言語モデルの言語獲得装置

三田 雅人, 染谷 大河, 吉田 遼, 大関 洋平

東京大学 THE UNIVERSITY OF TOKYO

{mita,taiga98-0809,yoshiryo0617,oseki}@g.ecc.u-tokyo.ac.jp

概要

大規模言語モデルは汎用的な文法能力を示す一方，学習効率では人間と大きな乖離がある．本研究では，生得的知識が文法の仮説空間を制約して効率的学習を可能にするという**言語獲得装置**仮説に着想を得て，新たな**事前事前学習**手法を提案する．提案手法は，自然言語による事前学習に先立ち，ミニマリスト文法から生成された自然言語の背後にある言語構造をモデルに先行学習させる．実験の結果，提案手法は通常のランダム初期化と比較して，同等の文法能力を維持しつつ，最大 35%のトークン効率改善を達成した．さらに，提案手法を抽象化した形式言語による統制実験により，先行研究で強調されてきたチョムスキー階層と回路複雑性の交差で整理される表現力とは別に，機能範疇の配置がもたらす**依存同定曖昧性の低さ**という新たな軸が学習効率に寄与しうることが示唆された．

1 はじめに

大規模言語モデル（LLM）はスケーリング則 [1] により人間並みの文法能力を示す一方で，言語獲得効率では依然として人間に大きく劣る．LLM が人間と同等の能力を獲得するには，人間が言語獲得で経験する刺激と比べて数桁多い学習データが必要であることが指摘されている [2]．この非効率性は，LLM が自然言語に特化した帰納バイアスを持たず，極めて自由度の高い状態からの統計学習に依存していることに起因しており [3]，人間の効率的な帰納バイアスを取り入れる余地は大きいと考えられる．

人間の効率的な言語獲得を説明する枠組みとして，**言語獲得装置（Language Acquisition Device; LAD）**仮説がある [4]．LAD 仮説は，生得的知識が文法の仮説空間を制約することで，限られた刺激（“刺激の貧困” [5]）からでも効率的な言語獲得が可能になるとする．

本研究では，LAD が示唆する帰納バイアスを言語モデルに組み込むことで，効率的な言語獲得が誘発されるかを検証する．具体的には，ミニマリスト文法（MG） [6] に基づき，自然言語の背後にある抽象的な統語構造を生成し，自然言語による事前学習（pre-training; PT）に先行して学習させる新たな**事前事前学習（pre-pretraining; PPT）**を提案する．そして，C4[7] を用いて訓練した Pythia-1B [8] に基づき評価実験を行ったところ，PPT 無しのベースラインと比較して，同等の文法能力を維持しつつ**最大 35%のトークン効率改善**を達成した．

さらに，この効率化の要因を切り分けるため，抽象化モデルによる統制実験を行った．その結果，MP-Struct の主要部駆動の構成を抽出した MP-Struct Core は，語彙数を揃えた対照系列を有意に上回り，先行研究 [9] で最良な形式言語バイアスとして報告された k-Shuffle Dyck を超える学習効率を達成した．このことは，PPT 言語の有効性をチョムスキー階層 [10] に基づく表現力と回路複雑性 [11] の交差として整理する先行研究の枠組み [9] に加えて，機能範疇が担う**構造的標識**[1] の配置が**依存同定曖昧性**を低下させることが，学習効率の差を説明しうる要因であることを示唆する．

2 関連研究

言語獲得装置 LAD 仮説は，人間は生得的な普遍文法（universal grammar; UG），すなわち生物学的に決定された一連の構造的制約を備えていると提唱する．この UG が可能な文法の仮説空間を劇的に狭めることで，「刺激の貧困」と呼ばれる貧弱な入力からであっても，効率的な言語獲得が可能になる [4, 5]．例えば，子供は単純な線形規則ではなく，階層構造に依存する規則を優先して学習すること

