

# 研究紹介

## (音楽情報処理の話 + 最近やってるNLP研究)

---

稲葉 達郎

M2 @ Kyoto Univ. / RA @ NII

2025/3/5, Tohoku NLP Group MiCS



# 自己紹介

## 経歴

- 2019~2023年 京都大学工学部電気電子工学科
  - ・ B4では旧黒橋研 (言語メディア研) で NLP の研究
- 2023~2025年 京都大学情報学研究科知能情報学コース
  - ・ 河原研 (音声メディア研) で音楽情報処理の研究
- 2025年~ 博士課程進学...?

## 研究キーワード

- ・ 音楽と言語のマルチモーダルモデル
- ・ 音楽/言語モデルの解釈可能性

## 趣味

- ・ 音楽 ([作曲](#)・[ギター](#)・[ピアノ](#))
- ・ お酒・サッカー

# 音楽情報処理とは

## 音楽に関連するデータの処理全般

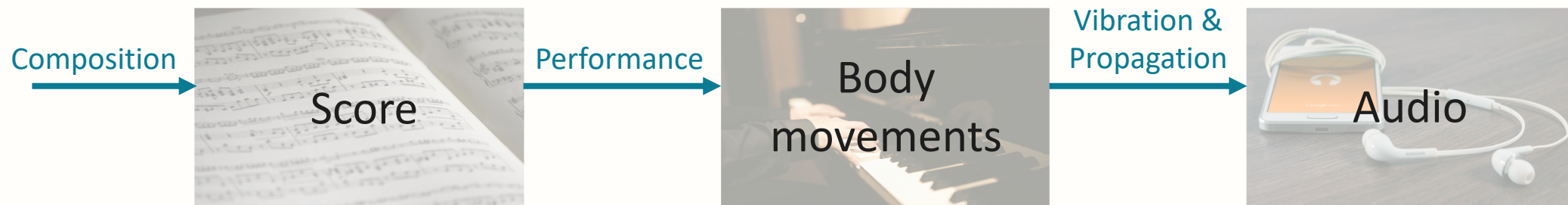
- Music Information Retrieval (MIR) という研究分野名がよく使われる
- MIR はインターネットとデジタル音楽の普及から音楽情報を検索・取得する技術として発展

分類の観点 [Goto, 10]	説明
認識・理解 vs 生成	音楽を入力とするか出力とするか
信号処理 vs 記号処理	音楽を音響信号として扱うか楽譜・演奏表現として扱うか
リアルタイム vs 非リアルタイム	インタラクティブ性の有無
システムを作る vs 人間を知る	工学の立場でシステムを作るのか科学の立場で人間を解明するか

# 音楽情報処理: タスク例

我々が普段聴いている音楽 (Audio) ができるまで

- 作曲 (Composition) により譜面 (Score) ができる
- 譜面が演奏 (Performance) により身体動作 (Body movements) となる
- 身体動作が音響信号に変換 (Vibration & propagation) され、オーディオ (Audio) となる



