Transformer の音楽生成への応用例/ LLM の原理解明に向けて

稲葉 達郎

D1 @ Tohoku Univ. / RA @ NII

2025/6/2, 九州大学

* 共同最終著者

自己紹介

経歴

2019/4~2023/3　京都大学工学部電気電子工学科
2023/4~2025/3　京都大学情報学研究科知能情報学コース 修士課程
2025/4~　東北大学情報科学研究科 博士課程

研究キーワード

- 深層学習モデルの原理解明
  - 学習ダイナミクス解析
  - 構造表現能力の理解
- 音楽モデル ⇄ 言語モデル への応用

趣味

- 音楽 (作曲/ギター/ピアノ)
- ゲーム、サッカー

自己紹介

経歴

2019/4~2023/3　京都大学工学部電気電子工学科
2023/4~2025/3　京都大学情報学研究科知能情報学コース 修士課程
2025/4~　東北大学情報科学研究科 博士課程

研究キーワード

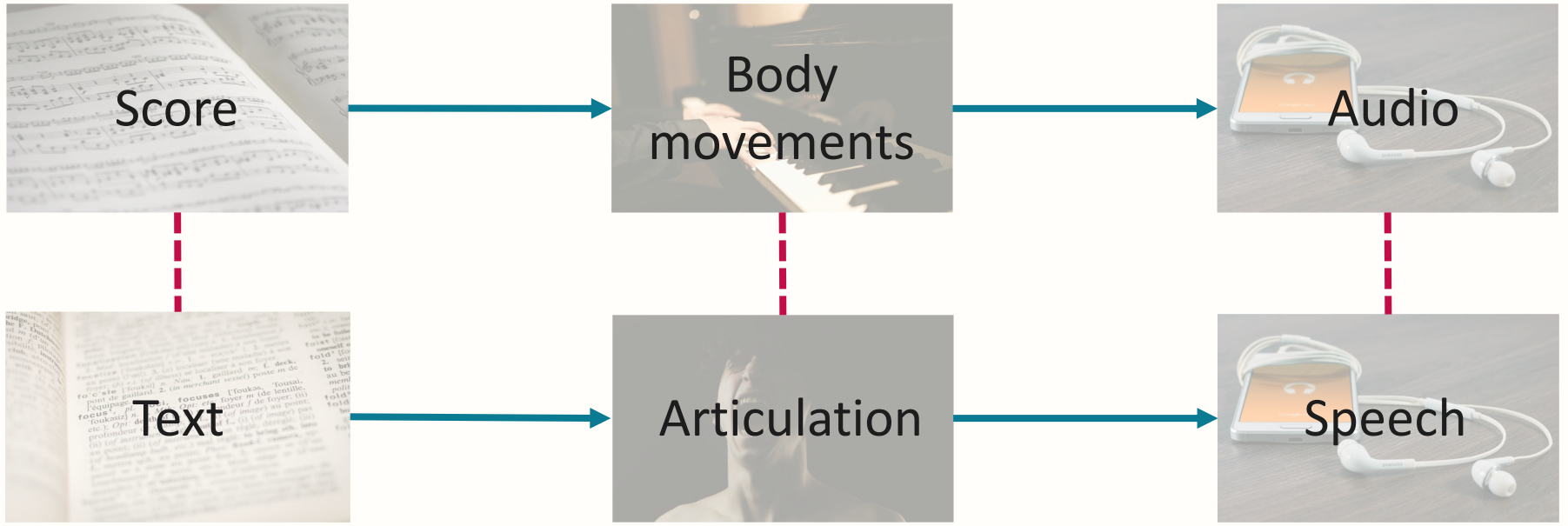
- 深層学習モデルの原理解明
  - 学習ダイナミクス解析
  - 構造表現能力の理解
- **音楽モデル ⇄ 言語モデル への応用**

趣味

- 音楽 (作曲/ギター/ピアノ)
- ゲーム、サッカー

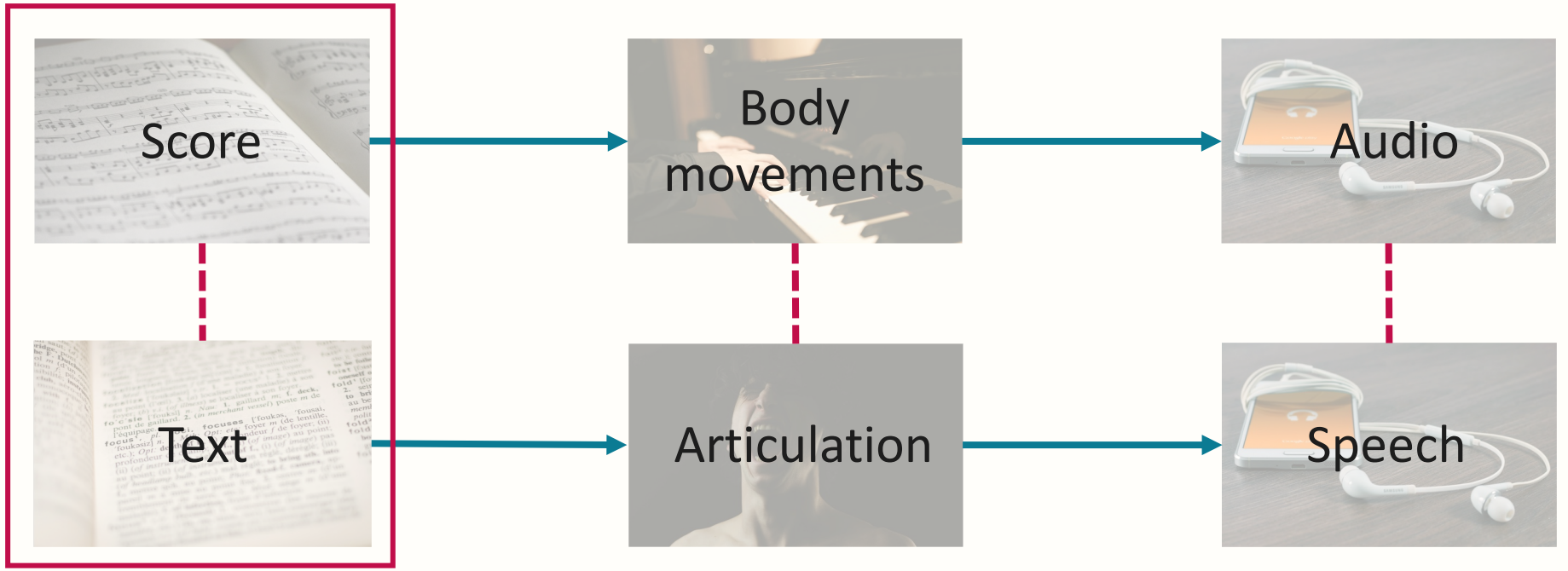
音楽と言語の対応

- 音楽における「楽譜→身体動作→オーディオ」という過程は言語における「テキスト→発話→音声」という過程と非常に酷似
- タスクや手法も対応
  - 楽譜生成 ⇔ テキスト生成、自動採譜 (Audio-to-Score) ⇔ 音声認識 (Speech-to-Text)、楽器音合成 (Score-to-Audio) ⇔ 音声合成 (Text-to-Speech)、楽譜分類 ⇔ テキスト分類



音楽と言語の対応

- 音楽における「楽譜→身体動作→オーディオ」という過程は言語における「テキスト→発話→音声」という過程と非常に酷似
- タスクや手法も対応
  - 楽譜生成 ⇔ テキスト生成、自動採譜 (Audio-to-Score) ⇔ 音声認識 (Speech-to-Text)、楽器音合成 (Score-to-Audio) ⇔ 音声合成 (Text-to-Speech)、楽譜分類 ⇔ テキスト分類



音楽と言語の構造

- 対応関係の他にも構造上の共通点が多くある。一方で目的の違い等から生まれた相違点もある
- 言語 ⇔ 音楽へ手法転用する際にはこれらの相違点を計算機上でどう扱うかが重要

	音楽	言語	備考
最小単位	音 (Pitch)	音素 (Phoneme)	
意味を持つ最小単位	2音以上	形態素から	
構造の組み方	調性・和声・リズム / 音楽理論	文法 (Syntax)	音楽を説明する時に参照される理論はあるが、言語ほど絶対的な規則ではない
階層構造	音→和音(or メロディー)→フレーズ→楽曲	音素→形態素→単語→句→分→文章 (談話)	言語が一次元の階層構造を持つのに対して、音楽は時間・ピッチ・楽器方向と多次元な階層構造をもつ
目的	感情表現	情報伝達	

音楽と言語に関する研究

[Patel+, 08] Music, language, and the brain.

- 音楽と言語を構造処理する時の脳内メカニズムに**共通点**がある
  - 言語の文法違反 (e.g., The pizza was in the eaten) と音楽の和声違反 (e.g., 不協和音) に対して脳は似た処理を行う
  - 主にブローカ野が関与
  - ただし完全に同じではなく、リソースを**部分的**に共有
    - 言語の意味処理や、音楽の感情的な解釈などは、異なる神経メカニズムによって処理される

[Papadimitriou+, 20] Learning Music Helps You Read: Using Transfer to Study Linguistic Structure in Language Models

- 構造を持つ非言語データ (音楽やコード等) での事前学習は、自然言語の文法理解を向上させる

[Huang+, 22] MuLan: A Joint Embedding of Music Audio and Natural Language

- オーディオ音楽と自然言語のタグを対照学習により対応づける

音楽生成における相対性・循環性の重要性

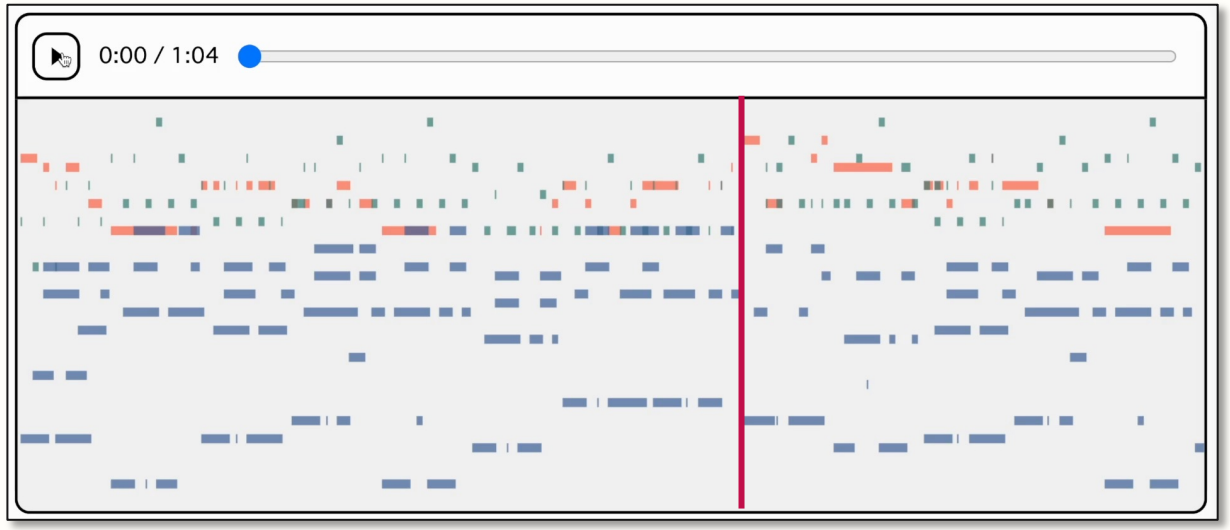
稲葉 達郎[1], 吉井 和佳[2], 中村 栄太[3]

[1] 東北大学 [2] 京都大学 [3] 九州大学

音楽モデル ← 言語モデルへの応用

研究概要

- 記号音楽生成において Transformer の能力を最大限引き出す手法の模索
- 音符間の時刻と音高の**相対距離**を、小節単位とオクターブ単位の**循環性**を考慮したエンコーディングにより自己注意機構に組み込んだ
- 音楽特有の二次元的な**繰り返し構造**を効果的に捉え、高い一貫性を持つ音楽生成が可能

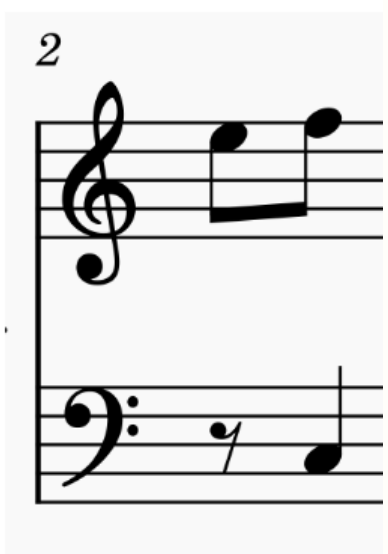


提案手法の生成音楽例

16秒までの4小節をプロンプトとして与え，
その続き12小節を生成させている

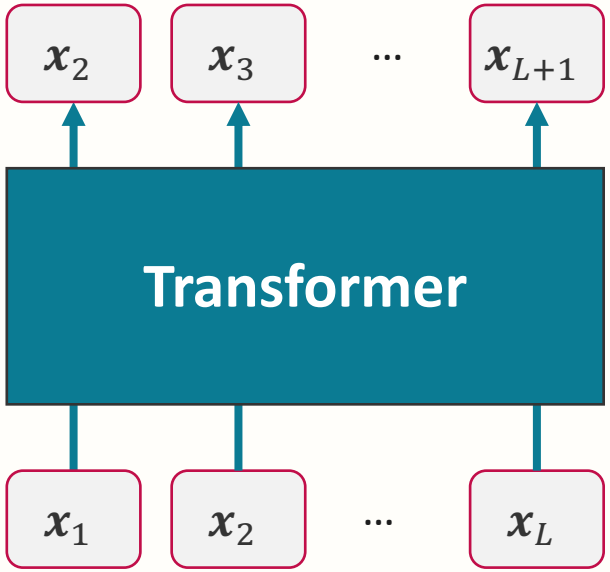
Transformer + 音楽生成

- 音楽を Transformer が処理できる**トークン列**に変換する必要がある
  - 音符単位表現
  - イベント単位表現



トークン化

$\{x_1, x_2, \ldots, x_L\}$

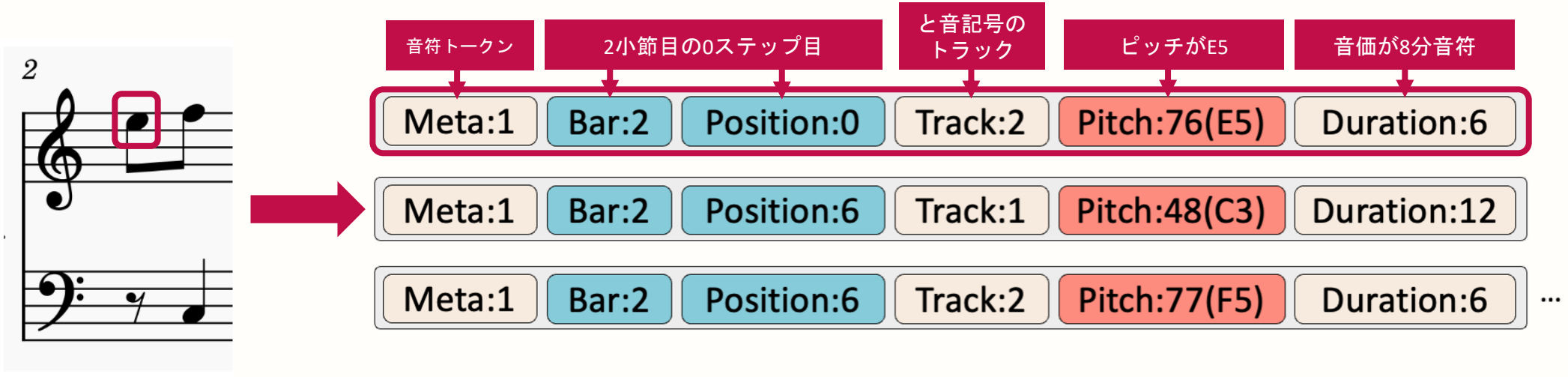


Note-based Rep. [Zeng+'21; Dong+'23]

- 基本一つのトークンが一つの音符を表し，各トークン（音符）は6つの変数の組で表現

$$\mathbf{x}_i = \{x_i^{\text{meta}}, x_i^{\text{bar}}, x_i^{\text{position}}, x_i^{\text{track}}, x_i^{\text{pitch}}, x_i^{\text{duration}}\}$$

- エンコードの際には各変数を別々に線形変換層により変換して和を取る
- デコードの際には出力の潜在表現に対し6種類の線形変換層により各変数を別々にデコード
- 各トークン種類ごとに辞書を持つ（語彙サイズは順に 3, 16, 48, 3, 128, 27）



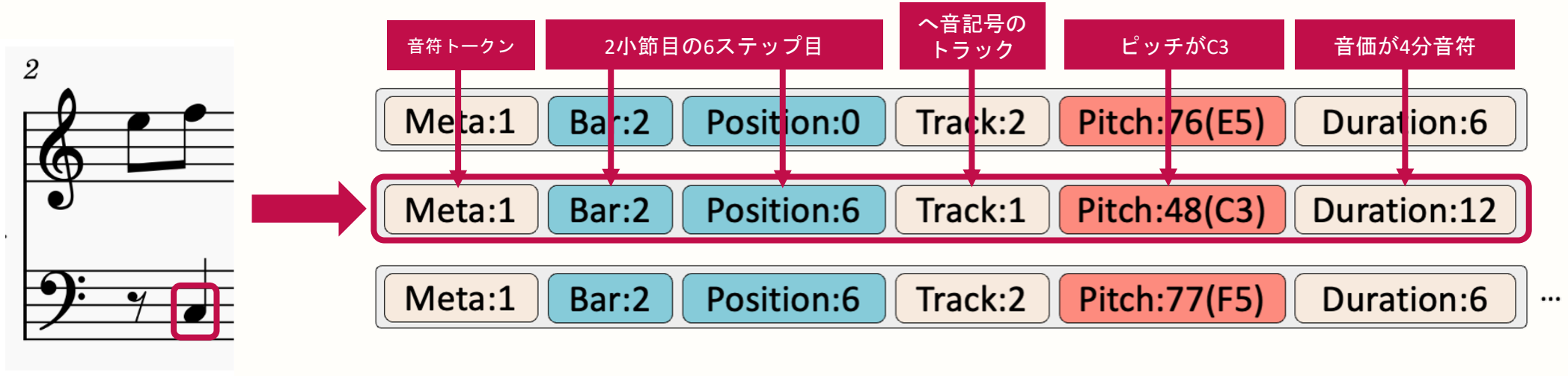
Zeng, M., Tan, X., Wang, R., Ju, Z., Qin, T. and Liu, T.-Y.: MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training
Dong, H.-W., Chen, K., Dubnov, S., McAuley, J. and Berg-Kirkpatrick, T.: Multitrack Music Transformer