1） 本稿では，依存関係の始点や統語領域の境界を系列上で一意に同定しやすくするトークンを指す．

表 1: MP-STRUCT 系列例．**Merge**（階層），**Agree**（*T-DP* 照合），**Move**（主語-痕跡結合）の相互作用を示す．

要素	系列セグメント / 機構の説明

階層系列 (MERGE)	`[ CP [ C ] [ [ TP % C 領域`
	`[ DP... ] [ T(+EPP) ] % T 領域`
	`[ [ VP V ... [ TR ] ] ] ] ] % vP 領域`

Agree	探査子 T が領域内の DP を特定し素性照合 ($uNum \leftarrow val$).
Move	構造的主語位置への移動．先行詞 DP が痕跡 TR を束縛.
階層性	依存関係は c-統御など特定の階層構造内でのみ有効.

が報告されている [12]．この観点から UG は，「自然言語らしくない」仮説を先験的に排除しうる強力な帰納バイアスとして解釈できる．さらに近年のミニマリスト・プログラム（MP）は，UG の計算機構を MERGE/MOVE/AGREE といった最小限の操作へ還元し，言語機能の計算効率性を追求することで，このモデルをより先鋭化させている [13, 14, 15]．

非自然言語による事前事前学習　非自然言語による PPT は，言語モデルの新たな学習パラダイムとして台頭している．既存研究では，MIDI やプログラミング言語といった階層構造を持つデータでの次トークン予測が，有用な帰納バイアスを与え，その後の自然言語学習の効率性を向上させることが報告されている [16, 17, 18]．近年，Hu ら [9] は「表現力仮説」を提唱し，このパラダイムを前進させた．この仮説は，効果的な PPT 言語が，Transformer が学習可能な回路計算量の制約内 [19] で，チョムスキー階層 [10] に基づく構造的表現力を最大化したものであるとするものである．Hu らは k-Shuffle Dyck（例：`([{])}`）による効率化を示したが，これは言語的な制約よりも**無制約な表現力**を優先している．しかし，そこで許容される「恣意的な複雑性」は，自然言語固有の非対称性や主要部駆動性とは**乖離**している．本研究はこの点に着目し，単なる表現力の追求に代わり，言語獲得理論に基づく「自然言語に適合した」構造的制約を導入する．

3　提案手法

LAD の工学的近似として，MG [6] の派生に AGREE [14, 15] 等の理論的進展を統合した構造生成器（MP-STRUCT）を構築する．本手法の目的は，自然言語による事前学習に先立ち，この生成器から得られる系列を用いて，モデルの初期分布に**統語構造に由来する帰納バイアス**を導入することである．生

Algorithm 1 データ生成手続き (MP-STRUCT)

記法: $\mathcal{L}$: 語彙, $V/N/D$: 語彙範疇, vP: 動詞句, T/C: 時制/補文, $Spec$: 指定部, t: 痕跡, $u/iNum$: 数素性.

入力: 語彙集合 $\mathcal{L}$, パラメータ θ
出力: 構造情報を含むトークン系列 S
Step 1: MERGE による基本構造
語彙項目 $V, D, N \sim \mathcal{L}$ をサンプリング
D, N ペアから DP_{subj}, DP_{obj} を構築
vP を構築: $vP = [_{vP}\ DP_{subj}\ [_{V'}\ V\ DP_{obj}]]$
Step 2: 機能範疇と AGREE
$T_{[uNum]}$ を既存の vP と MERGE
探索 (Probe): T が vP 内の DP_{subj} を探索
値付与: AGREE(T, DP_{subj}) により $uNum \leftarrow iNum$ をセット
Step 3: MOVE (制約付き参照)
EPP: DP_{subj} を Spec-TP へ移動
$TP = [_{TP}\ DP_{subj_i}\ T\ [_{vP}\ t_i\ V\ DP_{obj}]]$
Wh: $C_{[\pm wh]}$ を MERGE
if C is $[+wh]$ **and** $\exists DP_{[+wh]}$ **then**
　目標 G を Spec-CP へ移動 (最短移動)
　$CP = [_{CP}\ G_k\ C\ [_{TP} \ldots t_k \ldots]]$
end if
Step 4: 線形化
派生木を行きがけ順 (Pre-order) で走査
非終端記号，特徴，痕跡を保持
語彙終端 (V, N, D) を**削除** $\rightarrow$ 系列 S