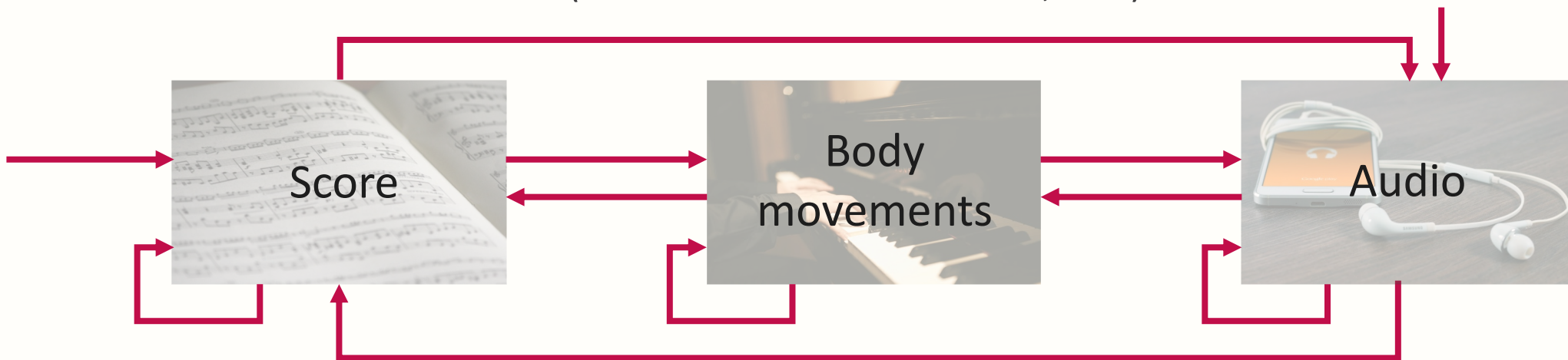
# 音楽情報処理: タスク例

我々が普段聴いている音楽 (Audio) ができるまで

- 作曲 (Composition) により譜面 (Score) ができる
- 譜面が演奏 (Performance) により身体動作 (Body movements) となる
- 身体動作が音響信号に変換 (Vibration & propagation) され、オーディオ (Audio) となる



計算機上でこの過程を真似る、あるいはその逆等を行うのが  
音楽情報処理 (Music Information Retrieval; MIR) タスクの一部



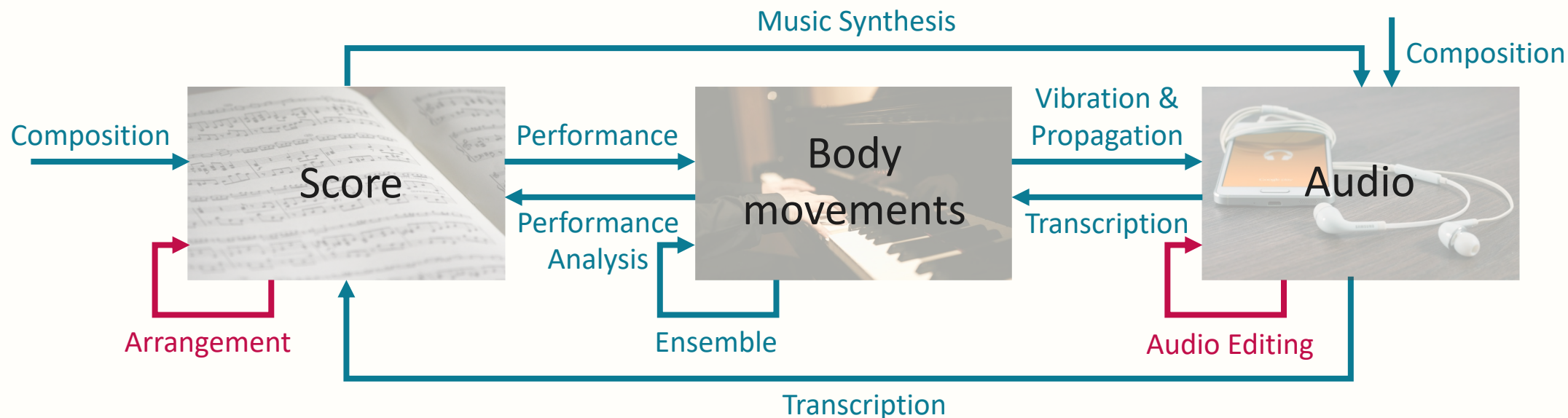
# 音楽情報処理: タスク例

## Score-to-Score の例

- Arrangement: ピアノソロをオーケストラ用に編曲 (Orchestration)、難易度を下げる (Reduction) 等
- Melody-to-Accompaniment: メロディを条件に伴奏を生成、リアルタイムでのタスク設定も

## Audio-to-Audio の例

- Audio Editing: ボーカルのピッチ補正、リバーブの追加・除去、自動ミキシング & マスタリング
- Source Separation: 元音源を楽器ごとに分離