Note-based Rep. [Zeng+'21; Dong+'23]

- 基本一つのトークンが一つの音符を表し，各トークン（音符）は6つの変数の組で表現

$$\mathbf{x}_i = \{x_i^{\text{meta}}, x_i^{\text{bar}}, x_i^{\text{position}}, x_i^{\text{track}}, x_i^{\text{pitch}}, x_i^{\text{duration}}\}$$

- エンコードの際には各変数を別々に線形変換層により変換して和を取る
- デコードの際には出力の潜在表現に対し6種類の線形変換層により各変数を別々にデコード
- 各トークン種類ごとに辞書を持つ（語彙サイズは順に 3, 16, 48, 3, 128, 27）



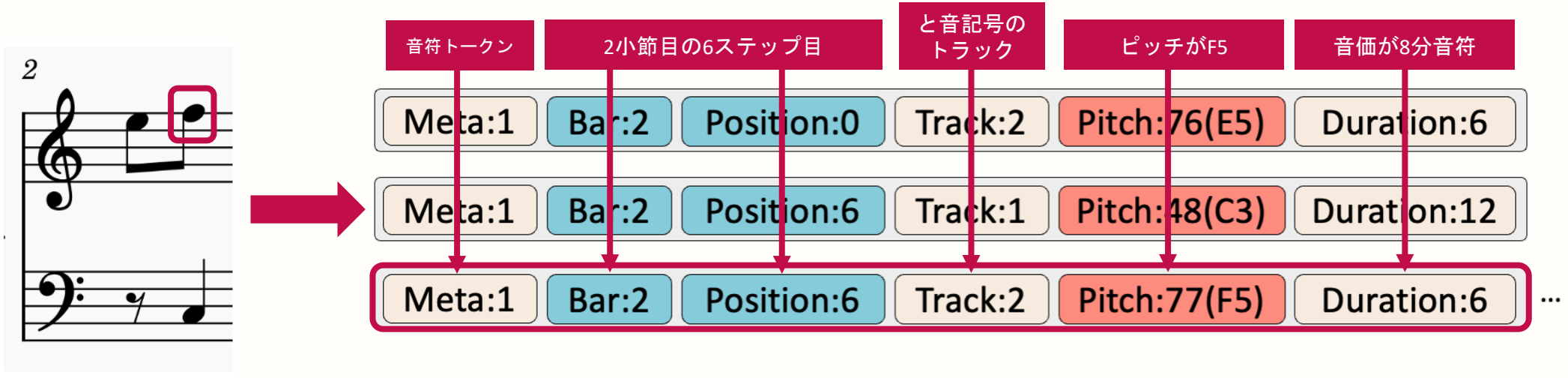
Zeng, M., Tan, X., Wang, R., Ju, Z., Qin, T. and Liu, T.-Y.: MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training
Dong, H.-W., Chen, K., Dubnov, S., McAuley, J. and Berg-Kirkpatrick, T.: Multitrack Music Transformer

Note-based Rep. [Zeng+'21; Dong+'23]

- 基本一つのトークンが一つの音符を表し，各トークン（音符）は6つの変数の組で表現

$$\mathbf{x}_i = \{x_i^{\text{meta}}, x_i^{\text{bar}}, x_i^{\text{position}}, x_i^{\text{track}}, x_i^{\text{pitch}}, x_i^{\text{duration}}\}$$

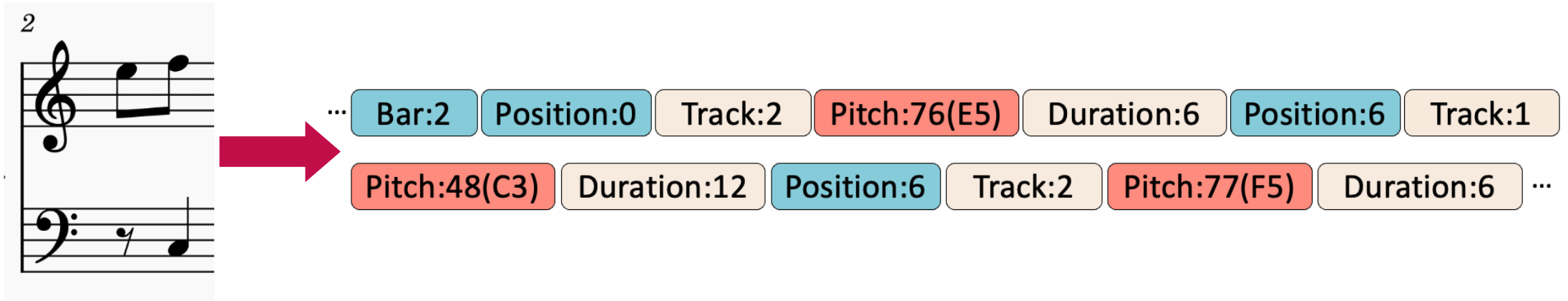
- エンコードの際には各変数を別々に線形変換層により変換して和を取る
- デコードの際には出力の潜在表現に対し6種類の線形変換層により各変数を別々にデコード
- 各トークン種類ごとに辞書を持つ（語彙サイズは順に 3, 16, 48, 3, 128, 27）



Zeng, M., Tan, X., Wang, R., Ju, Z., Qin, T. and Liu, T.-Y.: MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training
Dong, H.-W., Chen, K., Dubnov, S., McAuley, J. and Berg-Kirkpatrick, T.: Multitrack Music Transformer

Event-based Rep. [Oore+'18; Huang+'20]

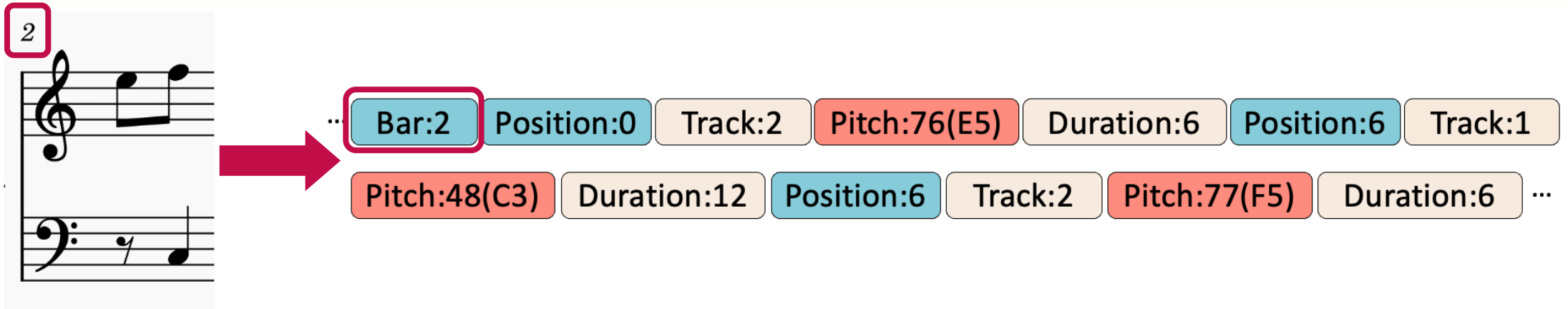
- 時刻と音高でソートした音符を順に Position, Track, Pitch, Duration の 4 トークンで表現し、これを順にトークン系列に追加
- 小節が変わった際には Bar トークンを追加
- トークン列の最初と最後にそれぞれ BOS トークンと EOS トークン
- 辞書は一つのみ（語彙サイズは 2+16+48+3+128+27=224）



Oore, S., Simon, I., Dieleman, S., Eck, D. and Simonyan, K.: This time with feeling: learning expressive musical performance
Huang, Y.-S. and Yang, Y.-H.: Pop Music Transformer: Beat-Based Modeling and Generation of Expressive Pop Piano Compositions

Event-based Rep. [Oore+'18; Huang+'20]

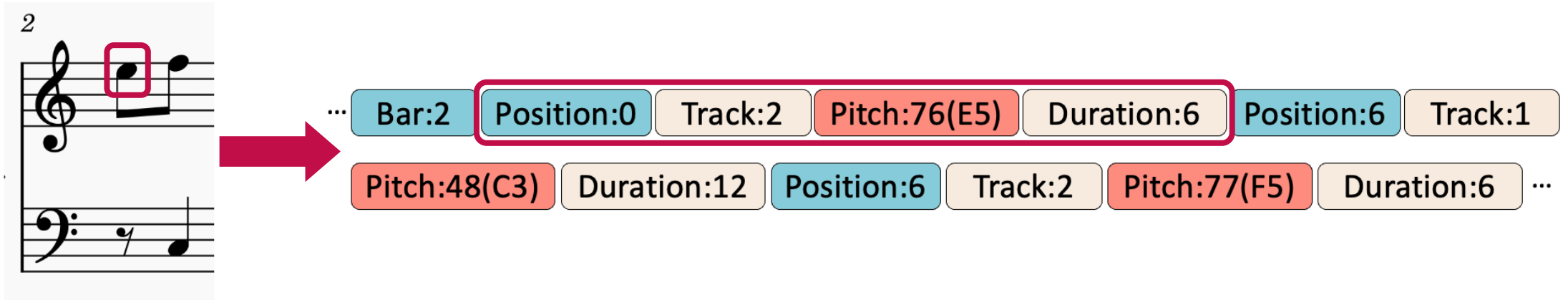
- 時刻と音高でソートした音符を順に Position, Track, Pitch, Duration の 4 トークンで表現し、これを順にトークン系列に追加
- 小節が変わった際には Bar トークンを追加
- トークン列の最初と最後にそれぞれ BOS トークンと EOS トークン
- 辞書は一つのみ（語彙サイズは 2+16+48+3+128+27=224）



Oore, S., Simon, I., Dieleman, S., Eck, D. and Simonyan, K.: This time with feeling: learning expressive musical performance
Huang, Y.-S. and Yang, Y.-H.: Pop Music Transformer: Beat-Based Modeling and Generation of Expressive Pop Piano Compositions

Event-based Rep. [Oore+'18; Huang+'20]

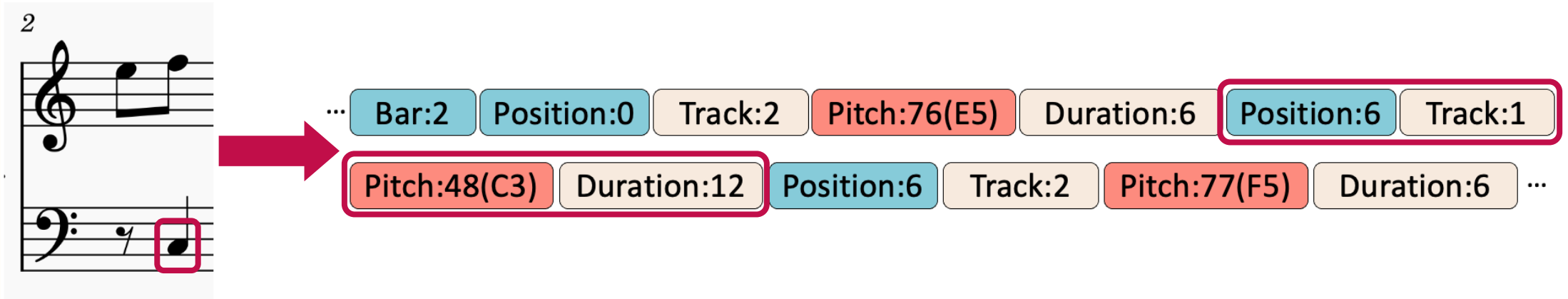
- 時刻と音高でソートした音符を順に Position, Track, Pitch, Duration の 4 トークンで表現し、これを順にトークン系列に追加
- 小節が変わった際には Bar トークンを追加
- トークン列の最初と最後にそれぞれ BOS トークンと EOS トークン
- 辞書は一つのみ（語彙サイズは 2+16+48+3+128+27=224）



Oore, S., Simon, I., Dieleman, S., Eck, D. and Simonyan, K.: This time with feeling: learning expressive musical performance
Huang, Y.-S. and Yang, Y.-H.: Pop Music Transformer: Beat-Based Modeling and Generation of Expressive Pop Piano Compositions

Event-based Rep. [Oore+'18; Huang+'20]

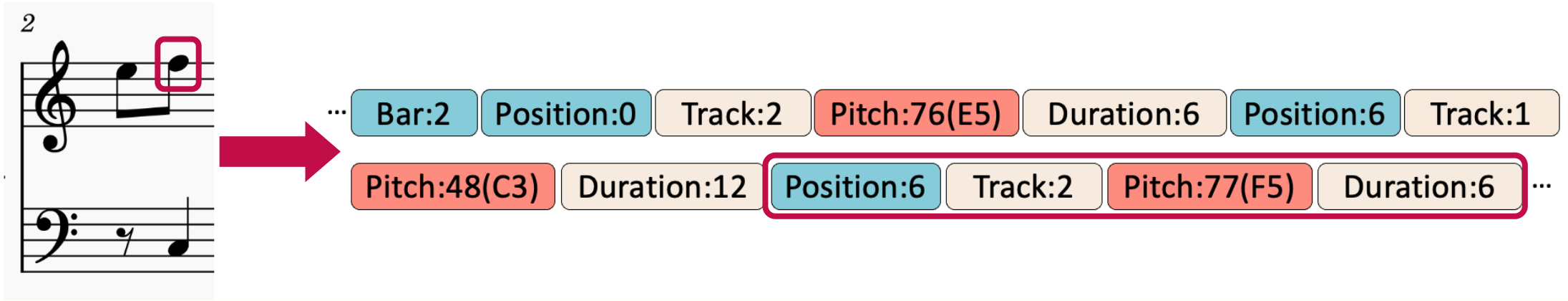
- 時刻と音高でソートした音符を順に Position, Track, Pitch, Duration の 4 トークンで表現し、これを順にトークン系列に追加
- 小節が変わった際には Bar トークンを追加
- トークン列の最初と最後にそれぞれ BOS トークンと EOS トークン
- 辞書は一つのみ（語彙サイズは 2+16+48+3+128+27=224）



Oore, S., Simon, I., Dieleman, S., Eck, D. and Simonyan, K.: This time with feeling: learning expressive musical performance
Huang, Y.-S. and Yang, Y.-H.: Pop Music Transformer: Beat-Based Modeling and Generation of Expressive Pop Piano Compositions

Event-based Rep. [Oore+'18; Huang+'20]

- 時刻と音高でソートした音符を順に Position, Track, Pitch, Duration の 4 トークンで表現し、これを順にトークン系列に追加
- 小節が変わった際には Bar トークンを追加
- トークン列の最初と最後にそれぞれ BOS トークンと EOS トークン
- 辞書は一つのみ（語彙サイズは 2+16+48+3+128+27=224）



Oore, S., Simon, I., Dieleman, S., Eck, D. and Simonyan, K.: This time with feeling: learning expressive musical performance
Huang, Y.-S. and Yang, Y.-H.: Pop Music Transformer: Beat-Based Modeling and Generation of Expressive Pop Piano Compositions

音楽における相対性・循環性

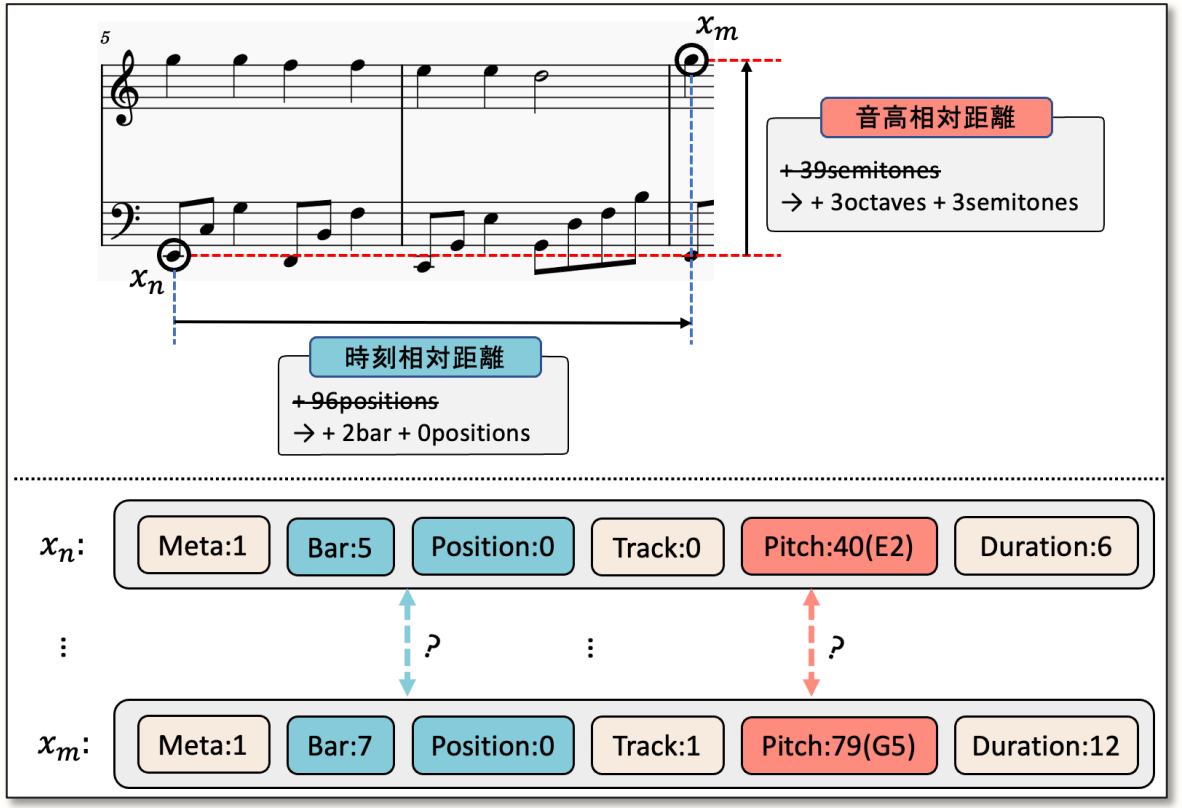
- 相対性: 音符間の時刻と音高の**相対性**は重要
  - 時刻: 時刻シフトした音楽は同じ音楽と認識可能
  - 音高: 移調された音楽も同じ音楽と認識可能