成プロセス（Algorithm 1）は以下の通りである．

Step 1: Merge による基底構造の構築　語彙集合から要素を選択し，MERGE 操作によりボトムアップに vP/VP を形成する．これにより，平坦な文字列ではなく再帰的な階層構造から成る「統語的骨格」を確立する．

Step 2: 機能範疇と Agree（非対称な依存）　機能範疇である時制辞（T）を導入する．この主要部は解釈不可能素性（$uNum$）を持つ「探査子（probe）」として機能し，自身の領域内を探索して主語 DP の素性を照合・値を決定する．このプロセスにより，依存関係は単なる線形な共起ではなく，機能範疇による構造的な探索に基づく非対称な関係として定義される．

Step 3: Move（制約された参照）　拡大投射原理（EPP）[2] を満たすため，主語 DP を Spec-TP へ移動させ（さらに任意で Wh 移動を行い），元の位置に痕跡（Trace）を残す．これは上位の機能範疇（T/C）と下位の領域を結ぶ長距離依存を形成するが，無制約な記憶ではなく，階層的な配置（Phase 等）によって厳密に制約された操作となる．

Step 4: 線形化　派生した木構造を走査し，語彙項目を削除して，構造括弧，非終端ラベル，素性，移動の痕跡のみからなる系列を出力する（表 1 参照）．これにより，モデルは語彙の意味的共起に頼

2)　定形節は構造的主語（Spec-TP）を持たなければならないとする要請 [13].

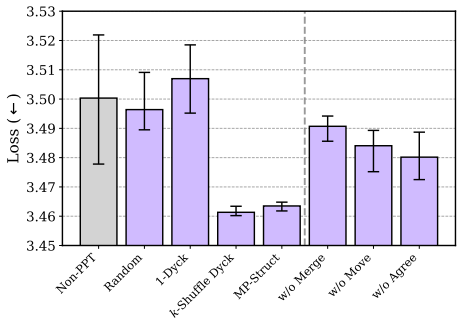

図 1: 25,000 ステップ時点における最良の損失.

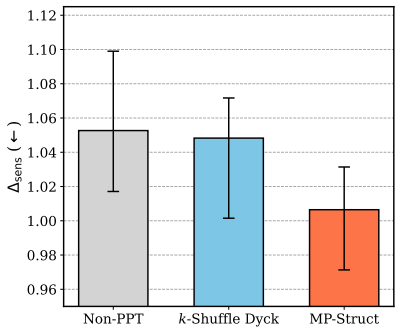

図 2: **意味的撹乱への頑健性** ($\Delta_{\text{sens}} = \mathcal{L}_{\text{JW}} - \mathcal{L}_{\text{NL}}$).

表 2: 実験結果. † は Non-PPT と比較して 5%水準で有意差があることを示す.

モデル	BLiMP	MRS	Efficiency
Non-PPT	0.758	–	–
Random	0.761	–	–
1-Dyck	0.759	–	–
k-Shuffle Dyck	**0.764**†	15.6 ±3.31	0.29 ±0.07
MP-Struct	0.755	15.3 ±2.86	0.29 ±0.06
Generic *k*-SD	0.760	12.0 ±5.15	0.22 ±0.10
MP-Struct Core	**0.764**†	**16.2** ±2.97	**0.31** ±0.06