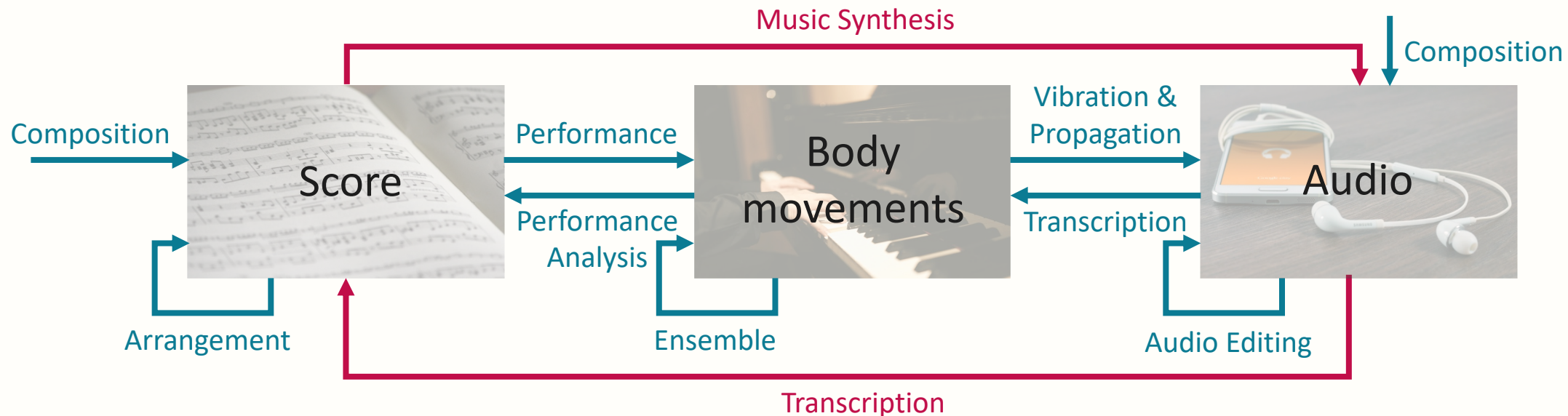
# 音楽情報処理: タスク例

## Score-to-Audio の例

- Music Synthesis: 楽譜から音響信号を作成
  - 事前に録音した音を楽譜データにマッピング
  - 楽器の物理特性をモデリングしてシミュレーション
  - DNNで生成

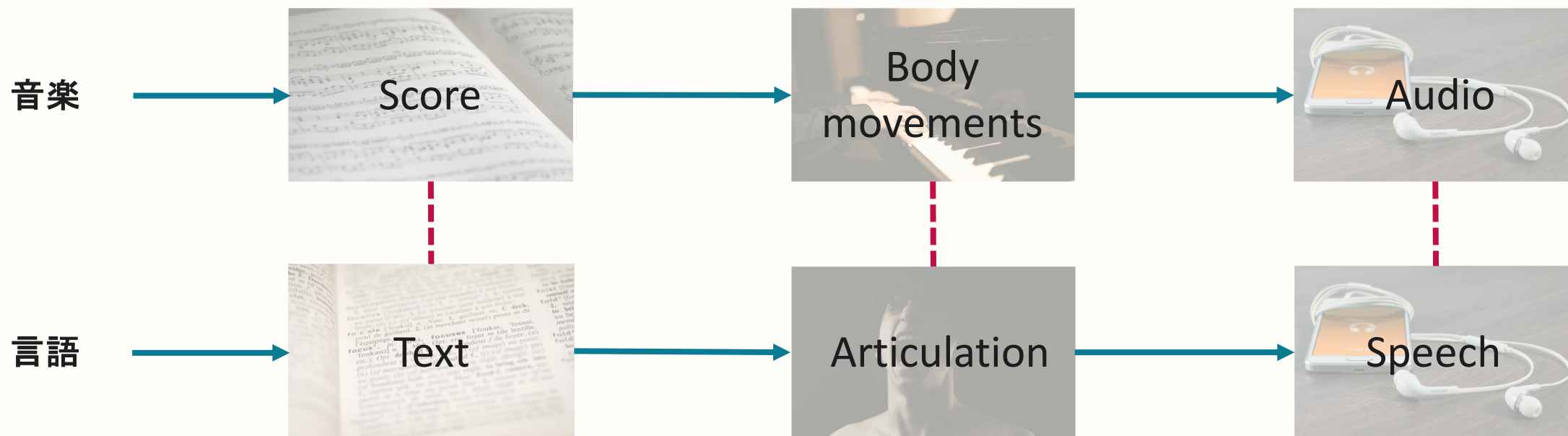
## Audio-to-Score の例

- Automatic Music Transcription (AMT): 自動採譜
  - コード認識、テンポ/ビート認識、演奏表現の認識等も個々にタスクとして存在