- 循環性: 各方向における**循環性**も音楽構造上重要
  - 時刻: 特定小節 (1,2,4,8) 間隔ごとに似たテーマが演奏される音楽は多い
  - 音高: 1オクターブ高い音楽は元の音の2倍の振動数を持ち、似た響きを持つ

問題点: 相対性・循環性の欠落

- 音符の絶対的時刻・音高の情報は含まれている一方、音符間の**相対距離**とその相対距離に内在する**循環性**が表現できていない

- 右図の音符単位表現の例において以下の情報は表現できていない
  - x_n と x_m の時刻相対距離が 96 positions (i.e., 2小節)
  - x_n と x_m の音高相対距離が 39 半音 (i.e., 3 オクターブと 3 半音)



Oore, S., Simon, I., Dieleman, S., Eck, D. and Simonyan, K.: This time with feeling: learning expressive musical performance
Huang, Y.-S. and Yang, Y.-H.: Pop Music Transformer: Beat-Based Modeling and Generation of Expressive Pop Piano Compositions

Attention [Vaswani+, 17]

- Query と Key から Attnetion Map を計算し Value の重み付き和を出力

$$\mathrm{Attn}(X) = \mathrm{Softmax}\left(\frac{QK^{\top}}{\sqrt{D_h}}\right)V$$

- ただし、$Q = XW_Q, K = XW_K, V = XW_V$ で、$W_Q, W_K, W_V \in \mathbb{R}^{D \times D_h}$ は学習可能行列

Relative Attention [Huang+, 18]

- Attention の計算に**インデックスの相対位置情報**を取り込む

$$\mathrm{RelAttn}(X) = \mathrm{Softmax}\left(\frac{QK^{\top} + S_{\mathrm{rel}}^{\mathrm{idx}}}{\sqrt{D_h}}\right)V$$

- ただし、$S_{\mathrm{rel}}^{\mathrm{idx}} = Q\mathbf{R}_{\mathrm{idx}}^{\top} = Q\ \mathrm{LPE}(\mathbf{I}^{\top}\cdot\mathbf{1} - \mathbf{1}\cdot\mathbf{I}^{\top})^{\top}$ で、$\mathbf{I} = \{1,2,\ldots,L\}$、LPEは学習可能埋め込みを表す。
  - $(\mathbf{I}^{\top}\cdot\mathbf{1} - \mathbf{1}\cdot\mathbf{I}^{\top})$ はインデックス間の相対距離表す行列
  - その埋め込みを Q と掛けることにより、QK^Tと同じ $\mathbb{R}^{L\times L}$ の形に変換

RIPO Attention [Guo+, 22]

- **音符間の相対時間・相対ピッチの情報**をさらに取り込む

$$\mathrm{RIPOAttn}(X) = \mathrm{Softmax}\left(\frac{QK^\top + S_{\mathrm{rel}}^{\mathrm{idx}} + S_{\mathrm{rel}}^{\mathrm{t}} + S_{\mathrm{rel}}^{\mathrm{p}}}{\sqrt{D_h}}\right)V$$

- ただし、$S_{\mathrm{rel}}^{\mathrm{t}} = Q\mathbf{R}_{\mathrm{t}}^\top = Q\,\mathrm{SPE}(\mathbf{T}^\top \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{T})^\top, S_{\mathrm{rel}}^{\mathrm{p}} = Q\mathbf{R}_{\mathrm{p}}^\top = Q\,\mathrm{SPE}(\mathbf{P}^\top \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{P})^\top$ で、$\mathbf{T}$ と $\mathbf{P}$ は以下のようになる

Meta:1	Bar:2	Position:0	Track:2	Pitch:76(E5)	Duration:6
Meta:1	Bar:2	Position:6	Track:1	Pitch:48(C3)	Duration:12
Meta:1	Bar:2	Position:6	Track:2	Pitch:77(F5)	Duration:6

…

$$\mathbf{P} = \{76, 48, 77, \dots\}$$

$$\mathbf{T} = \{48, 54, 54, \dots\}$$

Z. Guo, J. Kang, and D. Herremans, “A domain- knowledge-inspired music embedding space and a novel attention mechanism for symbolic music modeling,”

RIPO Attention [Guo+, 22]

- **音符間の相対時間・相対ピッチの情報**をさらに取り込む

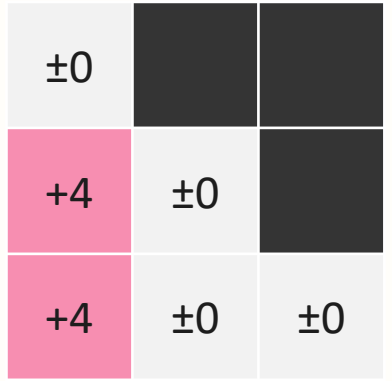
$$\mathrm{RIPOAttn}(X) = \mathrm{Softmax}\left(\frac{QK^\top + S^{\mathrm{idx}}_{\mathrm{rel}} + S^{\mathrm{t}}_{\mathrm{rel}} + S^{\mathrm{p}}_{\mathrm{rel}}}{\sqrt{D_h}}\right)V$$

- ただし、$S^{\mathrm{t}}_{\mathrm{rel}} = Q\mathbf{R}^\top_{\mathrm{t}} = Q\,\mathrm{SPE}(\mathbf{T}^\top \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{T})^\top, S^{\mathrm{p}}_{\mathrm{rel}} = Q\mathbf{R}^\top_{\mathrm{p}} = Q\,\mathrm{SPE}(\mathbf{P}^\top \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{P})^\top$ で、$\mathbf{T}$ と $\mathbf{P}$ は以下のようになる

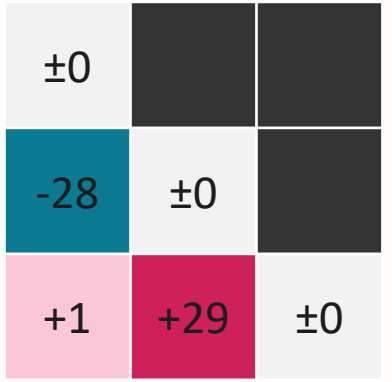
$$\mathbf{T} = \{48, 54, 54, \ldots\}$$

$$\mathbf{P} = \{76, 48, 77, \ldots\}$$

相対化



$\mathbf{T}^\top \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{T}$



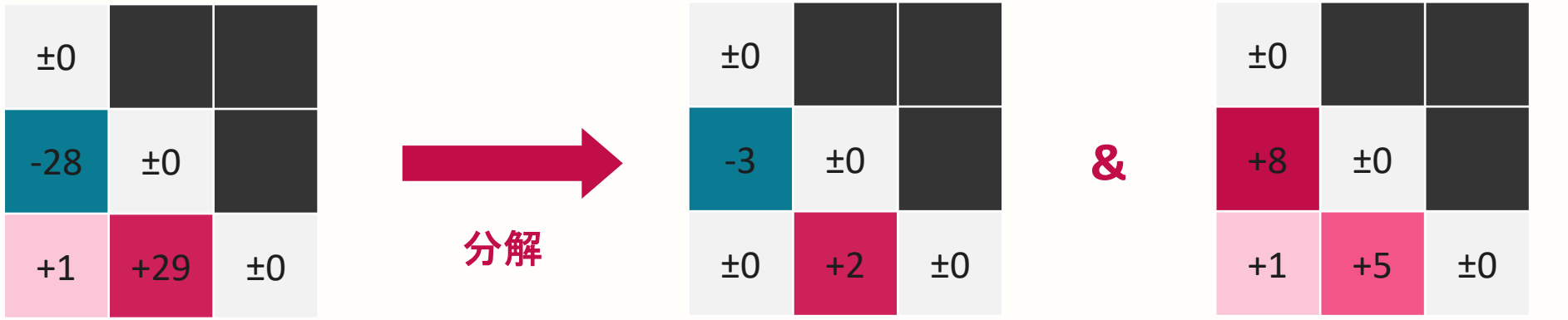
$\mathbf{P}^\top \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{P}$

提案手法: Circular Relative Attention

- 音符間の相対時間・相対ピッチの情報を**循環性を考慮して**取り込む

$$\mathrm{CirRelAttn}(X) = \mathrm{Softmax}\left(\frac{QK^{\top} + S^{\mathrm{idx}}_{\mathrm{rel}} + S^{\mathrm{t}}_{\mathrm{rel}} + S^{\mathrm{p}}_{\mathrm{rel}}}{\sqrt{D_h}}\right)V$$

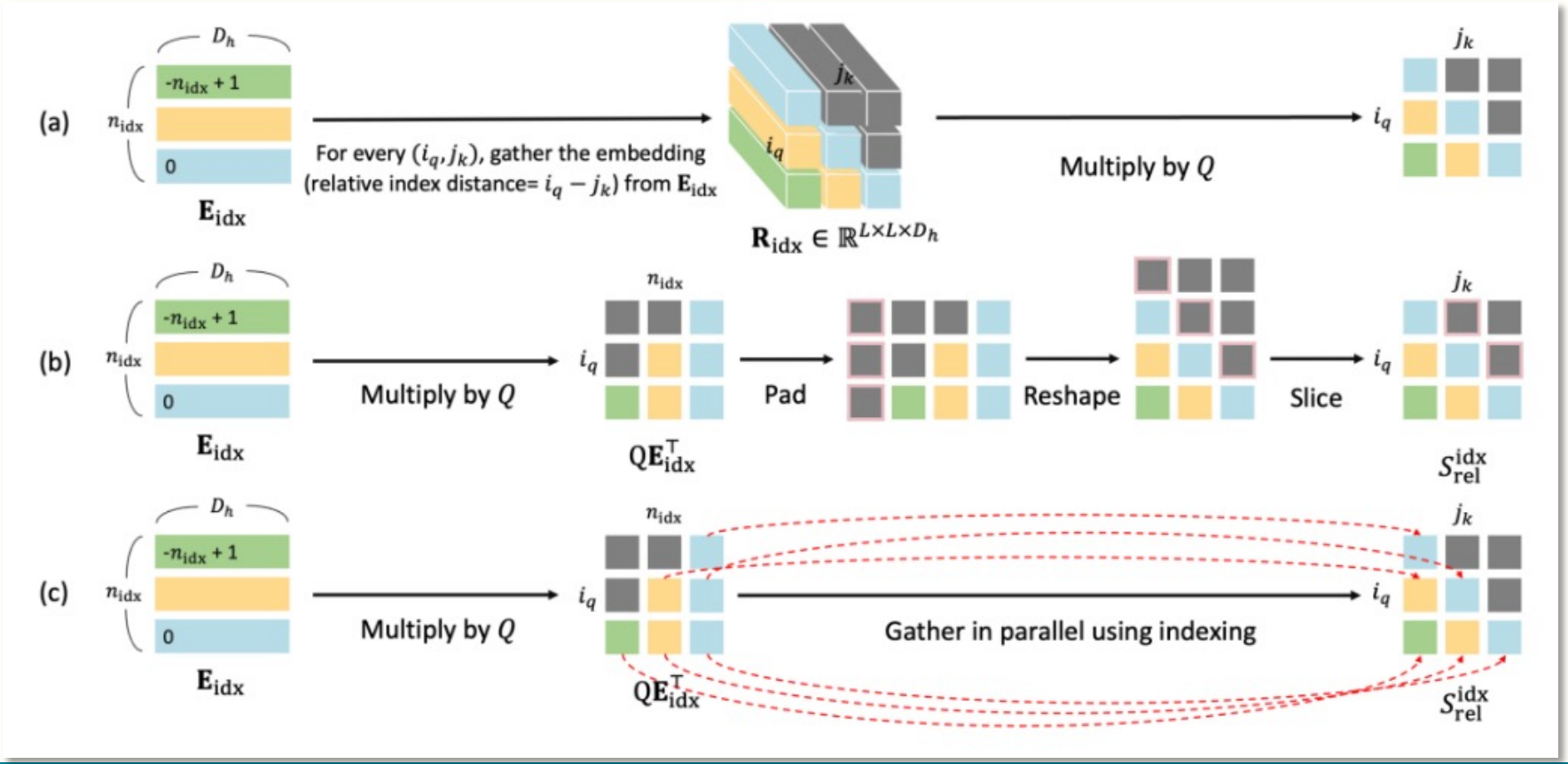
- ただし、 $S^{\mathrm{t}}_{\mathrm{rel}} = Q\left(\mathbf{R}_{\mathrm{bar}} \odot \mathbf{R}_{\mathrm{position}}\right)^{\top}, S^{\mathrm{p}}_{\mathrm{rel}} = Q\left(\mathbf{R}_{\mathrm{octave}} \odot \mathbf{R}_{\mathrm{semitone}}\right)^{\top}$
  - $\mathbf{R}_{\mathrm{bar}} = \mathrm{LPE}\left((\mathbf{T}^{\top} \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{T}) \,//\, \mathrm{BarRes}\right),\ \mathbf{R}_{\mathrm{position}} = \mathrm{LPE}((\mathbf{T}^{\top} \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{T}) \,\%\, \mathrm{BarRes})$
  - $\mathbf{R}_{\mathrm{octave}} = \mathrm{LPE}\left((\mathbf{P}^{\top} \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{P}) \,//\, 12\right),\ \mathbf{R}_{\mathrm{semitone}} = \mathrm{LPE}((\mathbf{P}^{\top} \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{P}) \,\%\, 12)$



$\mathbf{P}^{\top} \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{P}$ $\quad (\mathbf{P}^{\top} \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{P})//12$ $\quad (\mathbf{P}^{\top} \cdot \mathbf{1} - \mathbf{1} \cdot \mathbf{P})\%12$

提案手法: Circular Relative Attention

- 実装の際には各 $\mathbf{R} \in \mathbb{R}^{\mathrm{L}\times\mathrm{L}\times\mathrm{D_h}}$ を直接計算することを避けることで、メモリ効率を $O(L^2 D_h)$ から $O(LD_h)$ に削減



← **Naive Approach**
- $O(L^2 D_h)$

← **Skew Algorithm** [Huang+, 18]
- $O(LD_h)$
- 相対インデックス距離にのみ適用可

← **Index-based Algorithm**
- $O(LD_h)$
- 相対時間距離・相対ピッチ距離にも適用可

実験設定

モデル: 4層 の Decoder 型 Transformer

データセット: POP909 [Wang+, 20]

- 繰り返し構造が豊富に含まれるポピュラー音楽のデータセット
- 16小節となるようにストライド1小節でデータ作成
- 学習データ: 35,452件，検証データ: 4,749件，テストデータ: 4,137件

ベースライン: Attention(Attn)・Relative Attention(RelAttn)・RIPO Attention(RIPOAttn)

主観評価

後続生成タスク

- 4小節をモデルに与え，続きの12小節を生成させる
- 各サンプルセットごとに3人が評価
- 10のサンプルセットで評価

評価項目(それぞれ1~5の5段階評価)

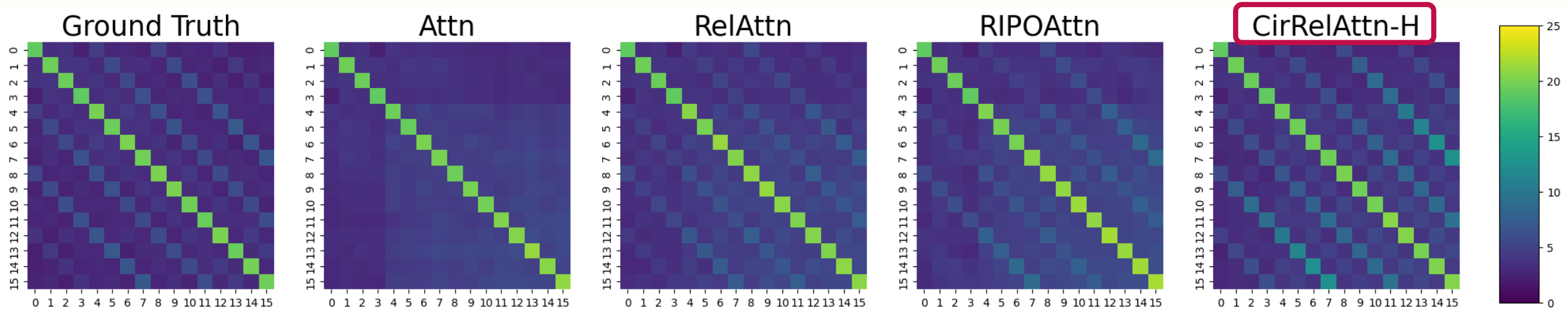
- Coherence・Musicality・Overall

- **Event-based Rep. + 提案手法は Coherence が特に高い → なぜ？**

Methods		Subjective Evaluation		
Rep.	Attn Type	Coherence	Musicality	Overall
Ground Truth		3.93	4.17	4.03
event	RelAttn [13]	2.93	2.69	2.79
	RIPOAttn [7]	3.00	3.21	3.03
	CirRelAttn-H	**4.31**	**3.41**	**3.69**
note	RelAttn [13]	2.28	2.69	2.45
	RIPOAttn [7]	**2.69**	**2.90**	**2.90**
	CirRelAttn-H	2.10	2.51	2.42

分析: 繰り返し構造

- Event-based Rep. + 提案手法では **4, 8, 12 小節単位で繰り返し構造**が多く含まれる
  - 下図は、Event-based Rep. + 各手法で生成した音楽の小説単位での類似度行列
  - 提案手法の CirRelAttn-H では斜めの縞が 4, 8, 12 小節間隔で濃く見える
- 「繰り返し構造が多く含まれること」が一貫性が高いと評価された一因と考えられる



