　UCB デコーディング

推論時には，各デノイジングステップにおいてどのトークンを再マスクするかを選択する必要がある．本手法ではこの選択を，限られた予算（再マスク可能なトークン数）の配分問題として捉え，多腕バンディット問題として定式化する．すなわち，各トークン位置を腕とみなし，次に再マスクすべきトークンを逐次選択する問題としてモデル化する．具体的には，各トークンに対して以下の二つの指標を定義し，それらを組み合わせたスコアに基づいてマスク対象を決定する．

価値ヒューリスティック (V_i)　バンディットにおける「活用」に対応する指標であり，モデルの予測確信度やエントロピーなどから算出される不確実性の尺度である．本研究で利用した価値ヒューリスティックの詳細は付録 C に示す．

UCB 項　バンディットにおける「探索」に対応する項であり，これまでに再マスクされた回数 $N_i(t)$ が少ないトークンほど高く評価する．これにより，一見妥当に見えるものの局所的な解に陥っている可能性のあるトークンの再評価を促す．ステップ t における各トークン i の総合スコアは，上記二つの指標を加算して以下のように定義される：

$$\mathrm{Score}_i(t) = V_i + C\sqrt{\frac{\log(T-t+1)}{N_i(t)+1}}.$$

ここで C は探索と活用のバランスを調整する探索係数である．各ステップでは，$Score_i(t)$ が大きい順にトークンを選択し，そのうち上位 n_t 個を再マスクして N_i を更新した後，次のステップへ進む．このような再マスキング戦略により，高い確信度を伴う誤りに対しても再検証の機会を与えつつ，モデルの不確実性推定を活用した探索が可能となる．

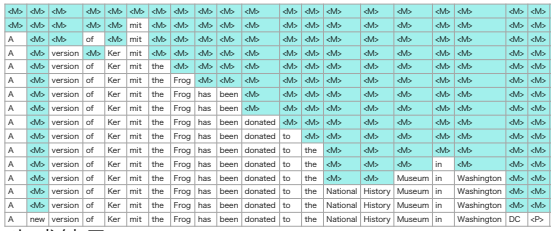
4　実験

4.1　実験設定

本研究では，テキスト要約タスク XSum [14]，質問生成タスク SQuAD [15]，算術タスク GSM8K [16] において，提案モデルをファインチューニングした

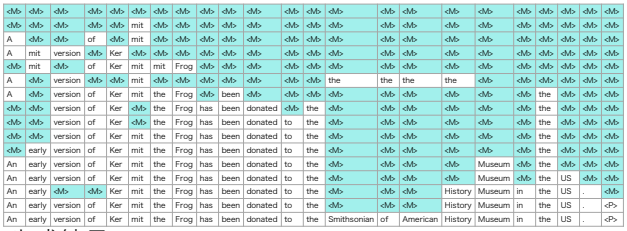
参照要約：
The original Kermit the Frog has been donated to the Smithsonian's National Museum of American History in Washington.

(a) UCBなし



生成結果：
A new version of Kermit the Frog has been donated to the National History Museum in Washington DC
LLM評価：
{Faithfulness: 2, Coverage: 2, Coherence: 3}

(c) UCBあり



生成結果：
An early version of Kermit the Frog has been donated to the Smithsonian of American History Museum in the US.
LLM評価：
{Faithfulness: **5**, Coverage: 2, Coherence: **4**}

図 2　UCB 項あり・なしによる反復的デノイジングプロセスの可視化（XSum データセット）

結果を評価した．評価指標として，テキスト要約タスクと質問生成タスクでは，ROUGE 等の自動評価指標に加えて，商用 LLM をジャッジとして生成文の忠実性や一貫性を 5 段階評価する LLM 評価も行った．データセットおよび LLM 評価の詳細は付録 A に示す．また，比較のためのベースライン拡散モデルは，訓練時に反復的な損失計算を行わず $k = 1$ とし，推論時にはランダムな再マスキングを用いる設定を意味する．使用したハイパーパラメータを付録 B に示す．

4.2　結果と考察

表 1 および表 2 より，提案フレームワークはテキスト要約タスク，質問生成タスク，算術推論タスクのすべてのデータセットにおいてベースライン拡散モデルを一貫して上回る性能を示した．