# 音楽と言語の対応

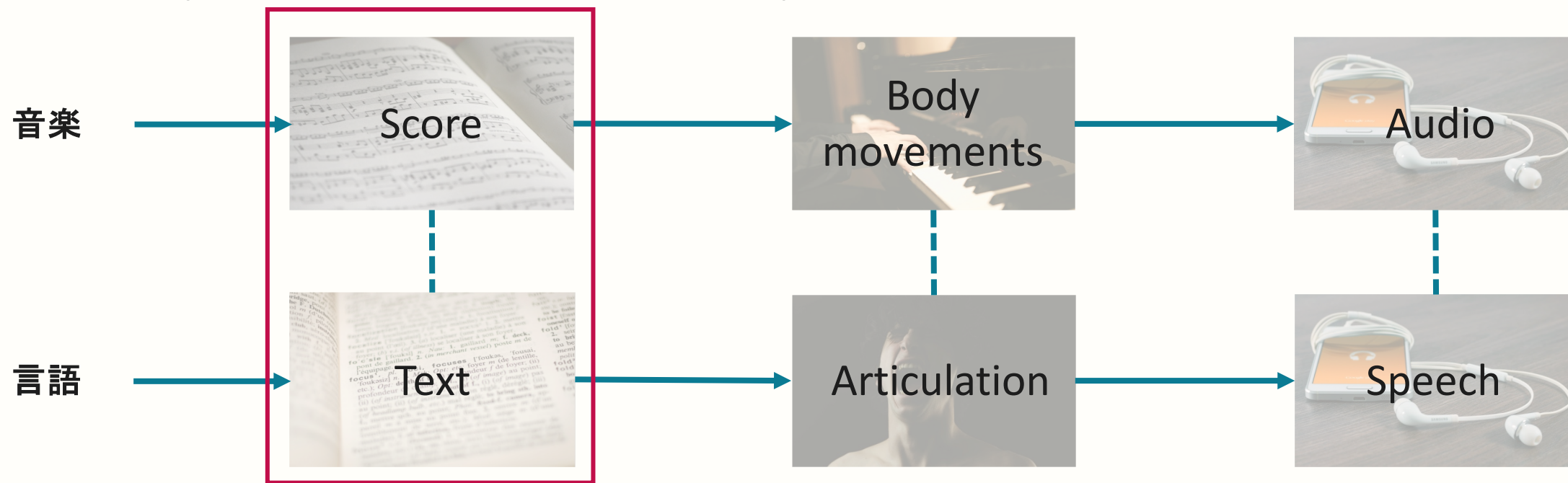
- 音楽における「楽譜→身体動作→オーディオ」という過程は言語における「テキスト→発話→音声」という過程と対応
- タスクや手法的にも対応
  - 楽譜生成  $\equiv$  テキスト生成、自動採譜 (Audio-to-Score)  $\equiv$  音声認識 (Speech-to-Text)、Music Synthesis (Score-to-Audio)  $\equiv$  Speech Synthesis (Text-to-Speech)、楽譜分類  $\equiv$  テキスト分類





# 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」という過程と対応
- タスクや手法的にも対応
  - 楽譜生成 ≡ テキスト生成、自動採譜 (Audio-to-Score) ≡ 音声認識 (Speech-to-Text)、Music Synthesis (Score-to-Audio) ≡ Speech Synthesis (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
- 手法の転用をする際にはこれらの相違点を計算機上でどう扱うべきかを考えることが重要
  - 構造の共通点・相違点が DNN 内部でどう表現されているかを比較出来たら面白そう