客観評価

- 15小節を与え続きの1小節を生成させ、それがテストデータとどれだけ似ているか
- 提案手法がほとんどの指標において既存手法を上回る

音符単位でどれだけ一致しているか

ピアノロール上でどれだけ一致しているか

リズムの類似度

クロマベクトルの類似度

音高範囲の類似度

Methods					Objective Evaluation					
No.	Rep.	Attn Type	params	time(ms/note)	Loss	$F1_{note}$	$F1_{pr}$	GS	CS	PRS
1		Attn [1]	4.32M	8.46	0.979	0.174	0.239	0.846	0.620	0.955
2		RelAttn [13]	4.84M	9.73	0.937	0.218	0.291	0.848	0.650	0.955
3	event	RIPOAttn [7]	4.85M	18.3	0.904	0.233	0.305	0.855	0.660	0.956
4		CirRelAttn-S	4.88M	22.5	0.876	0.268	0.341	0.855	0.673	0.957
5		CirRelAttn-H	4.88M	20.8	**0.856**	**0.293**	**0.361**	**0.858**	**0.685**	**0.958**
6		Attn [1]	3.53M	4.29	4.42	0.188	0.267	0.784	0.619	0.941
7		RelAttn [13]	3.66M	5.79	4.38	0.186	0.271	**0.798**	0.628	**0.946**
8	note	RIPOAttn [7]	3.67M	5.93	4.40	0.180	0.257	0.786	0.612	0.942
9		CirRelAttn-S	3.70M	8.58	4.37	0.192	0.273	0.785	0.623	0.943
10		CirRelAttn-H	3.70M	6.54	**4.29**	**0.215**	**0.294**	0.787	**0.632**	0.944

データ量の影響

- 学習データ量を減らした実験設定でも客観評価を行った

結果

- データ量に左右されず、提案手法は有効
- データ量を 1/8 にしてもベースライン手法と同等以上の性能を発揮

Methods		Objective Evaluation					
Attn Type	Data Size	Loss	$F1_{note}$	$F1_{pr}$	GS	CS	PRS
Attn	1/1	4.42	0.188	0.267	0.784	0.619	0.941
RelAttn		4.38	0.186	0.271	**0.798**	0.628	**0.946**
RIPOAttn		4.40	0.180	0.257	0.786	0.612	0.942
CirRelAttn-H		**4.29**	**0.215**	**0.294**	0.787	**0.632**	0.944
Attn	1/2	4.44	0.180	0.261	0.775	0.618	0.940
RelAttn		4.41	0.194	0.276	**0.795**	0.630	**0.947**
RIPOAttn		4.42	0.176	0.255	0.778	0.610	0.938
CirRelAttn-H		**4.33**	**0.217**	**0.295**	**0.795**	**0.631**	0.944
Attn	1/4	4.51	0.179	0.261	0.775	0.615	0.939
RelAttn		4.46	0.185	0.267	**0.795**	0.625	0.943
RIPOAttn		4.48	0.177	0.255	0.778	0.610	0.938
CirRelAttn-H		**4.39**	**0.214**	**0.289**	0.791	**0.631**	**0.944**
Attn	1/8	4.61	0.179	0.260	0.779	0.612	0.941
RelAttn		4.58	0.172	0.255	**0.791**	0.619	**0.944**
RIPOAttn		4.58	0.167	0.247	0.775	0.600	0.936
CirRelAttn-H		**4.48**	**0.200**	**0.276**	0.790	**0.620**	0.942

まとめ

- 時刻と音高の相対距離をそれぞれ**小節とオクターブを基に分解**し、自己注意機構の計算に組み込む手法を提案
  - メモリ効率の良い実装も提案
- 相対距離を分解して扱うことで、モデルは**音楽構造をより効果的に学習**し、テストデータをより高い精度で予測できることを客観評価から確認
  - 異なる学習データ量・モデルサイズ・ハイパーパラメータでも実験・評価
- 提案法が**繰り返し構造を多く含む一貫性の高い音楽を生成**することもリスニングテストによる主観評価と小節単位の類似度から確認

How a Bilingual LM Becomes Bilingual: Tracing Internal Representations with Sparse Autoencoders

稲葉 達郎[1,2] 鴨田 豪[3,4] 乾 健太郎[5,1,6] 磯沼 大[2,1,8] 宮尾 祐介[8,1] 大関 洋平[8]
Benjamin Heinzerling[6,1] 高木 優[7]

[1] 東北大学 [2] LLMC [3] NINJAL [4] 総研大 [5] MBZUAI [6] 理研 [7] 東大 [8] 名工大

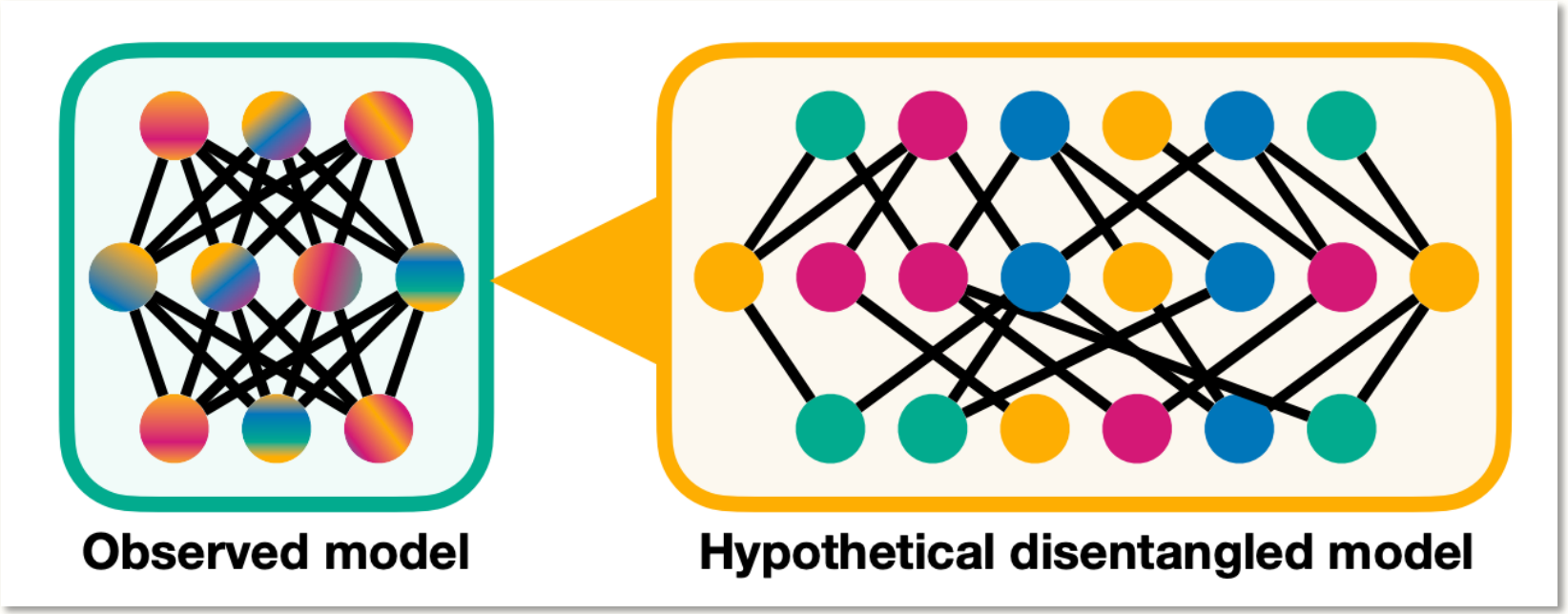
- 共同最終著者, Under Review [これの前身の論文]

研究概要

- LM の内部表現が含む情報が ①学習ステップ ②層方向 ③モデルサイズ方向でどう変わるか
- 結果 (Observation)
  - ① 言語を個別に学習後、言語間の対応関係を習得 (monolingual→bilingual)
  - ② トークンを個別に表現→言語を超えてまとめる→コンテキストレベルの情報を処理
  - ③ 大きいモデルほどより言語内・言語間の対応関係やコンテキストレベルの情報を扱えるように
- また Bilingual な表現がモデルの性能に大きく関わる

言語モデル解釈の難しさ [Bereska+, 24]

- 言語モデルの内部表現は **Polysemantic** (多義的) で解釈するのが難しい
- 内部表現を **Monosemantic** (一義的) な表現の足し合わせで表現したい



Polysemantic な表現のもつれを解いて Monosemantic に分解したい

スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

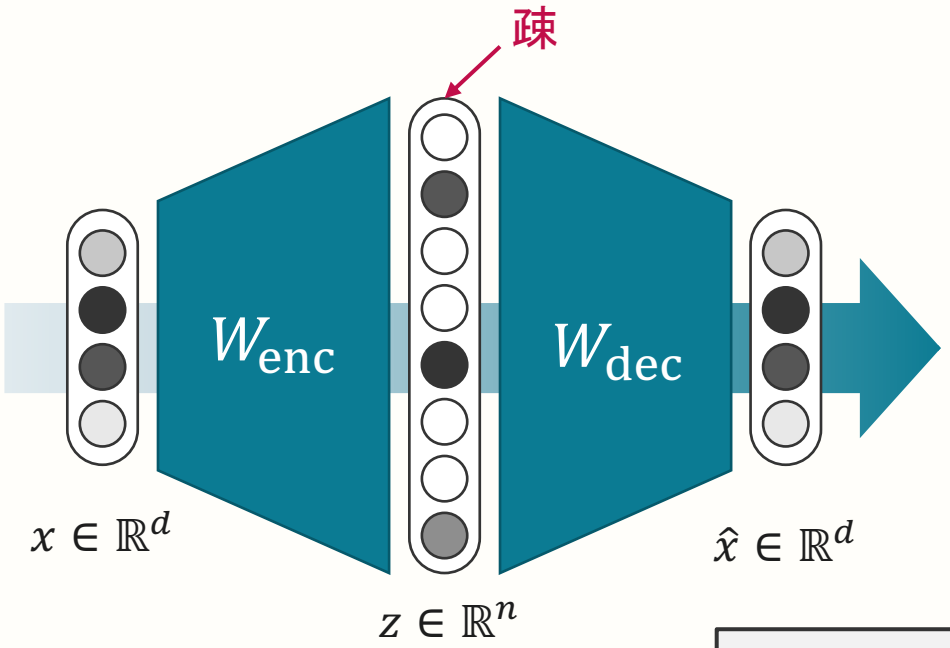
- Polysemantic な表現を Monosemantic な表現の足し合わせに分解
- 中間層が**疎**になるように制約をかけた**オートエンコーダ**
  - 入力表現の次元数 < 中間層の次元数

定式化

$$z = \mathrm{ReLU}\left(W_{\mathrm{enc}}\left(x - b_{\mathrm{pre}}\right)\right)$$
$$\hat{x} = W_{\mathrm{dec}} z + b_{\mathrm{pre}}$$

損失

$$\mathcal{L} = \|x - \hat{x}\|_2^2 + \lambda \|z\|_1$$



$x, b_{\mathrm{pre}} \in \mathbb{R}^d, z \in \mathbb{R}^n, d < n$
$W_{\mathrm{enc}} \in \mathbb{R}^{n \times d}, W_{\mathrm{dec}} \in \mathbb{R}^{d \times n}$

※ b_{pre} と ReLU は省略して図示

[Olshausen+, 97] Sparse coding with an overcomplete basis set: A strategy employed by V1?
[Huben+, 23] Sparse Autoencoders Find Highly Interpretable Features in Language Models

スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

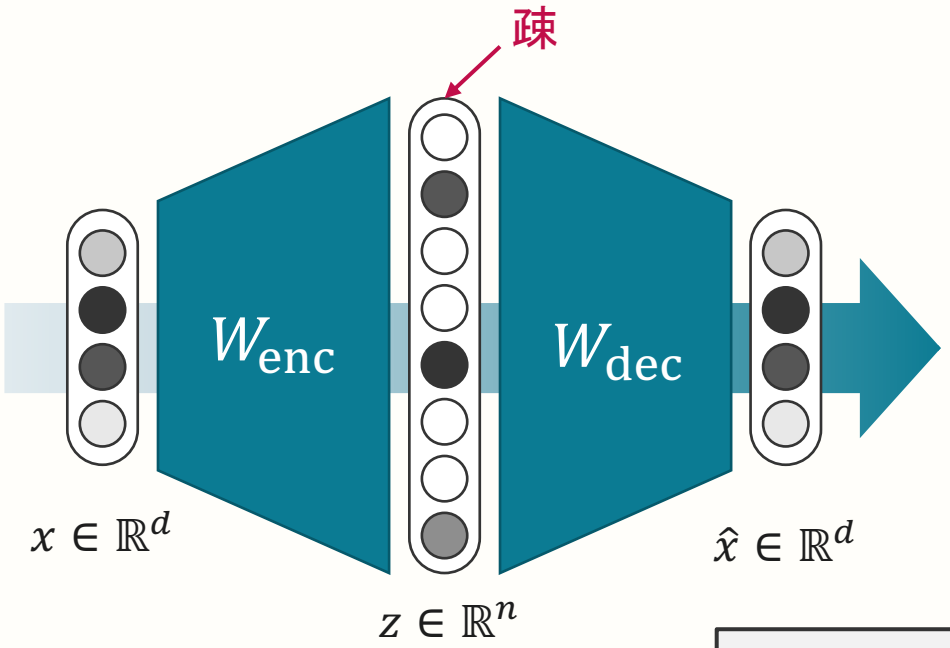
- Polysemantic な表現を Monosemantic な表現の足し合わせに分解
- 中間層が**疎**になるように制約をかけた**オートエンコーダ**
  - 入力表現の次元数 < 中間層の次元数

定式化

$$z = \mathrm{ReLU}\left(W_{\mathrm{enc}}\left(x - b_{\mathrm{pre}}\right)\right)$$
$$\hat{x} = W_{\mathrm{dec}} z + b_{\mathrm{pre}}$$

損失

$$\mathcal{L} = \underbrace{\|x - \hat{x}\|_2^2}_{\text{再構成損失}} + \underbrace{\lambda \|z\|_1}_{\text{疎にする制約}}$$



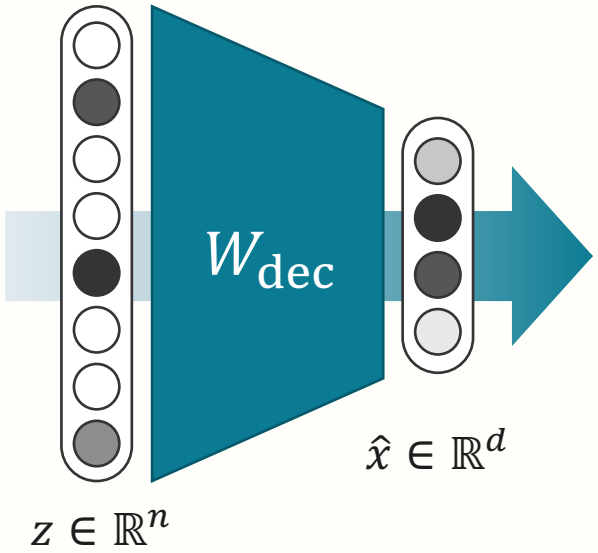
$x, b_{\mathrm{pre}} \in \mathbb{R}^d, z \in \mathbb{R}^n, d < n$
$W_{\mathrm{enc}} \in \mathbb{R}^{n \times d}, W_{\mathrm{dec}} \in \mathbb{R}^{d \times n}$

※ b_{pre}と ReLU は省略して図示

[Olshausen+, 97] Sparse coding with an overcomplete basis set: A strategy employed by V1?
[Huben+, 23] Sparse Autoencoders Find Highly Interpretable Features in Language Models

スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

- $W_{\text{dec}} \in \mathbb{R}^{d \times n}$ を n 個の d 次元ベクトルと見ると、n 個の d 次元ベクトルからいくつかを選び、その重み付き足し合わせで入力ベクトルを再構成するネットワーク

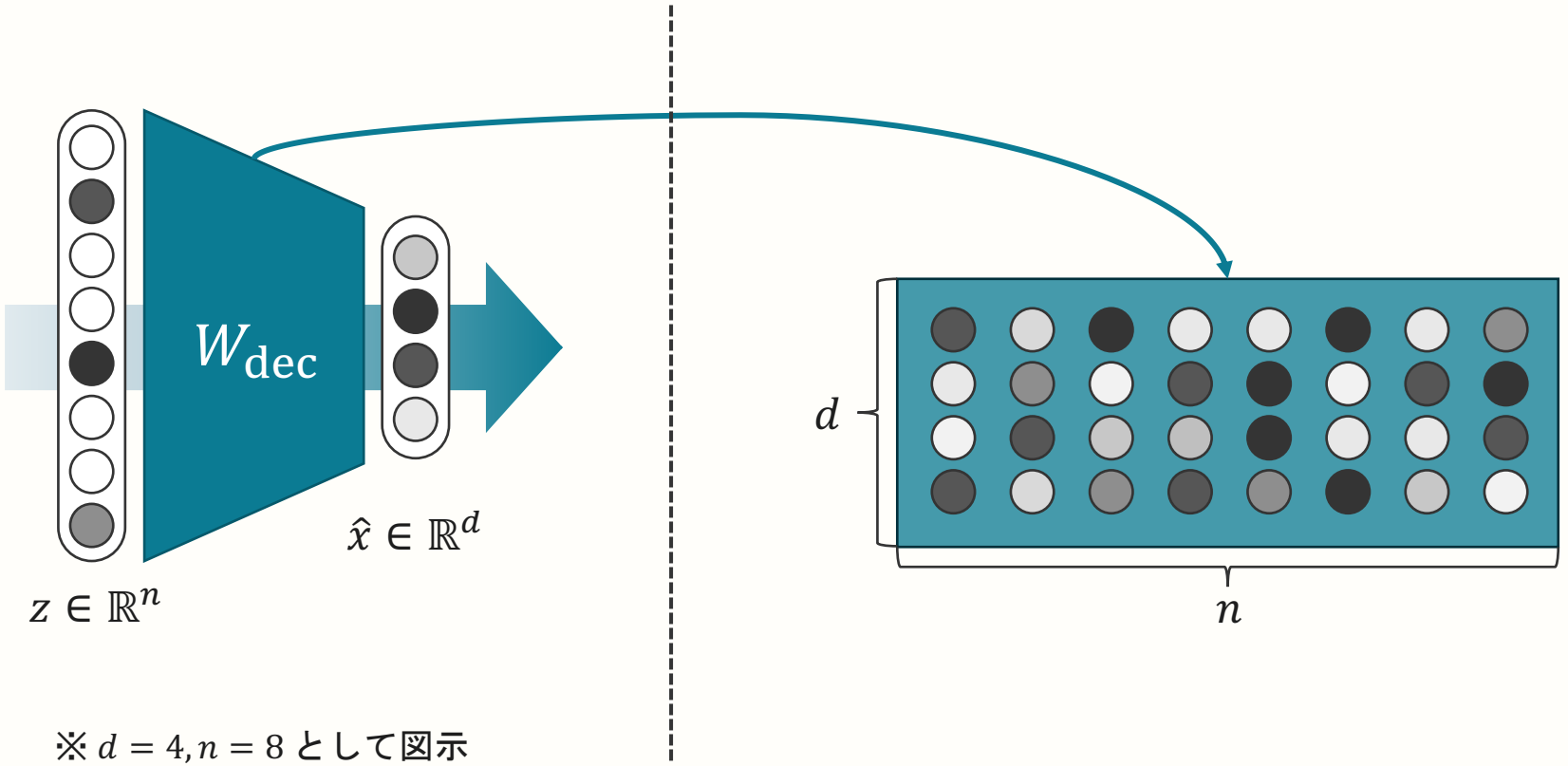


※ $d = 4, n = 8$ として図示

[Olshausen+, 97] Sparse coding with an overcomplete basis set: A strategy employed by V1?
[Huben+, 23] Sparse Autoencoders Find Highly Interpretable Features in Language Models