各要素の寄与を個別に分析すると，UCB デコーディングのみを導入した場合は，自己修正学習のみを導入した場合と比べて，一貫して大きな性能向上を示す．この結果は，再マスキング対象の選択戦略が生成品質に与える影響が大きく，推論過程における探索と活用の制御が特に重要であることを示唆している．一方で，自己修正学習を単独で導入した場合にも性能改善が確認されており，モデル自身の予測を学習過程に組み込むことが，生成品質の向上に寄与している．

さらに，自己修正学習と UCB デコーディングを組み合わせた場合には，これらの効果が相補的に作用し，すべての設定において最も高い性能を達成した．これは，自己修正学習によって得られる不確実性推定を，UCB デコーディングによる探索と活用の制御に活かすことで，再マスキング戦略がより効果的に機能したためであると考えられる．本研究で用いた価値ヒューリスティックの種類と，UCB 探索項の有無による性能比較については，付録 C に示す．また，学習時の自己修正深さ k と性能の関係についての検証結果は，付録 D に示す．

図 2 は，UCB 探索項の有無における拡散ステップ全体でのトークンの再マスキングパターンを示している．UCB なしの設定では，再マスキング領域が徐々に限られたトークン集合へと集中し，高い信頼度をもつトークンはステップを通じてほとんど変更されない．これに対し，UCB ありの設定では，再マスキングパターンが不確実な領域と，過去に高信頼と判断された領域との間で交互に切り替わっており，一度高信頼で予測されたトークンであっても再マスクされる場合があることを示している．

5　おわりに

本稿では，離散拡散型テキスト生成モデルにおける訓練時と推論時の乖離という課題に着目し，自己修正学習と UCB デコーディングを統合した生成フレームワークを提案した．自己修正学習は推論時の誤り分布を反映した学習を可能にし，UCB デコーディングは不確実性に基づく探索と活用の制御を実現する．複数のデータセットにおける実験の結果，自動評価指標と LLM 評価指標の双方において性能向上が確認された．今後は，本手法を大規模事前学習へ拡張し，自己修正と探索を中核とする信頼性指向の生成基盤モデルへと発展させる予定である．

謝辞

この成果は，NEDO（国立研究開発法人新エネルギー・産業技術総合開発機構）の委託業務（JPNP25006）の結果得られたものです．また，産総研及び AIST Solutions が提供する ABCI 3.0 を「ABCI 3.0 開発加速利用」の支援を受けて利用しました．

参考文献

[1] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics.

[2] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. **arXiv preprint arXiv:2407.21783**, 2024.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In **Proceedings of the 34th International Conference on Neural Information Processing Systems**, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[4] Masaki Asada and Makoto Miwa. Addressing the training-inference discrepancy in discrete diffusion for text generation. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, **Proceedings of the 31st International Conference on Computational Linguistics**, pp. 7156–7164, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

[5] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In **Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1**, NIPS'15, p. 1171–1179, Cambridge, MA, USA, 2015. MIT Press.

[6] T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. **Adv. Appl. Math.**, Vol. 6, No. 1, p. 4–22, March 1985.

[7] Masaki Asada and Makoto Miwa. Principled self-correction in discrete diffusion: A ucb-guided framework for text generation. In **The 19th Conference of the European Chapter of the Association for Computational Linguistics**, Rabat, Morocco, 2026. Association for Computational Linguistics.

[8] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation. In **Proceedings of the 36th International Conference on Neural Information Processing Systems**, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[9] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, **Advances in Neural Information Processing Systems**, 2021.

[10] 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 preprint arXiv:2502.09992**, 2025.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Advances in neural information processing systems**, pp. 5998–6008, 2017.

[12] Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In Trevor Cohn, Yulan He, and Yang Liu, editors, **Findings of the Association for Computational Linguistics: EMNLP 2020**, pp. 2401–2410, Online, November 2020. Association for Computational Linguistics.

[13] Kun Zhou, Yifan Li, Xin Zhao, and Ji-Rong Wen. Diffusion-NAT: Self-prompting discrete diffusion for non-autoregressive text generation. In Yvette Graham and Matthew Purver, editors, **Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1438–1451, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

[14] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[15] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, **Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing**, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[16] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. **arXiv preprint arXiv:2110.14168**, 2021.