ることなく，階層構造と機能的依存の相互作用を直接学習する．

4 実験

LAD が示唆する言語普遍的な構造制約を PPT を通じてモデルの初期分布へ反映させることで，自然言語学習の効率が促進されるかを検証する．

4.1 実験設定

Hu ら [9] のブロック学習パラダイムに準拠し，`Pythia-1B` [8] をベースモデルとして用いる．提案手法では PT 直前に PPT を挿入し，PPT 後のパラメータを PT の初期値として転送する．PT には `C4`[7] を用い 25,000 ステップ学習し，PPT は 500 ステップに統一する．すべての実験は乱数シード 3 種で実施し，平均を報告する．詳細は付録 A に示す．

ベースライン PPT の導入効果を測る **Non-PPT** に加え，構造を持たないデータによる初期化効果を対照とする **Random** を設定する．さらに，構造的特徴の質による差異を検討するため，単純再帰のみを持つ **1-Dyck** と，交差依存を含み最良な帰納バイアスと報告 [9] された ***k*-Shuffle Dyck** を採用する[3]．

評価指標 本研究では，文法汎化性能（BLiMP [20]）と学習効率の二軸で評価を行う．効率指標には，PPT による PT 代替効果を表す **MRS (Marginal Rate of Substitution)** と，総学習量の削減率を示す **Efficiency Gain** [9] を採用する[4]．

4.2 結果

図 1（左）および表 2 に示す通り，MP-Struct は Non-PPT や Random を有意に上回り，MRS 15.3，**平均 29%（最大 35%）のトークン削減**を達成した．これは，モデルが統語的骨格から獲得した構造処理機構を，自然言語学習へ効果的に転移させていることを示唆する．この効率化の要因を解明するアブレーション分析（図 1 右）では，Merge，Agree，Move のいずれを欠いても損失が悪化することが判明した．この結果は，効率化が特定の操作単体ではなく，階層構造（Merge）と機能的依存（Agree/Move）の**相乗的な相互作用**に起因することを裏付けている．

さらに特筆すべきは，MP-Struct が既存の最良手法である *k*-Shuffle Dyck と同等の学習効率（平均 29% 改善）を達成している点である．一方で，BLiMP スコア（表 2）は Non-PPT と同水準（0.755 vs 0.758）であった．これらの事実は，導入された構造的バイアスが，最終的な文法汎化性能を押し上げるというよりは，そこに至る学習プロセスの大幅な効率化に主として寄与することを示唆している．

5 分析

5.1 帰納バイアスの質: 構造的頑健性

効率改善の要因が，語彙の共起統計ではなく，意味から独立した構造処理メカニズムの獲得にあるかを検証する．そのために，学習用 `C4` および評価用 `WikiText` [21] から，機能語と語順を保持したまま内容語をランダムに置換した Jabberwocky（JW） [22] データを作成した．これにより，統語処理と意味的

3) 同研究において最良性能とされた $k = 64$ を採用した．

4) 詳細な定義と計算例は付録 B を参照.

表 3: 機能範疇を構造的標識とする **MP-Struct Core** と **Generic k-SD** の系列例．色: 階層, 依存, ヘッド.

条件	系列例
1. Generic k-SD	`[0 (1 (2 ]0 )1 [0 )2 ]0`
2. MP-Struct Core	`[0 H_C [0 H_T (1 [0 H_v ]0 )1 ]0 ]0`

相関を分離し，モデルが純粋な構造規則に基づいて生成を行う能力を定量化する．具体的には，意味情報の有無への感受性を $\Delta_{\text{sens}} = \mathcal{L}_{\text{JW}} - \mathcal{L}_{\text{NL}}$ として定義する．ここで $\mathcal{L}_{\text{NL}}$ は自然言語，$\mathcal{L}_{\text{JW}}$ は JW での損失である．**Δ_{sens} が小さいほど**，モデルが表層的な意味共起に依存せず，抽象的な統語構造に対して頑健であることを意味する．