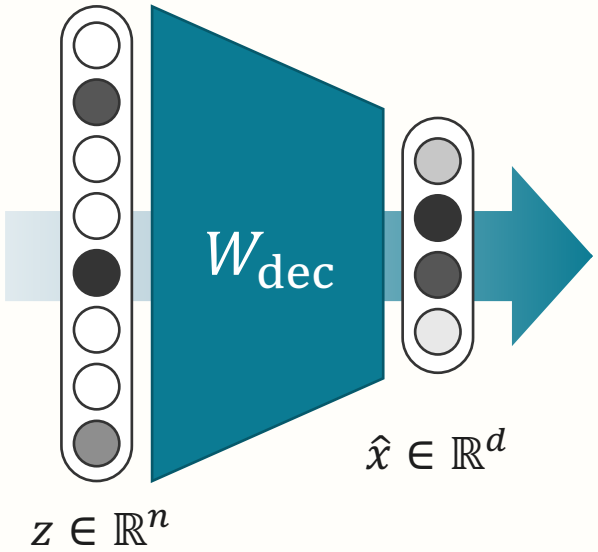
スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

- $W_{\text{dec}} \in \mathbb{R}^{d \times n}$ を n 個の d 次元ベクトルと見ると、n 個の d 次元ベクトルからいくつかを選び、その重み付き足し合わせで入力ベクトルを再構成するネットワーク

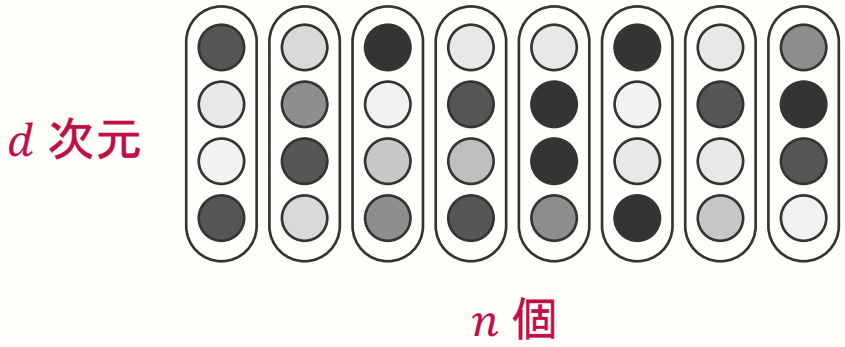


スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

- $W_{\text{dec}} \in \mathbb{R}^{d \times n}$ を n 個の d 次元ベクトルと見ると、n 個の d 次元ベクトルからいくつかを選び、その重み付き足し合わせで入力ベクトルを再構成するネットワーク



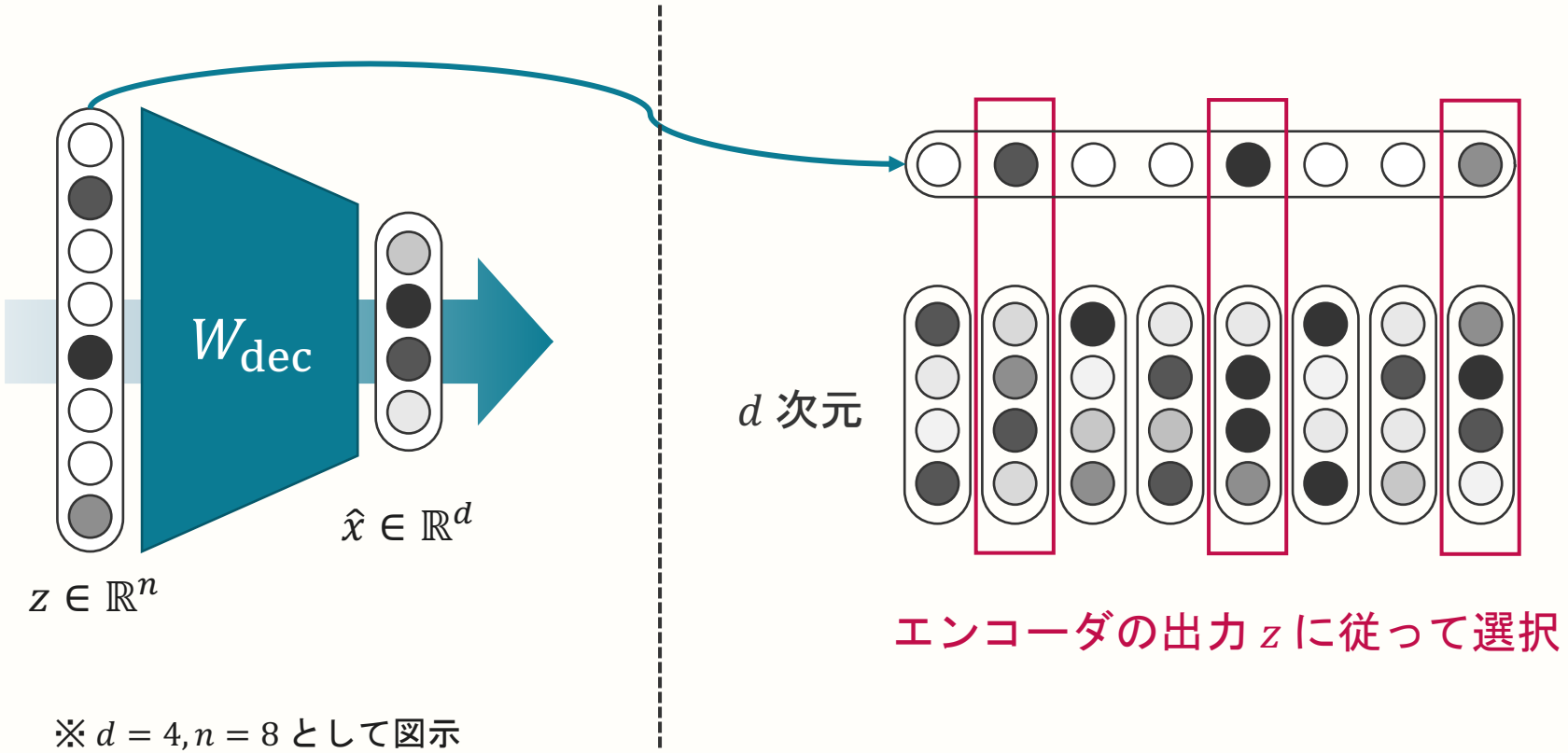
※ $d = 4, n = 8$ として図示



[Olshausen+, 97] Sparse coding with an overcomplete basis set: A strategy employed by V1?
[Huben+, 23] Sparse Autoencoders Find Highly Interpretable Features in Language Models

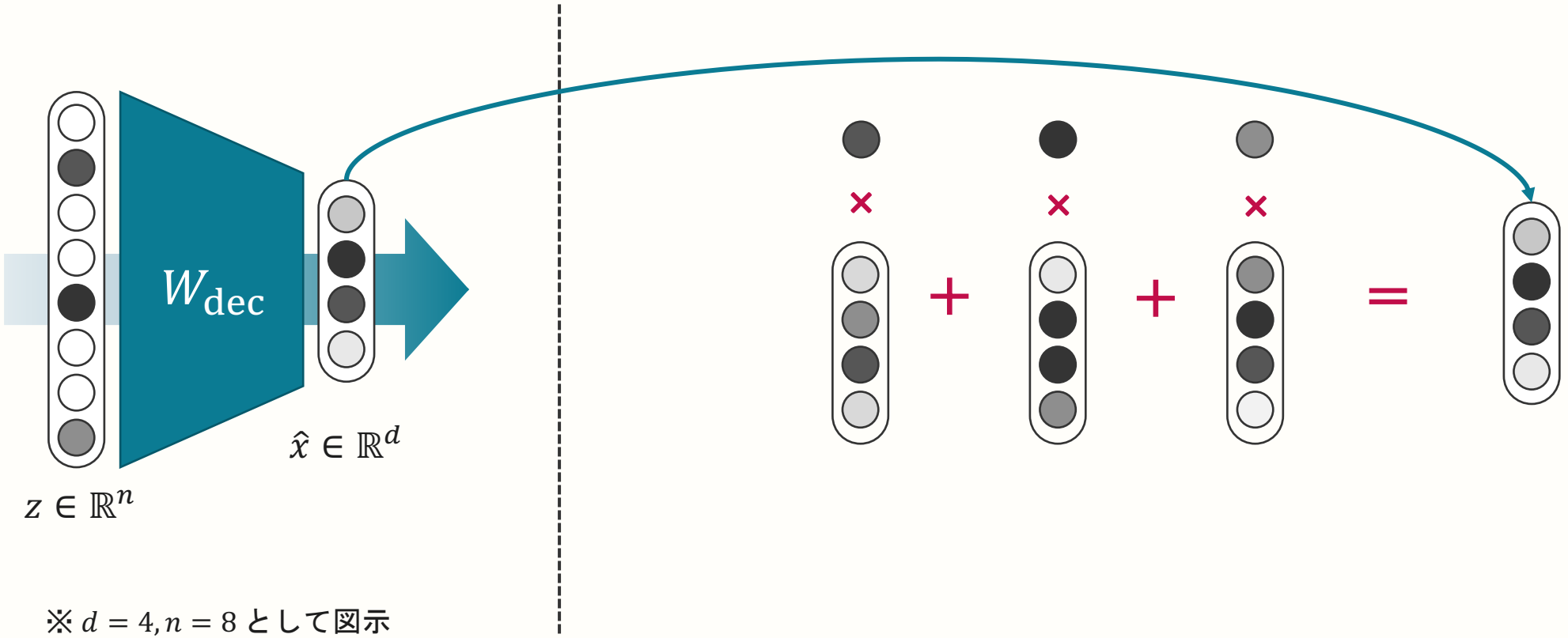
スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

- $W_{\text{dec}} \in \mathbb{R}^{d \times n}$ を n 個の d 次元ベクトルと見ると、n 個の d 次元ベクトルからいくつかを選び、その重み付き足し合わせで入力ベクトルを再構成するネットワーク



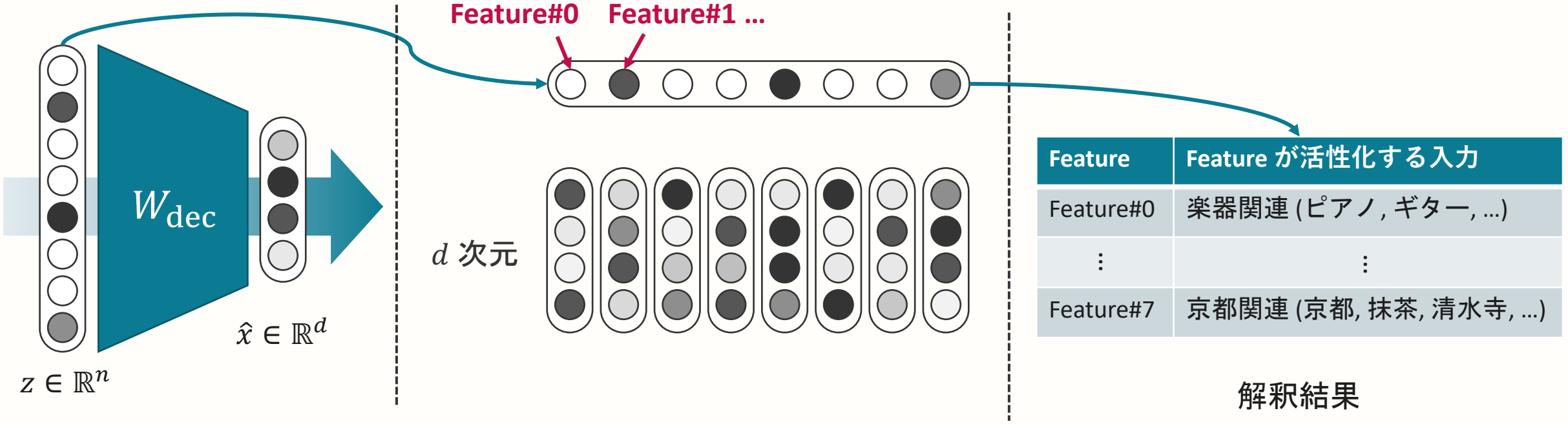
スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

- $W_{\text{dec}} \in \mathbb{R}^{d \times n}$ を n 個の d 次元ベクトルと見ると、n 個の d 次元ベクトルからいくつかを選び、その重み付き足し合わせで入力ベクトルを再構成するネットワーク



スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

- 各ベクトルがどのような入力の時に選ばれるかから、そのベクトルが持つ意味を推定
- 本研究では、各ベクトルに対応する中間層の各次元を Feature と呼び、再構成にその次元 (ベクトル) が使用される時 Feature が活性化しているとみなす



Feature	Feature が活性化する入力
Feature#0	楽器関連 (ピアノ, ギター, …)
⋮	⋮
Feature#7	京都関連 (京都, 抹茶, 清水寺, …)

解釈結果

※ 上の例では、Feature#0 は活性化していないが、Feature#1 は活性化している

スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

-
-

Unsupervised に学習ができる特徴量抽出機

学習した特徴量を見ることで元の表現が含みがちだった情報がわかる

TopK-SAE [Gao+, 24]

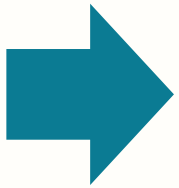
- 中間層の活性化関数を ReLU から TopK に変更
  - Sparsity を K の値で直接コントロール可能で、学習が安定しやすい

定式化

$$z = \mathrm{ReLU}\left(W_{\mathrm{enc}}\left(x - b_{\mathrm{pre}}\right)\right)$$
$$\hat{x} = W_{\mathrm{dec}} z + b_{\mathrm{pre}}$$

損失

$$\mathcal{L} = \|x - \hat{x}\|_2^2 + \lambda \|z\|_1$$



定式化

$$z = \mathrm{TopK}\left(W_{\mathrm{enc}}\left(x - b_{\mathrm{pre}}\right)\right)$$
$$\hat{x} = W_{\mathrm{dec}} z + b_{\mathrm{pre}}$$

損失

$$\mathcal{L} = \|x - \hat{x}\|_2^2$$

評価指標 I: 言語

SAE の Features がどのようなトークンに発火する傾向があるかを測りたい

言語

- (i) 日本語に発火するか (ii) 英語に発火するか (iii) 両方に発火するか
  日本語文章への発火率が90%以上　英語文章への発火率が90%以上　その他

	Activating Tokens	Language
(i)	犬は怖い/猫は可愛い/猫カフェ	Japanese
(ii)	Dogs are scary/Cats are cute/Cat cafe	English
(iii)	Dogs are scary/猫は可愛い/Cat cafe	Mixed

評価指標 II: Monosemanticity

SAE の Features がどのようなトークンに発火する傾向があるかを測りたい

Monosemanticity

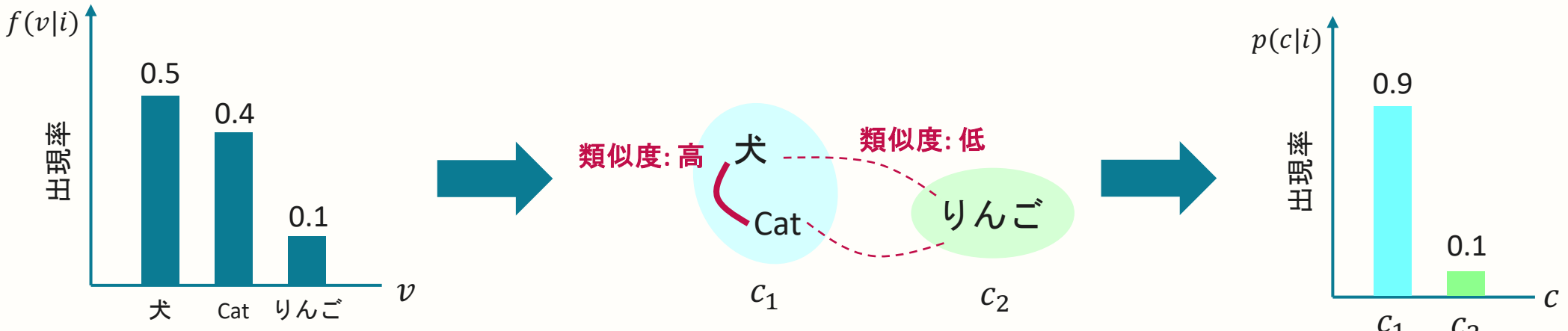
- 発火トークンたちの意味がどれだけ意味的にまとまっているか（0~1）
- 大きいほど単一意味(monosemantic)、小さいほど意味がばらばら(polysemantic)