	音楽	言語	説明
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

# スパースオートエンコーダを用いた 大規模言語モデルのチェックポイント横断分析

---

稲葉 達郎<sup>1,2</sup> 乾 健太郎<sup>3,4,5</sup> 宮尾 祐介<sup>6,2</sup> 大関洋平<sup>6</sup>

Benjamin Heinzerling<sup>5,4\*</sup> 高木 優<sup>2\*</sup>

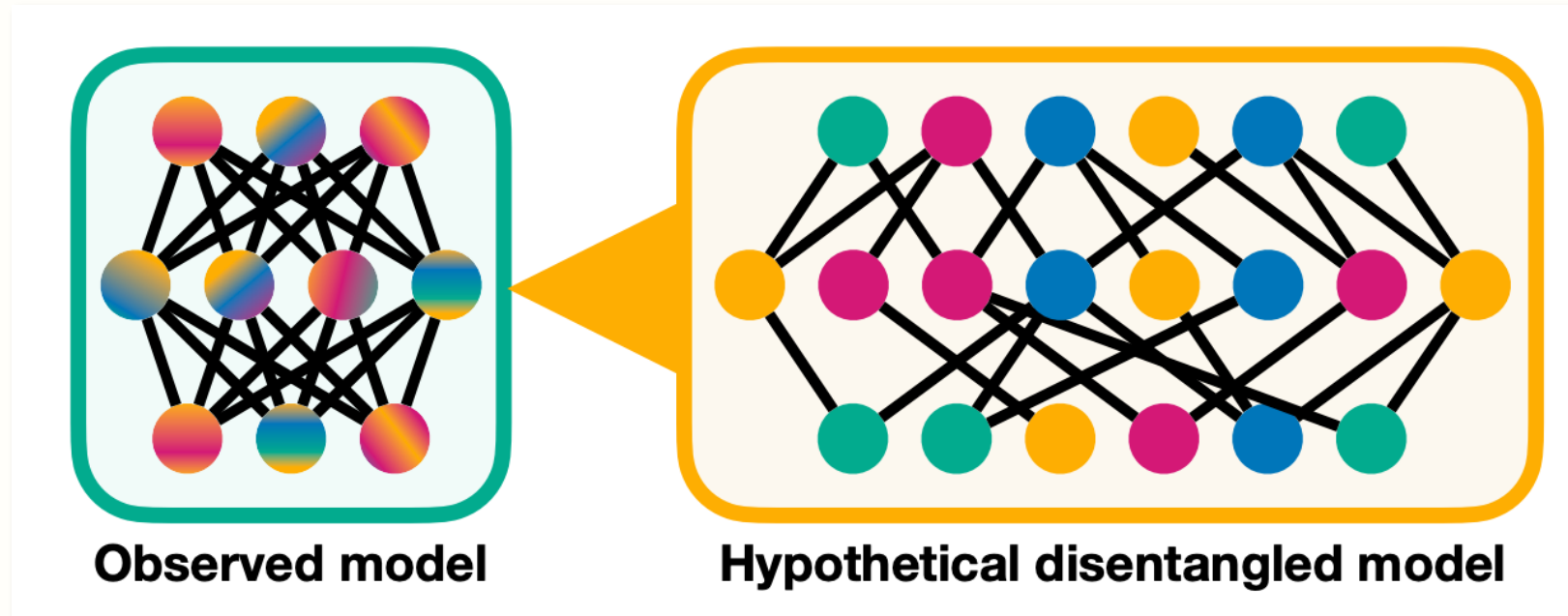
<sup>1</sup> 京都大学 <sup>2</sup> NII LLMC <sup>3</sup> MBZUAI <sup>4</sup> 東北大学 <sup>5</sup> 理化学研究所 <sup>6</sup> 東京大学

# 研究概要

- 大規模言語モデルの内部表現が含む情報が学習経過に伴いどう変化するか
- スパースオートエンコーダ (SAE) をLLM の各チェックポイントごとに個別に学習し分析することで、LLM の内部表現が含む情報がどう変遷していくか
- 結果
  - 言語を個別に学習後、言語間の対応関係を習得していそう
  - トークンレベルの知識を学習後、抽象度の高い概念レベルの知識を習得していそう