結果（図 2）として，Non-PPT は Δ_{sens} が大きく意味共起への依存が示唆された一方，MP-Struct は強力なベースラインである k-Shuffle Dyck よりも低い値を達成した．k-Shuffle Dyck は長距離依存の記憶を要するが，その記号は構造的に交換可能であり，自然言語のような機能的な区別を持たない．対照的に MP-Struct は，機能範疇が内容語の構成を予測する非対称な抽象化を促進する．この自然言語の主要部駆動な性質に近いバイアスが，意味情報が利用できない状況下でも機能する，より質的に優れた構造処理メカニズムを形成したと考えられる．

5.2 抽象化モデルによる統制実験

効率性の要因をより切り分けるため，複雑さを調整した対照実験を行う．公平な比較のため，本研究の MP-Struct に含まれる操作タイプを厳密に模倣し，生成器は 1 種類の再帰構造（Merge）と，4 種類の機能的依存関係（Agree-Sg, Agree-Pl, Move, Selection）に基づいて系列を生成する（詳細は付録 C 参照）．本実験では，これらの要素の**配置**がもたらす影響を，**依存同定曖昧性**の観点から分析する．依存同定曖昧性とは，系列上に現れる依存関係の終点（例：`)1`）から見て，対応する始点（例：`(1`）を**一意に同定する手がかりがどれだけ与えられているか**（あるいは，どれだけ不足しているか）を表す概念である．同定の手がかりが乏しい系列では，終点に対して候補となる始点が多数残りやすく，依存同定は曖昧になる．逆に，終点の近傍や関連位置に構造的標識が配置されていれば，候補が強く絞り込まれ，依存同定は曖昧になりにくい．この観点に基づき，以下の対照的な 2 条件を定義した．

- **1. Generic k-SD:** 基本要素をランダムに混合した条件．表 3 の例 `[0 (1 (2...` のように，依存関係の始点 `(1`, `(2` が無秩序に並ぶため，終点 `)1` から正しい先行要素を見分ける手がかりが弱い．その結果，終点に対して競合する候補が残りやすく，**依存同定曖昧性は高い**．
- **2. MP-Struct Core:** MP-Struct の根幹に基づき，依存関係を階層構造に従属させた抽象化モデル．表 3 の例 `H_T (1 ...` のように，機能範疇トークン（例：`H_T`）が依存関係の直前に配置され，明示的な構造的標識として働く．この標識により終点から参照すべき領域が局所化され，候補が大きく絞られるため，**依存同定曖昧性は低い**．

表 2 下段に示すように，MP-Struct Core は MRS **16.2**，平均 Efficiency Gain **31%（最大 37%）**を達成した．[5] 着目すべき点は，これらの指標が先行研究の最良系列（k-Shuffle Dyck）だけでなく，MP-Struct をも上回っていることである．このことは，少なくとも本設定においては，派生に伴うノイズを捨象しつつ「どこを手がかりに依存を同定できるか」を系列内に明示することが，学習効率を押し上げうることを示唆する．

Hu ら [9] は，PPT 言語の有効性をチョムスキー階層（構造的表現力）と回路複雑性（Transformer の計算的制約）の交差として整理し，その枠組みのもとで k-Shuffle Dyck の効率性を示した．一方，本節の統制実験で k-Shuffle Dyck を上回る効率が観察されたことは，この整理に加えて，系列内の**構造的標識**の配置が**依存同定曖昧性**を低下させうるという性質が，学習効率の差と整合的であることを示唆する．すなわち，PPT 言語の有効性には，表現力と回路計算量だけでは捉えきれない要因が関与している可能性が示された．

6 おわりに

本稿では，MP-Struct を用いた PPT を提案し，UG 由来の構造制約を PPT として導入することで，文法能力を維持したまま学習効率を改善できることを示した．さらに抽象化モデルの統制実験から，PPT 言語の有効性をチョムスキー階層と回路複雑性で整理する既存枠組み [9] に加え，系列内の**構造的標識**の配置が**依存同定曖昧性**を低下させうることが，学習効率の差と整合的であることが示唆された．

5) 学習曲線を付録 D に示す．

謝辞

本研究は，JSPS 科研費 JP24H00087，JP24KJ0800，JST さきがけ JPMJPR21C2，JST CREST JPMJCR2565，JST BOOST JPMJBY24B2 の支援を受けたものです．

参考文献

[1] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[2] Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In **Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning**, pp. 1–34. Association for Computational Linguistics, December 2023.