Activating Tokens	Monosemanticity
犬は怖い/りんご飴/I love that guitar	0.00
犬は怖い/Cat cafe/りんご飴	0.50
Dogs are cute/猫は可愛い/Cat cafe	1.00

評価指標 II: Monosemanticity

計算方法

1. Token Entropy を計算: $H_{\text{token}}(i) = -\sum_{v \in V} f(v|i) \log f(v|i)$
2. Semantic Entropy を計算:
   1. 単語埋め込み表現を使用して、**コサイン類似度**をもとにクラスタリング
   2. クラスターレベルでエントロピーを計算: $H_{\text{semantic}}(i) = -\sum_{c \in C_i} p(c|i) \log p(c|i)$



***i*-th Feature の発火パターン**　　**単語埋め込み空間でクラスタリング**

※ 上記例において i-th Feature のクラスターは2つ $C_i = \{c_1, c_2\}$

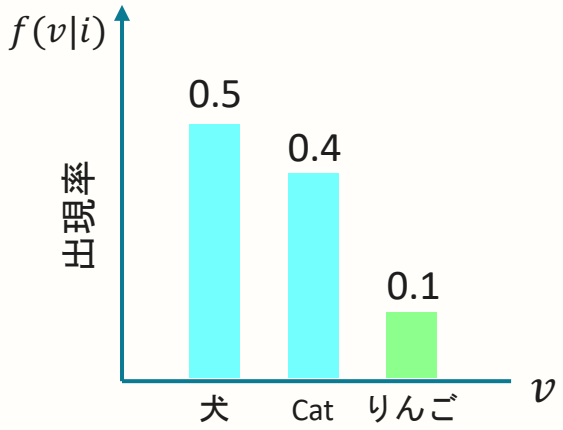
評価指標 II: Monosemanticity

計算方法

3. Token entropy と Semantic entropy から Monosemanticity を計算

$$R_{\text{mono}}(i) = 1 - \frac{H_{\text{semantic}}(i)}{H_{\text{token}}(i)}$$

- $R_{\text{mono}} \sim 1 \rightarrow H_{\text{token}} \gg H_{\text{semantic}}$ → 意味的にまとまりのあるトークンに発火 (monosemantic)
- $R_{\text{mono}} \sim 0 \rightarrow H_{\text{token}} \approx H_{\text{semantic}}$ → いろんな意味のトークンに発火 (polysemantic)



$$H_{\text{token}} = 0.5 \log 0.5 + 0.4 \log 0.4 + 0.1 \log 0.1 \fallingdotseq 0.41$$

$$H_{\text{semantic}} = 0.9 \log 0.9 + 0.1 \log 0.1 \fallingdotseq 0.14$$

$$R_{\text{mono}} = 1 - \frac{0.14}{0.41} \fallingdotseq 0.66$$

そこそこ Monosemantic

※ $H_{\text{token}}(i)$が0のとき、R_{mono}は1とする

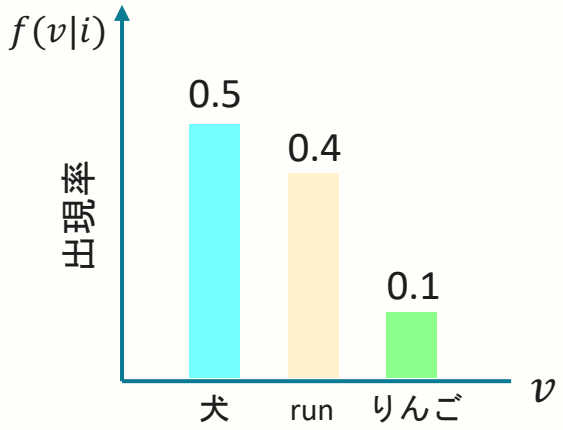
評価指標 II: Monosemanticity

計算方法

3. Token entropy と Semantic entropy から Monosemanticity を計算

$$R_{\text{mono}}(i) = 1 - \frac{H_{\text{semantic}}(i)}{H_{\text{token}}(i)}$$

- $R_{\text{mono}} \sim 1 \rightarrow H_{\text{token}} \gg H_{\text{semantic}}$ → 意味的にまとまりのあるトークンに発火 (monosemantic)
- $R_{\text{mono}} \sim 0 \rightarrow H_{\text{token}} \approx H_{\text{semantic}}$ → いろんな意味のトークンに発火 (polysemantic)



$$H_{\text{token}} = 0.5 \log 0.5 + 0.4 \log 0.4 + 0.1 \log 0.1 \fallingdotseq 0.41$$

$$H_{\text{semantic}} = 0.5 \log 0.5 + 0.4 \log 0.4 + 0.1 \log 0.1 \fallingdotseq 0.41$$

$$R_{\text{mono}} = 1 - \frac{0.41}{0.41} \fallingdotseq 0.00$$

まったく Monosemantic じゃない
（→ Polysemantic）

※ $H_{\text{token}}(i)$が0のとき、R_{mono}は1とする

実験

分析対象

- LLM-jp familiy (150M, 440M, 980M, 1.8B, 3.7B) の全層 (12, 16, 20, 24, 28 層ずつ)
- Checkpoints は log スケールで大体等間隔に16箇所 (10, 20, 50, 100, 200, 500, …, 500000, 988240)*
- モデルサイズ x 層 x ckpt で計 1572 箇所を分析

SAE学習のデータセット: LLM-jp-corpus v3 の 日本語 wiki (50%) と 英語 wiki (50%)

- 計 100M トークンを 8:1:1 で訓練, 検証, テスト用に
- 64トークン分を LLM に入力
  → [BOS] トークンを除いた63トークン分の内部表現をL2正規化してSAEに入力

TopK-SAE のパラメータ: K=32, 中間層の次元数 n=32768

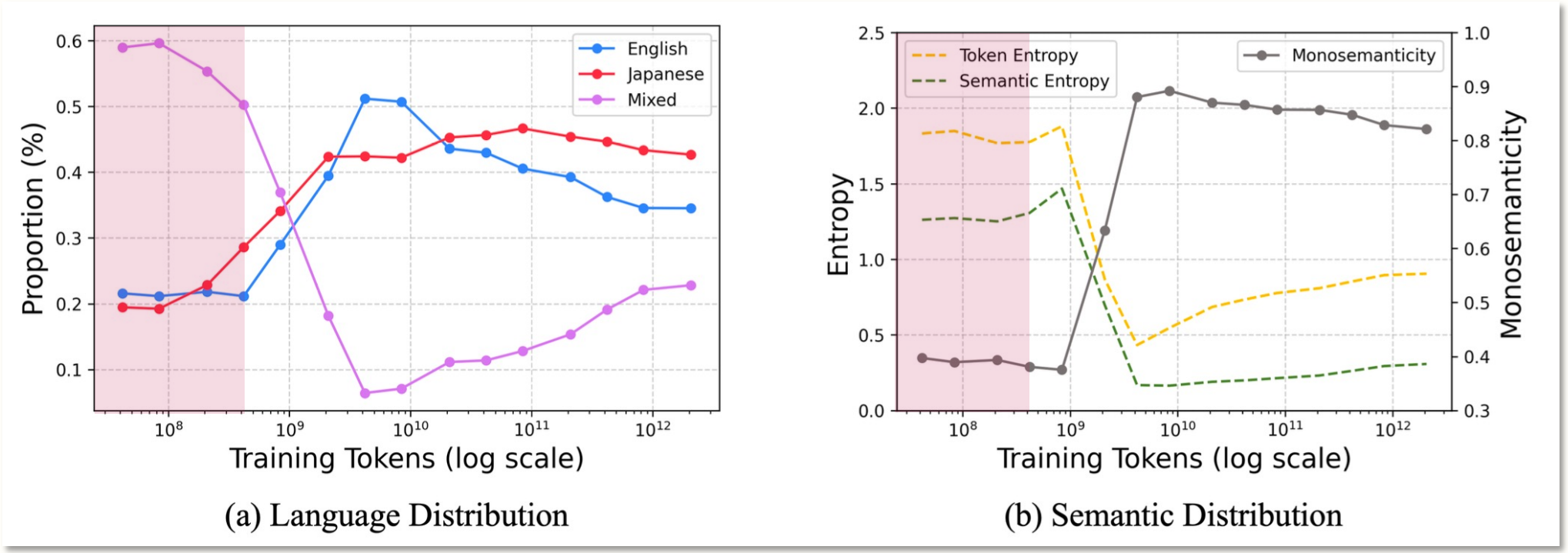
* 3.7B モデルのみバッチサイズが異なり、10, 20, 50, …, 200000, 494120の15箇所を使用

結果①: training 方向 (layer=14, size=3.7B)

学習初期 (~4 x 10^8)

- 両言語で発火する Mixed features の割合が多く、Monosemanticity は小さい

  → 日英バラバラのランダムなトークンで発火する feature が多い



結果①: training 方向 (layer=14, size=3.7B)

学習初期 (~4 x 10^8)

- 両言語で発火する Mixed features の割合が多く、Monosemanticity は小さい

  → 日英バラバラのランダムなトークンで発火する feature が多い

Activating tokens	Language	Monosemanticity
・Born 20 June 1967) is ・This American Life episodes ・西部、ジュネーブ州の	Mixed	0.19
・in Houston County, Albama ・Trichromia repanda is a ・大会は1938年の2月	Mixed	0.24

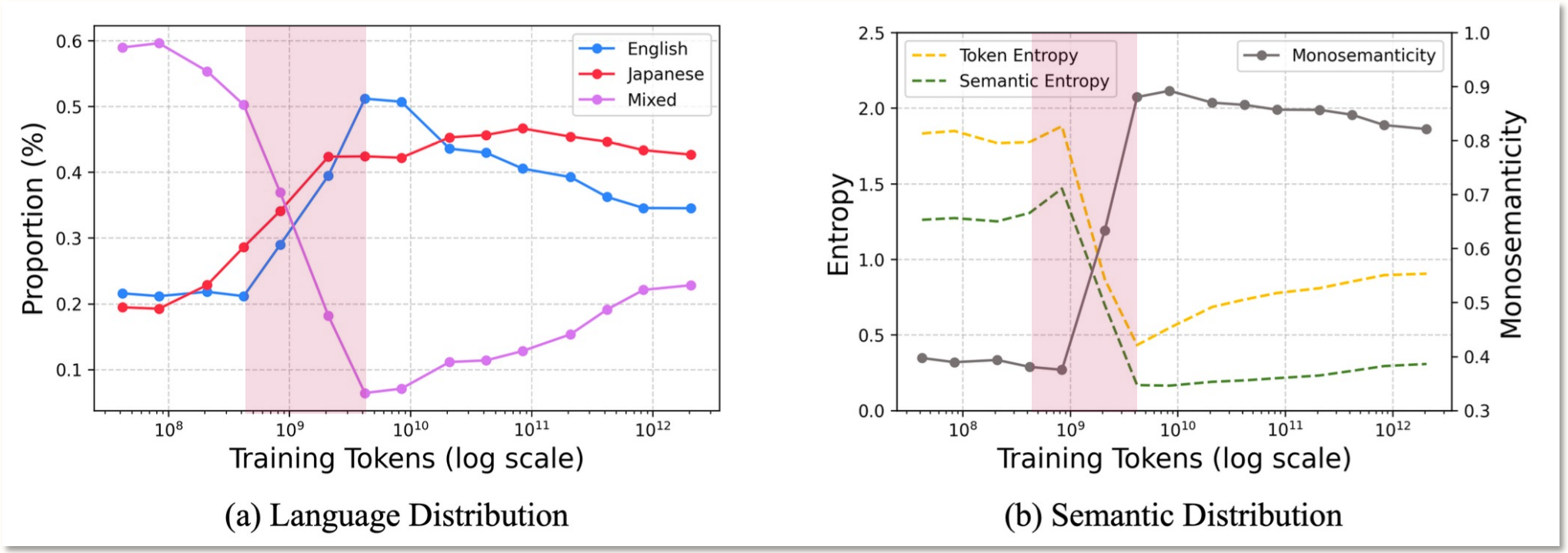
学習初期 feature 例

結果①: training 方向 (layer=14, size=3.7B)

学習中期 (4 x 10^8~4 x 10^9)

- English / Japanese の割合が増え (Mixed が減り)、Monosemanticity が上がる

  → feature が各言語内での特定の概念を捉えるようになる



(a) Language Distribution　　(b) Semantic Distribution

結果①: training 方向 (layer=14, size=3.7B)

学習中期 (4×10^8~4×10^9)

- English / Japanese の割合が増え (Mixed が減り)、Monosemanticity が上がる

  → feature が各言語内での特定の概念を捉えるようになる

・which give rise to ・secretly gave assistance to ・which had given some	English	1.00
・は、ドイツの哲学者 ・、日本の明治期の ・は、イギリスの法学者	Japanese	1.00

学習中期 feature 例

結果①: training 方向 (layer=14, size=3.7B)

学習中期 (4×10^8~4×10^9)

- English / Japanese の割合が増え (Mixed が減り)、Monosemanticity が上がる

  → feature が各言語内での特定の概念を捉えるようになる

  → **言語ごとに独立に意味を習得**

・which give rise to ・secretly gave assistance to ・which had given some	English	1.00
・は、ドイツの哲学者 ・、日本の明治期の ・は、イギリスの法学者	Japanese	1.00

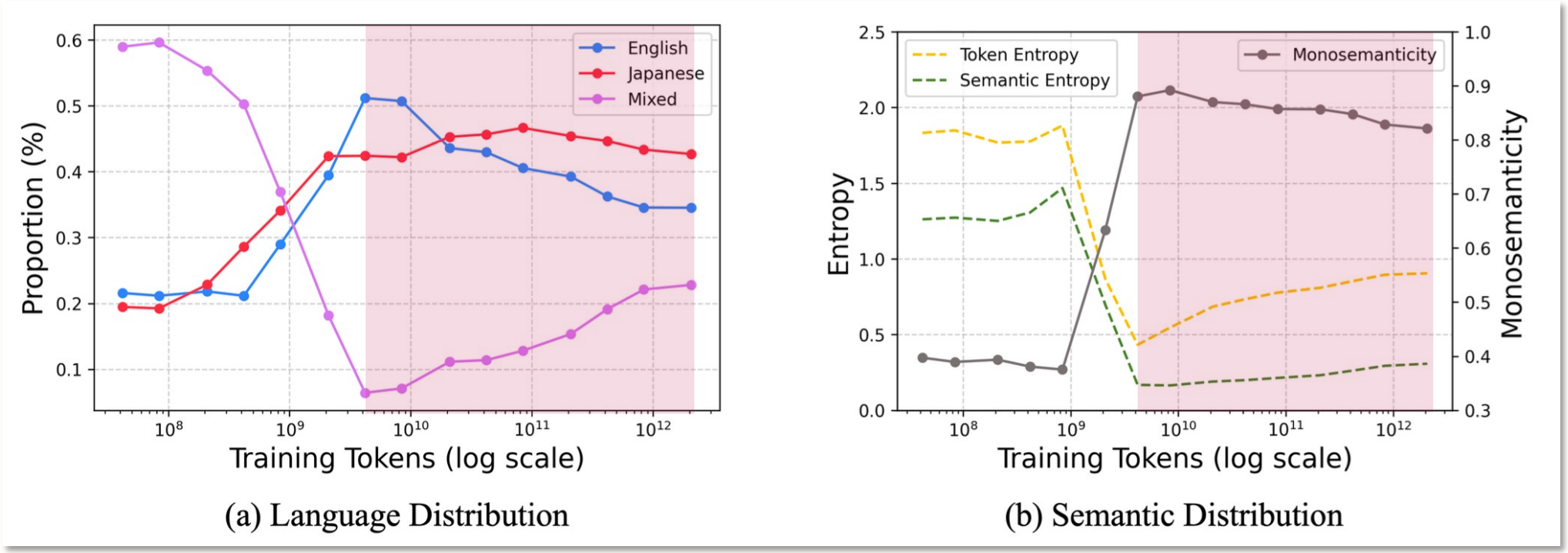
学習中期 feature 例

結果①: training 方向 (layer=14, size=3.7B)

学習後期 (4 x 10^9~)

- Mixed feature が少しずつ増え、Monosemanticity は比較的高いまま

  → 言語を超えて意味を捉える feature が増え始める



結果①: training 方向 (layer=14, size=3.7B)

学習後期 (4×10^9~)