# 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

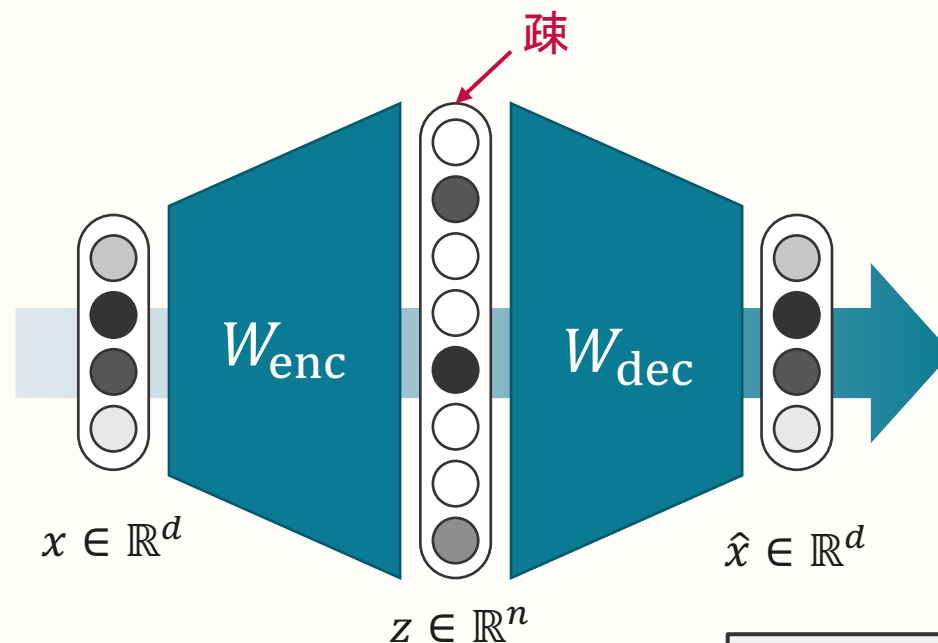
## 定式化

$$z = \text{ReLU}(W_{\text{enc}}(x - b_{\text{pre}}))$$

$$\hat{x} = W_{\text{dec}}z + b_{\text{pre}}$$

## 損失

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



$$\begin{aligned} x, b_{\text{pre}} &\in \mathbb{R}^d, z \in \mathbb{R}^n, d < n \\ W_{\text{enc}} &\in \mathbb{R}^{n \times d}, W_{\text{dec}} \in \mathbb{R}^{d \times n} \end{aligned}$$