[3] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In **International Conference on Learning Representations**, 2020.

[4] Noam Chomsky. **Aspects of the Theory of Syntax**. The MIT Press, Cambridge, 1965.

[5] Alexander Clark and Shalom Lappin. **Linguistic Nativism and the Poverty of the Stimulus**. Wiley-Blackwell, 2011.

[6] Edward Stabler. Derivational minimalism. Vol. 1328, pp. 68–95, 01 1996.

[7] Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. **Journal of Machine Learning Research**, Vol. 21, pp. 140:1–140:67, 2019.

[8] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, KyleO'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff ほか. Pythia: A suite for analyzing large language models across training and scaling. In **International Conference on Machine Learning**, pp. 2397–2430. PMLR, 2023.

[9] Michael Y. Hu, Jackson Petty, Chuan Shi, William Merrill, and Tal Linzen. Between circuits and Chomsky: Pre-pretraining on formal languages imparts linguistic biases. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 9691–9709, Vienna, Austria, July 2025. Association for Computational Linguistics.

[10] Noam Chomsky. On Certain Formal Properties of Grammars. **Information and Control**, Vol. 2, No. 2, pp. 137–167, 1959.

[11] Sanjeev Arora and Boaz Barak. **Computational Complexity: A Modern Approach**. Cambridge University Press, 2009.

[12] Stephen Crain and Mineharu Nakayama. Structure dependence in grammar formation. **Language**, Vol. 63, No. 3, pp. 522–543, 1987.

[13] N. Chomsky. **The Minimalist Program**. Current studies in linguistics series. MIT Press, 1995.

[14] Noam Chomsky. Minimalist inquiries: The framework. In Roger Martin, David Michaels, and Juan Uriagereka, editors, **Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik**, pp. 89–155. MIT Press, Cambridge, MA, 2000.

[15] Noam Chomsky. Derivation by phase. In **Ken Hale: A Life in Language**. The MIT Press, 04 2001.

[16] Isabel Papadimitriou and Dan Jurafsky. Learning Music Helps You Read: Using transfer to study linguistic structure in language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 6829–6839, Online, November 2020. Association for Computational Linguistics.

[17] Ryokan Ri and Yoshimasa Tsuruoka. Pretraining with artificial language: Studying transferable knowledge in language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 7302–7315, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[18] Isabel Papadimitriou and Dan Jurafsky. Injecting structural hints: Using language models to study inductive biases in language learning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 8402–8413, Singapore, December 2023. Association for Computational Linguistics.

[19] Andy Yang and David Chiang. Counting like transformers: Compiling temporal counting logic into softmax transformers. In **First Conference on Language Modeling**, 2024.

[20] Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for English. **Transactions of the Association for Computational Linguistics**, Vol. 8, pp. 377–392, 2020.

[21] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

[22] Lewis Carroll. **Through the Looking-Glass, and What Alice Found There**. Macmillan, 1871.

A　学習設定の詳細

表 4 に学習設定の詳細を示す．なお，実験には NVIDIA RTX 6000 Ada (48GB) GPU 1 枚を使用し，学習時間は各試行につき約 20 時間であった．

表 4: PPT および PT のハイパーパラメータ．

ハイパーパラメータ	値
バッチサイズ	16
勾配蓄積数	2
実質バッチサイズ	32
最大系列長	1024
学習率	5×10^{-4}
学習率スケジューラ	Cosine with warmup
最小学習率	5×10^{-5}
ウォームアップステップ数	1000
重み減衰	0.1
勾配クリッピング	1.0
最適化手法	AdamW
β_1, β_2	0.9, 0.999
ϵ	10^{-6}
混合精度	bf16