- Mixed feature が少しずつ増え、Monosemanticity は比較的高いまま

  → 言語を超えて意味を捉える feature (bilingual feature) が増え始める

  → **日英間の対応関係を習得**

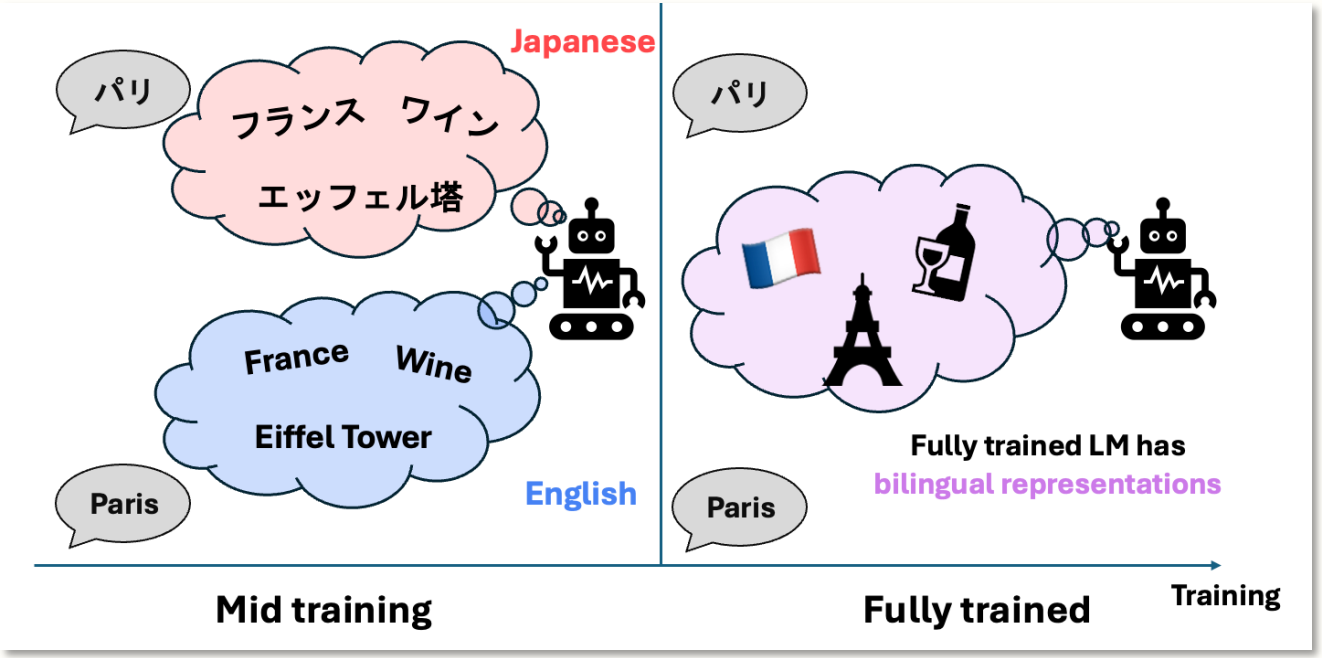
・It was last assigned to the ・The channel assigns series ・に割り当てられており、	Mixed	0.85
・different ritual and social ・as a ceremonial or heraldic ・のような儀式用の穀物	Mixed	0.62

学習後期 feature 例

結果①: training 方向 (layer=14, size=3.7B)

Training 方向考察まとめ

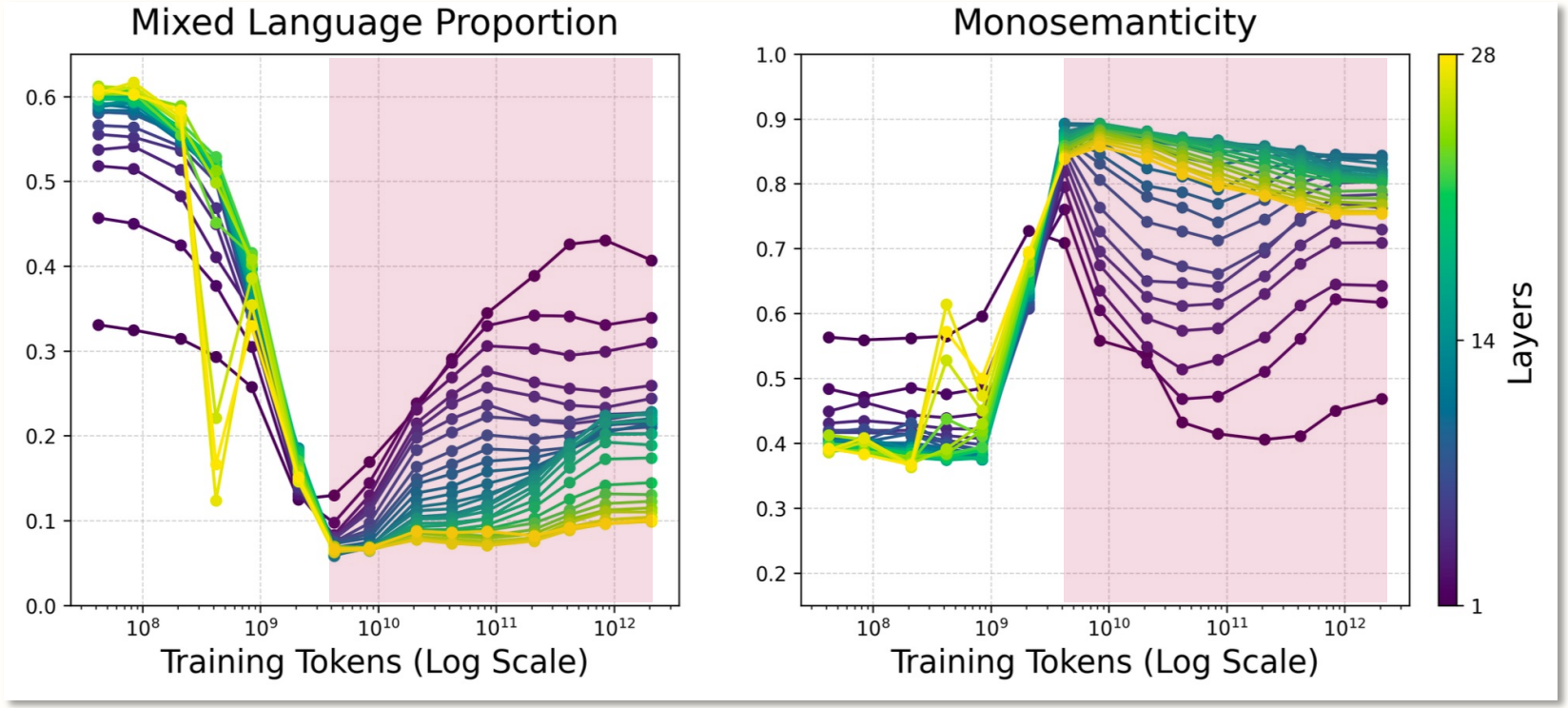
1. 学習中期にかけて、言語ごとに個別に意味を習得
2. 学習後期に、言語間 (日英間) の対応関係を習得



結果②: layer 方向 (size=3.7B)

- 学習後半にMixed の割合が増えつつ、Monosemanticity が高いままなのは**中間あたりの層**

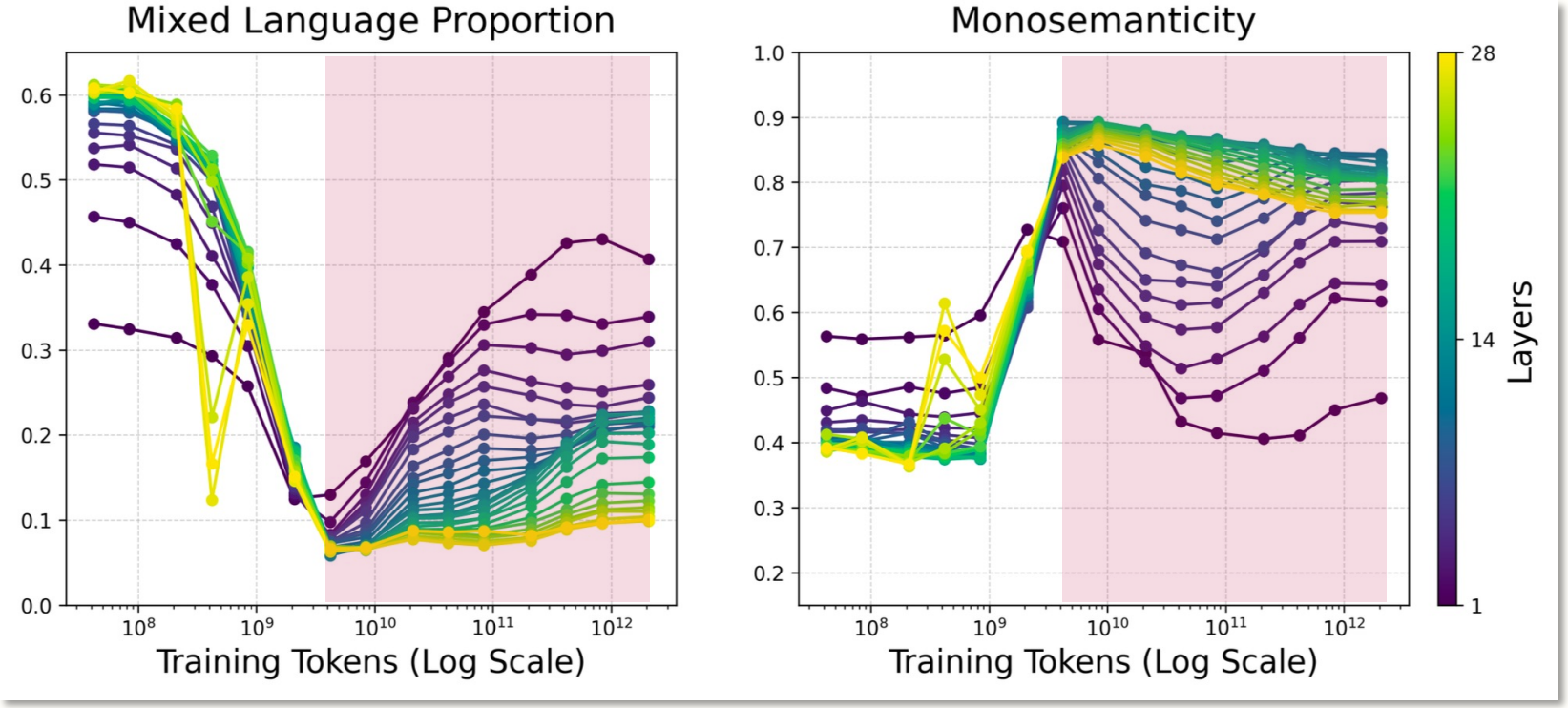
  → **中間あたりの層**が Bilingual な対応を学習する役割を果たす



結果②: layer 方向 (size=3.7B)

- **浅い層**は Mixed の割合が大きく増え、Monosemanticity は比較的減少

  → 学習後半に Polysemantic な feature となる



結果②: layer 方向 (size=3.7B)

- **浅い層**は Mixed の割合が大きく増え、Monosemanticity は比較的減少

  → 学習後半に Polysemantic な feature となる (random ではない)

Activating tokens	Language	Monosemanticity
・, surgeon, and laryngologist ・orthopedic surgeon in the ・、南極海、南極大陸を	Mixed	0.22
・A portion of the shoreline ・and delivery platform. ・social media platforms or	English	0.39

学習済み, 2層目の feature 例

結果②: layer 方向 (size=3.7B)

- **浅い層**は Mixed の割合が大きく増え、Monosemanticity は比較的減少

  → 学習後半に Polysemantic な feature となる (random ではない)

  → 入力の多様さ (vocabの数=100,000)に対応するために、各featureが複数の意味を含む

  → **浅い層**は異なる入力を異なる表現として扱っている? (10万個を区別しようとしてる?)

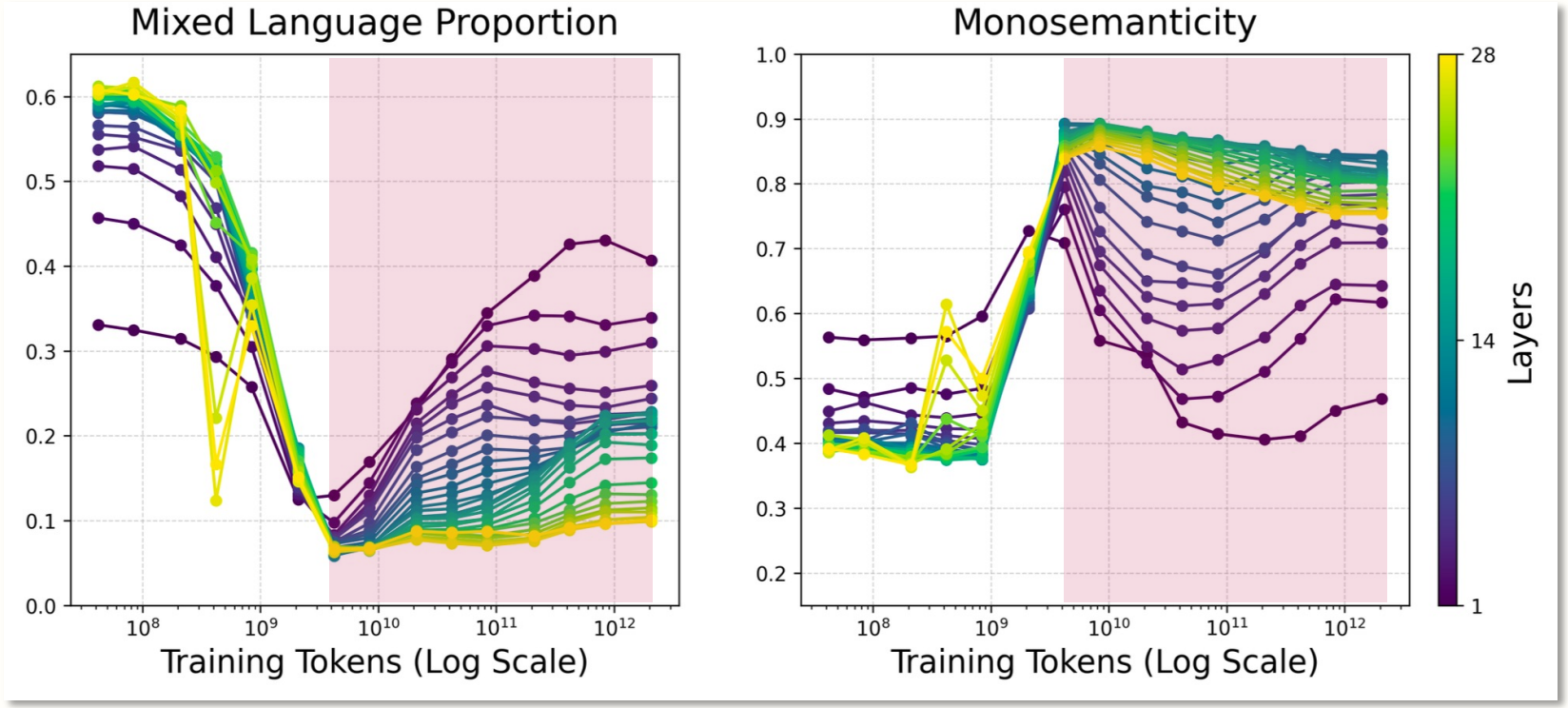
Activating tokens	Language	Monosemanticity
・, surgeon, and laryngologist ・orthopedic surgeon in the ・、南極海、南極大陸を	Mixed	0.22
・A portion of the shoreline ・and delivery platform. ・social media platforms or	English	0.39

学習済み, 2層目の feature 例

結果②: layer 方向 (size=3.7B)

- **深い層**は Mixed の割合があまり増えず、Monosemanticity は中層よりも減少

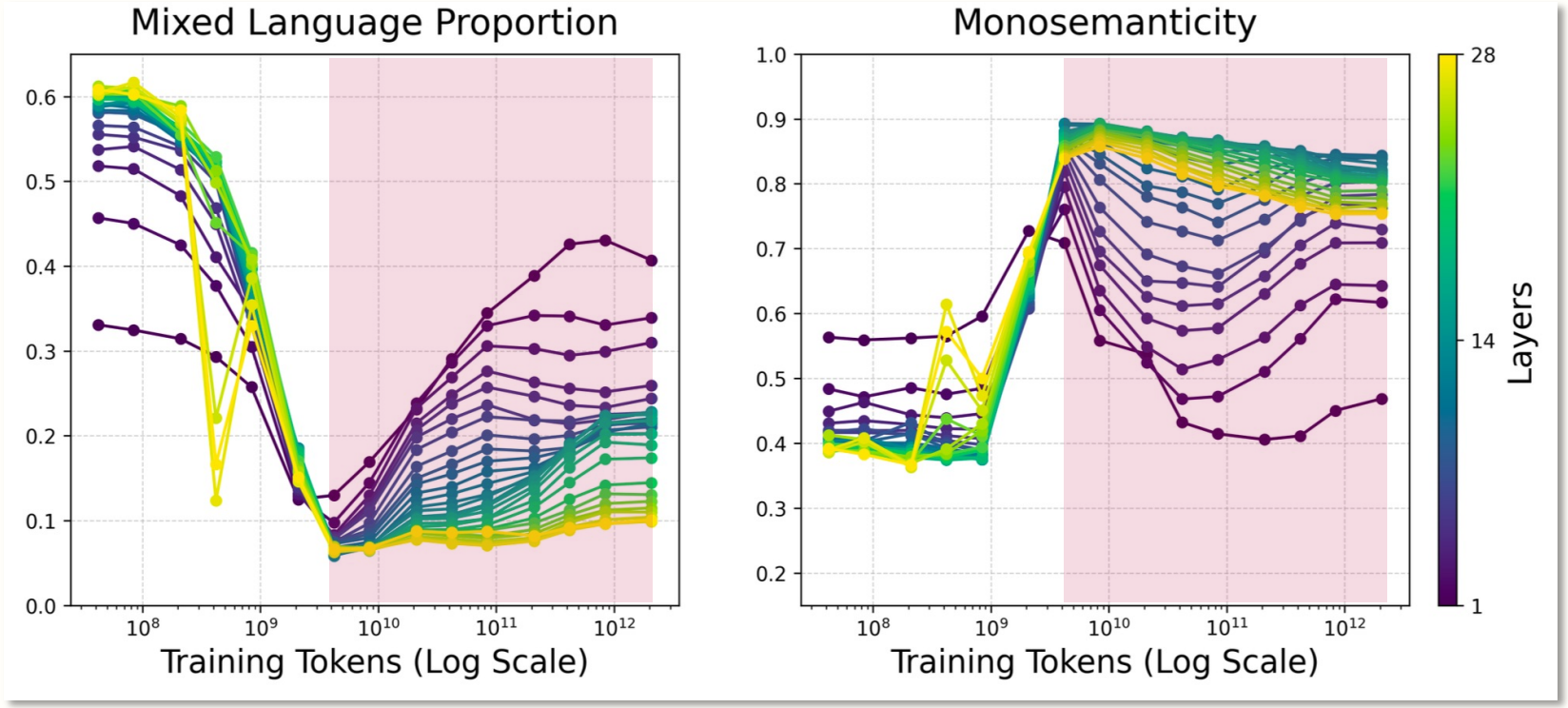
  → 単言語への発火だがちょっとだけバラバラになる



結果②: layer 方向 (size=3.7B)