※  $b_{\text{pre}}$  と ReLU は省略して図示

# 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

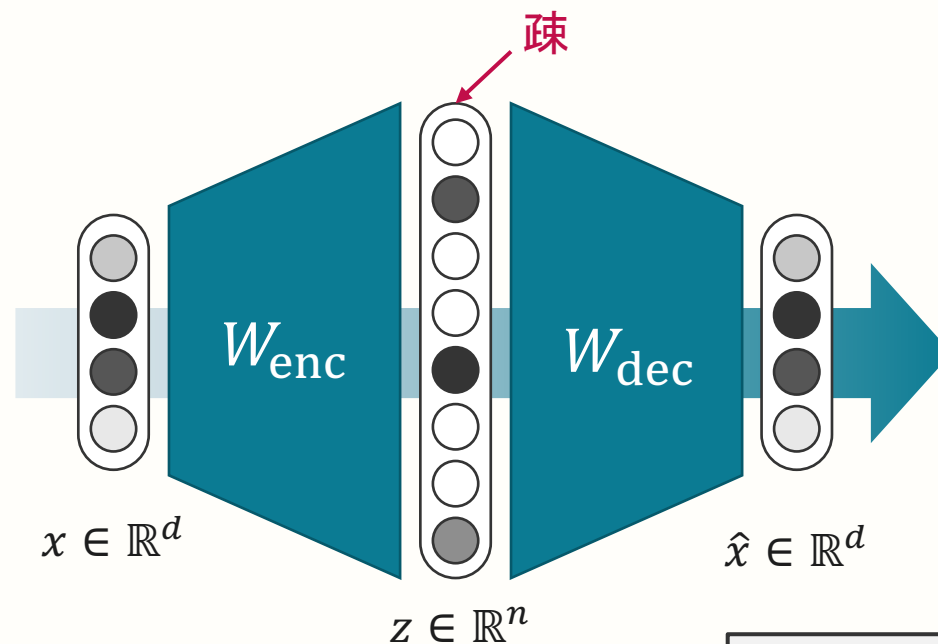
## 定式化

$$z = \text{ReLU} \left( W_{\text{enc}}(x - b_{\text{pre}}) \right)$$

$$\hat{x} = W_{\text{dec}}z + b_{\text{pre}}$$

## 損失

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



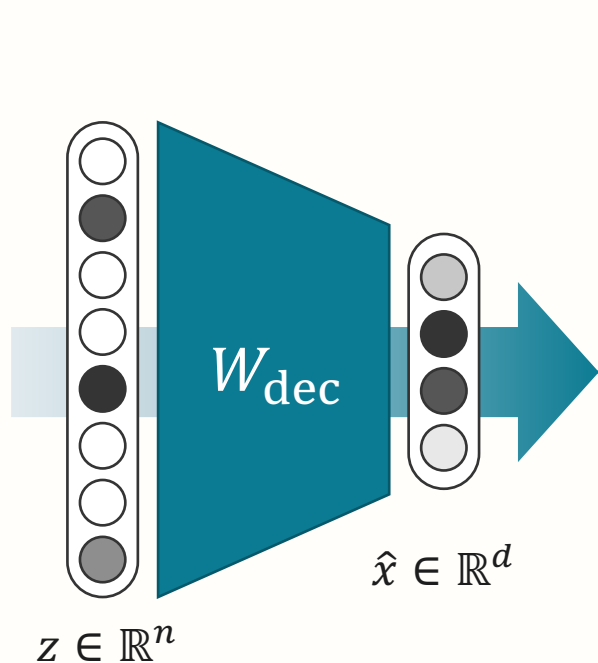
$$\begin{aligned} x, b_{\text{pre}} &\in \mathbb{R}^d, z \in \mathbb{R}^n, d < n \\ W_{\text{enc}} &\in \mathbb{R}^{n \times d}, W_{\text{dec}} \in \mathbb{R}^{d \times n} \end{aligned}$$

※  $b_{\text{pre}}$  と ReLU は省略して図示



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

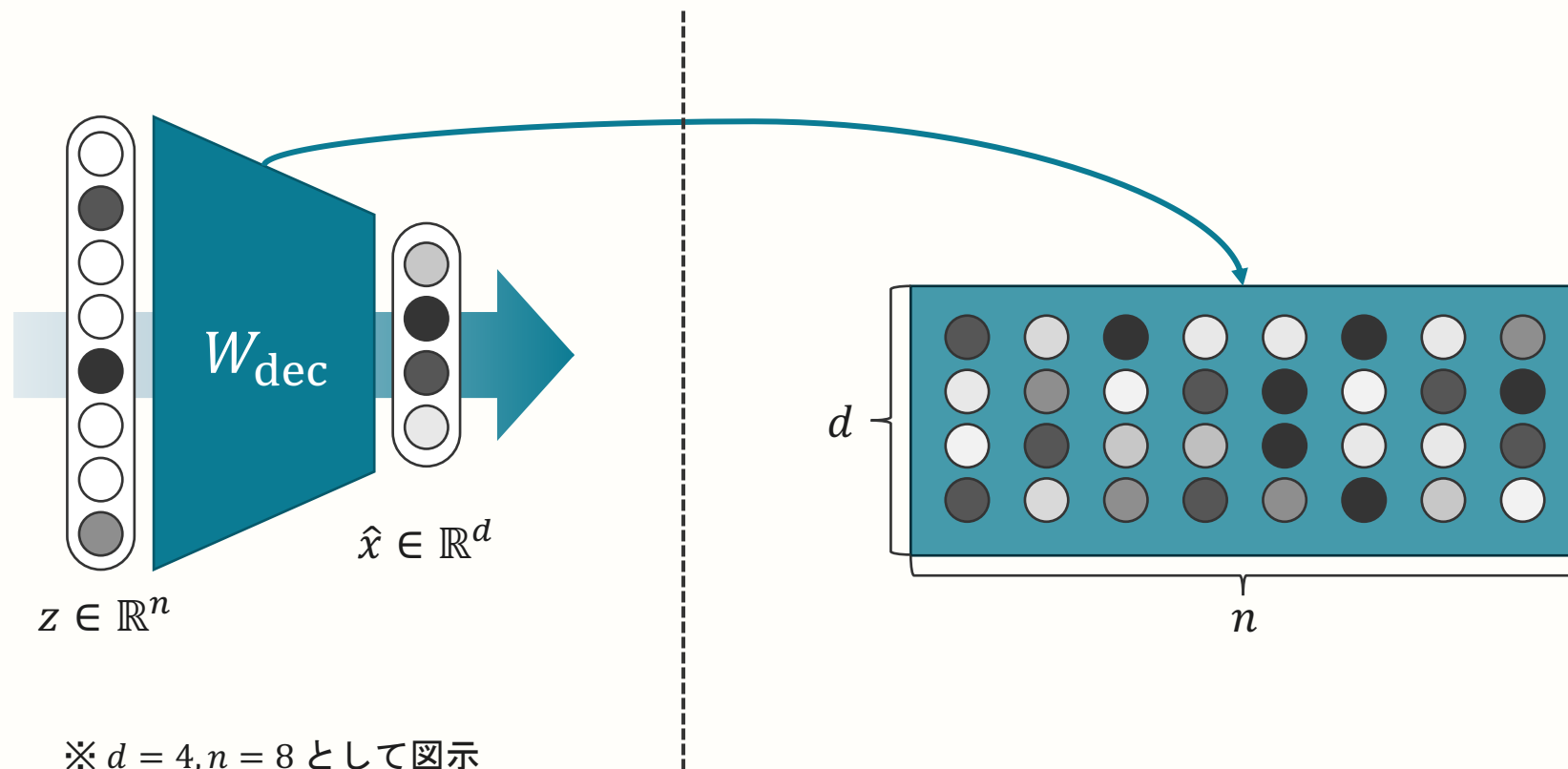
- $W_{\text{dec}} \in \mathbb{R}^{d \times n}$  を  $n$  個の  $d$  次元ベクトルと見ると、 $n$  個の  $d$  次元ベクトルからいくつかを選び、その重み付き足し合わせで入力ベクトルを再構成するネットワーク