B　学習効率指標の詳細

ベースライン（Non-PPT）の自然言語学習ステップ数を y_1，提案手法における形式言語 PPT の学習ステップ数を x とし，ベースラインが y_1 で達した損失と同等の性能に提案手法が初めて到達した時点の自然言語学習ステップ数を y_2 とする．このとき，

$$\text{MRS} = \frac{y_1 - y_2}{x} \tag{1}$$

は「PPT の 1 ステップが PT の何ステップ分の学習を節約したか」を表し，値が大きいほど PPT の代替効果が高い．また，

$$\text{Efficiency Gain} = 1 - \frac{y_2 + x}{y_1} \tag{2}$$

は，同等の性能に到達するために必要な総学習ステップ（$y_2 + x$）がベースライン y_1 からどれだけ削減されたかを表し，値が大きいほど学習効率の改善が大きい．

計算例　本実験における一つの試行（Seed=0）の結果を用いた実際の計算例を以下に示す．基準点を $y_1 = 25,000$ としたとき，ベースライン（Non-PPT）の損失は約 3.633 であった．提案手法（MP-Struct）がこの損失に初めて到達したのは $y_2 \approx 15,755$ ステップ時点である．形式言語学習ステップ数は $x = 500$ であるため，各指標は以下のように計算される．

$$\text{MRS} = \frac{25,000 - 15,755}{500} = \frac{9,245}{500} = 18.49 \tag{3}$$

$$\text{Efficiency Gain} = 1 - \frac{15,755 + 500}{25,000} = 1 - 0.65 = 0.35 \tag{4}$$

C　抽象条件における複雑性の設定

§5.2 では，MP-Struct の実構成に基づき，階層構造を 1 種類の括弧（1-Dyck），機能的依存を 4 種類の括弧（4-Shuffle Dyck）でモデル化した．この選択は恣意的なものではなく，MP-Struct 生成器に含まれる依存関係タイプの分析に基づいている．Algorithm 1 および表 1 に要約されるように，MP-Struct は以下の 5 つの異なる構造操作に基づいて系列を生成する．

1. **構造 (Merge):** 括弧（例：[...]）による再帰的骨格．**1-Dyck** に対応．
2. **依存 (Move/Agree/Select):** 骨格内部の 4 種の長距離依存．**4-Shuffle Dyck** に対応：
   (a) **一致（複数）：** T_{pl} と DP_{pl} 間の依存．
   (b) **一致（単数）：** T_{sg} と DP_{sg} 間の依存．
   (c) **移動:** 機能範疇（例：C）と痕跡（TR）間の依存．
   (d) **選択:** 限定詞（D）が名詞（N）を選択する局所依存．

実モデル（MP-Struct）の構成に基づく $k = 1+4$ を採用することで，**Generic k-SD** と **MP-Struct Core** の語彙サイズ（括弧の種類数）および回路複雑性は等価に保たれる．これにより，表現力に起因する交絡因子を排除し，構造的配置（および構造的標識の有無）がもたらす依存同定曖昧性の差異を，純粋に分離して評価することが可能となる．

D　学習曲線

図 3 に学習曲線（訓練損失の軌跡）を示す．

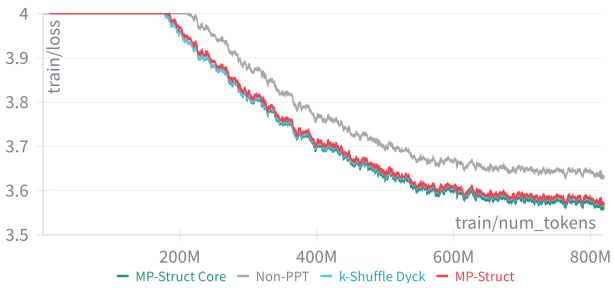

図 3: 訓練損失の軌跡．