- **深い層**は Mixed の割合があまり増えず、Monosemanticity は中層よりも減少

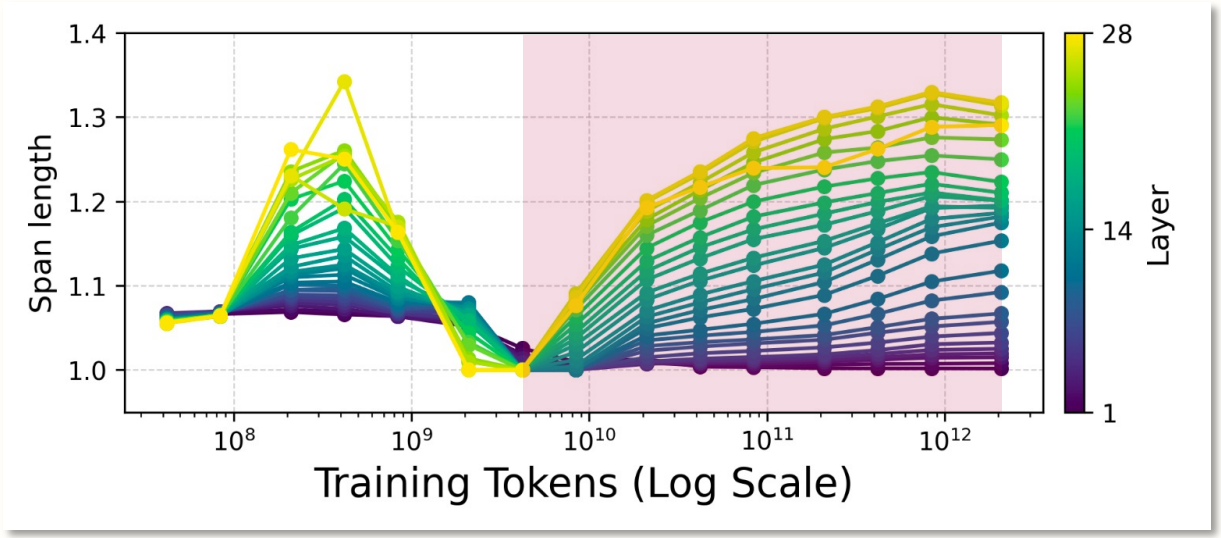
  → 単言語への発火だがちょっとだけバラバラになる → **なぜ?**



結果②: layer 方向 (size=3.7B)

- **深い層**は feature の span の平均長 (発火する時に何トークン連続で発火するか)をみると、学習後半に増加する傾向がある

  → feature がコンテキストレベルの意味を捉えるようになる



結果②: layer 方向 (size=3.7B)

- **深い層**は feature の span の平均長 (発火する時に何トークン連続で発火するか)をみると、学習後半に増加する傾向がある

  → feature がコンテキストレベルの意味を捉えるようになる

  → **深い層**は単言語のコンテキストレベルの情報を扱う?

・stuccoed brick building. ・-story wood-frame house ・brick and sandstone dwelling	English	0.55
・2丁目10番1号に所在する ・麻布台一丁目にある ・安井四丁目に鎮座する	Japanese	1.00

学習済み, 26層目の feature 例

結果②: layer 方向 (size=3.7B)

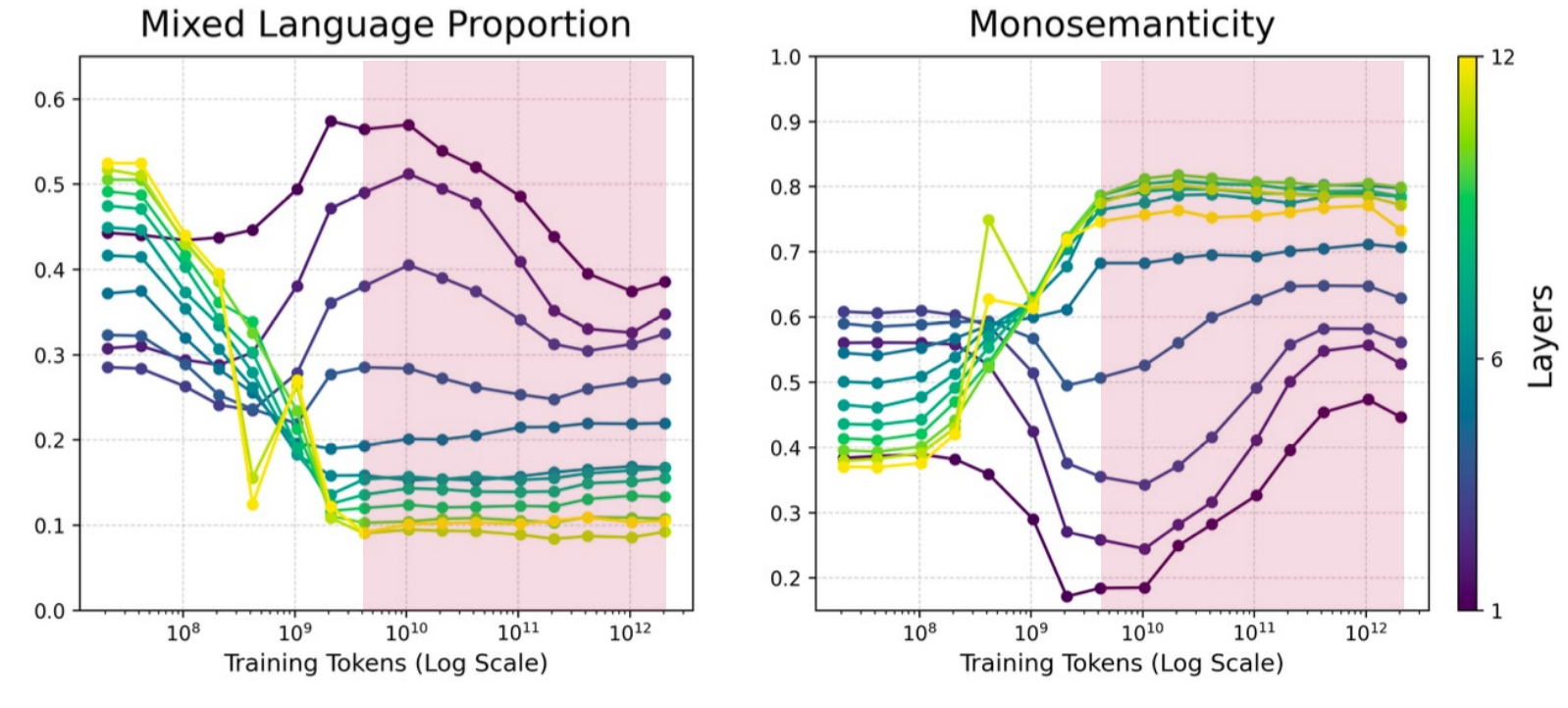
Layer 方向考察まとめ

1. 浅い層: 入力の多様さをそのままエンコードして区別して処理
2. 中間層: 同じ意味のものを言語を超えてまとめて処理
3. 深い層: コンテキストレベルの情報を処理

結果③: size 方向

- 150M では、学習後半にMixed の割合が増え、Monosemanticityが高いままの層が無い

  → bilingual な対応をとらえている feature (層) が無い?

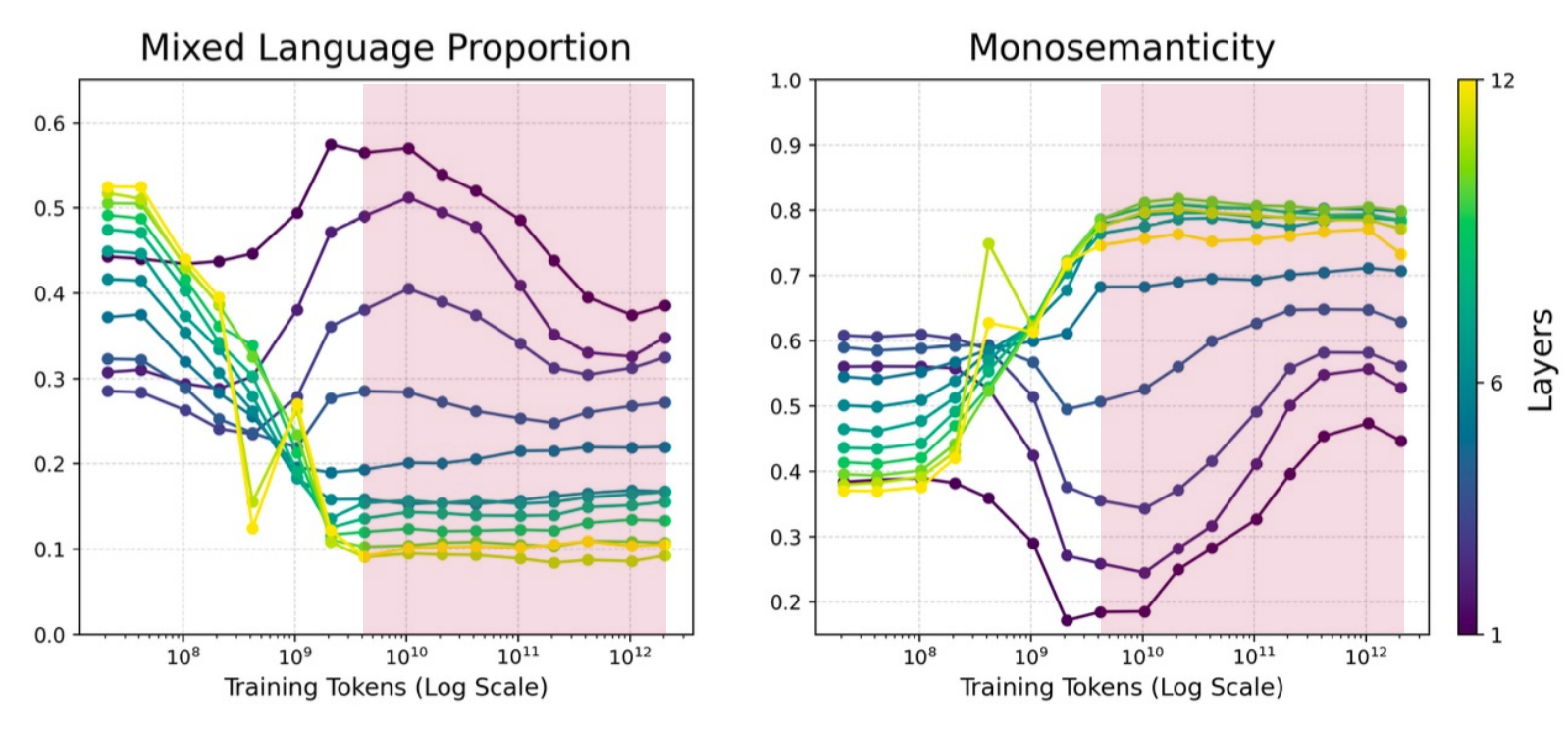


150M

結果③: size 方向

- 150M では、Monosemanticity が全体的に低い

  → 単言語内における意味の共通性を捉える能力が低い?

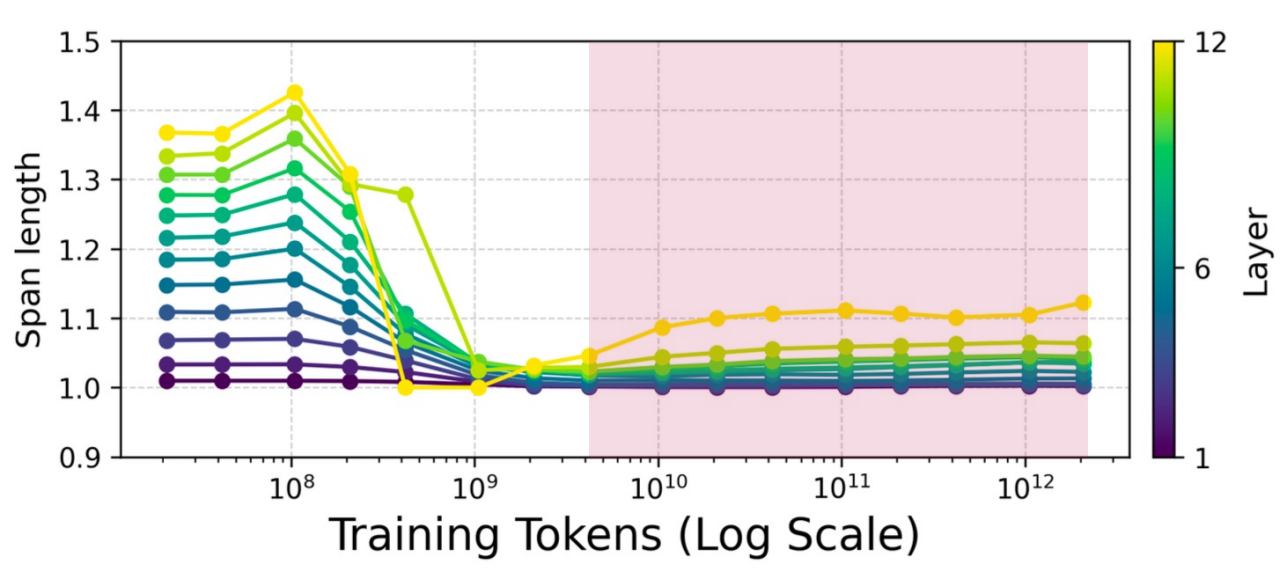


150M

結果③: size 方向

- 150M では、Span の平均長も大きいモデルに比べて短い

  → コンテキストレベルの意味を捉える能力も低い



150M

結果③: size 方向

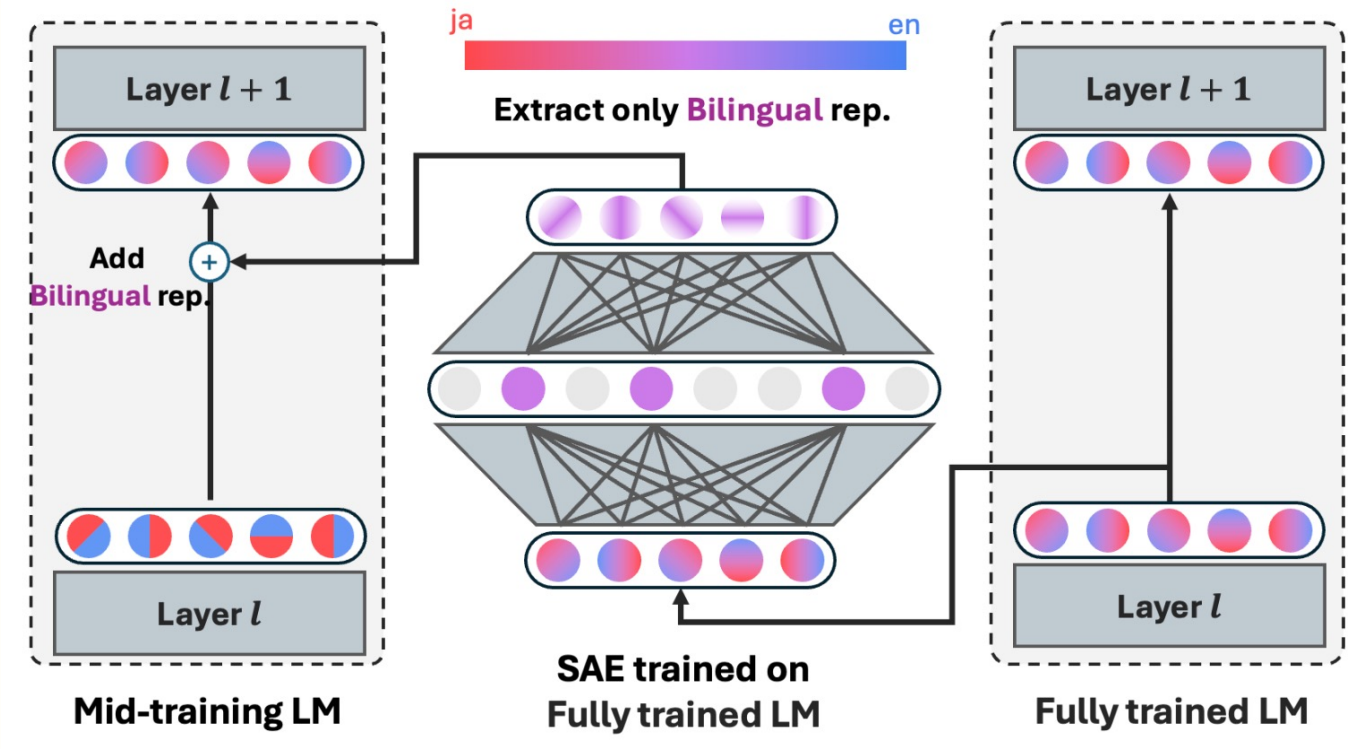
Size 方向考察まとめ

1. 小さいモデルは bilingual な対応をとらえれていない
2. 言語内で共通の意味を捉える能力はモデルサイズに比例する
3. コンテキストレベルの意味を捉える能力もモデルサイズに比例

介入実験: 目的・手法

目的: 学習後半に大きいモデルの中間層が習得する bilingual な表現が重要であることを示す

手法: 学習済みモデルのbilingualな表現をSAEを用いて抽出→学習中期のモデルの内部に加える



介入実験: 結果

- En/Ja な表現を学習済み→学習中期に加えると、各言語の能力が特に増加
- Bilingual な表現を学習済み→学習中期に加えると、En/Jaに比べて能力増加幅が大きい

	Perplexity (dif.)		
Add Rep.	English	Japanese	all
Baseline	17.57	19.54	15.39
English	−0.16	−0.11	−0.14
Japanese	−0.10	−0.36	−0.24
Bilingual	−0.37	−0.72	−0.56

介入後のPPL変化 (baselineは介入なし学習中期モデル)

まとめ

- SAEを用いて Training/Layer/Size の3方向で分析を行った
  - Training: 言語ごとに習得→言語間の対応を習得
  - Layer: トークンを個別の表現として→言語を超えて意味をまとめる→コンテキストレベルに
  - Size: 大きいモデルほど、言語内での共通の意味、Bilingual な対応、コンテキスト情報を捉える

今後の展望

- Instruction Tuning まで分析の幅を広げる
- Dead feature 分析から見つけた、Training 方向での使用しうる語彙種類数に関して