Hyperparameter	Value
Backbone Model	Llama-3.2-1B
Peak Learning Rate	1e-4
Total Batch Size	128
Warmup Ratio	0.05
Weight Decay	0.001
Diffusion Steps	32
Inference Diffusion Steps	16
Max New Tokens	64
UCB Exploration (C)	1.5

表 3　使用したハイパーパラメータ

A　LLM 評価の詳細

本研究では，要約タスクと質問生成タスクにおいて，自動評価指標 (ROUGE, BLEU, METEOR) を補完するために，LLM-as-a-Judge に基づく評価枠組みを採用した．

要約タスクでは，評価者は入力文書を文脈として参照しつつ，生成要約を参照要約と比較する．評価は以下の 3 側面から構成される．

- Faithfulness: 生成要約と入力文書との整合性
- Coverage: 生成要約の入力文書情報網羅性
- Coherence: 生成要約の自然さ

質問生成タスクでは，評価者は入力文書，指定された回答，参照質問，およびモデルが生成した質問を受け取り，以下の 3 側面から評価を行う．

- Consistency: 生成質問と入力文書との整合性
- Answer Consistency: 生成質問と回答との整合性
- Coherence: 生成質問の自然さ

各側面は 1〜5 の 5 段階で採点され，あわせて 1〜3 文程度の総合的な説明文が付与される．すべての LLM 評価は OpenAI API を通じて `gpt-4.1-mini` を用いて実施した．スコアはデータセットの全サンプルにわたって平均化した後，0〜100 の範囲に線形変換して報告している．

B　ハイパーパラメータ設定

本研究で用いた主要なハイパーパラメータ設定を表 3 に示す．最適化手法として AdamW を用い，学習率は線形ウォームアップとコサイン減衰を組み合わせたスケジュールを採用した．すべての実験は，8 基の NVIDIA H200 GPU を用いて実施した．最大エポック数は，XSum データセットでは 3，SQuAD データセットでは 6，GSM8K データセットでは 20 に設定した．

UCB	Value	R-L	Faith.	Cove.	Cohe.
無	Unconfidence	34.96	45.13	13.79	38.05
	Uncertainty	35.12	45.81	14.37	40.04
	Entropy	34.72	42.53	13.15	35.78
有	Unconfidence	35.11	**48.53**	**15.07**	**44.36**
	Uncertainty	**35.26**	47.51	15.04	43.18
	Entropy	34.75	43.32	13.49	36.74

表 4　推論時における価値ヒューリスティックおよび UCB 探索項の設定に関する比較（XSum データセット）

	XSum		SQuAD	
自己修正深さ (k)	R-L	Faith.	R-L	Cons.
1 (ベースライン)	34.04	44.11	45.71	75.65
2	35.05	47.37	**46.16**	76.17
3	**35.41**	**48.04**	45.99	**76.73**
4	33.92	41.59	**46.16**	75.43
5	33.93	39.34	45.31	70.76

表 5　学習時の自己予測の深さ k が検証セットの性能に与える影響

C　価値ヒューリスティック

本研究では，価値ヒューリスティック V_i として，

- Unconfidence: $1 - P_i^{(\text{top-1})}$
- Uncertainty: $1 - (P_i^{(\text{top-1})} - P_i^{(\text{top-2})})$
- Entropy: $-\sum_j P_{i,j} \log P_{i,j}$

の 3 種類を使用した．表 4 は，推論時における各種設定に関するアブレーションスタディの結果を示している．本実験では，(i) UCB に基づく探索項を有効化するか否か，および (ii) トークン選択に用いる価値ヒューリスティック V_i を変化させて評価を行った．UCB 探索項を無効化した場合，三手法の中では Uncertainty に基づく手法が最も高い性能を示した．一方，UCB 探索項を有効化すると，全てのヒューリスティックで性能が向上し，ROUGE-L では Uncertainty が，LLM 評価では Unconfidence が最高性能を示した．

D　自己修正深さに関する解析

表 5 より，自己修正の深さ k の影響を分析した結果，ある程度までは k を深くするほど推論時のエラーからの回復能力が向上し，検証セット上の性能も改善される傾向が見られた．ただし最適な k の値はデータセットによって異なり，本実験においては $k = 2$ または $k = 3$ が最も高い性能を示した．一方で，それ以上に再帰ステップを重ねると性能が低下する場合も確認された．