※  $d = 4, n = 8$  として図示

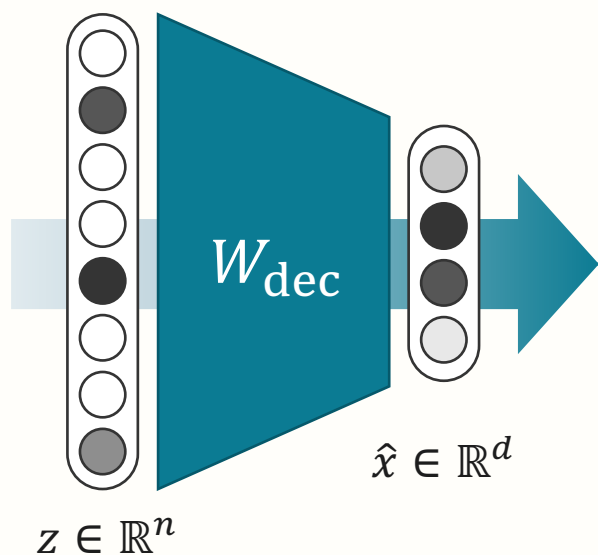
# スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

- $W_{\text{dec}} \in \mathbb{R}^{d \times n}$  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

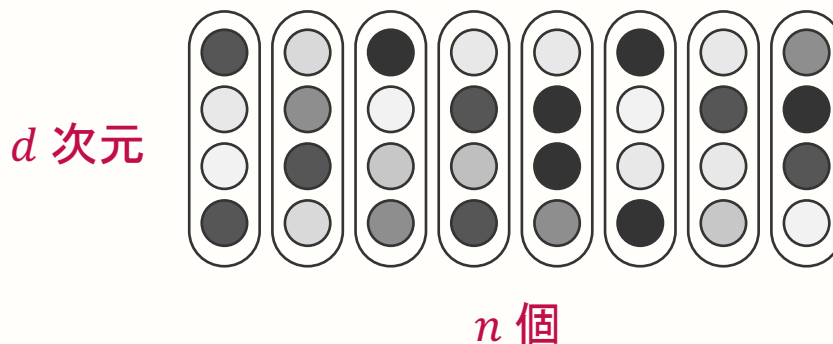


# スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

- $W_{\text{dec}} \in \mathbb{R}^{d \times n}$  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

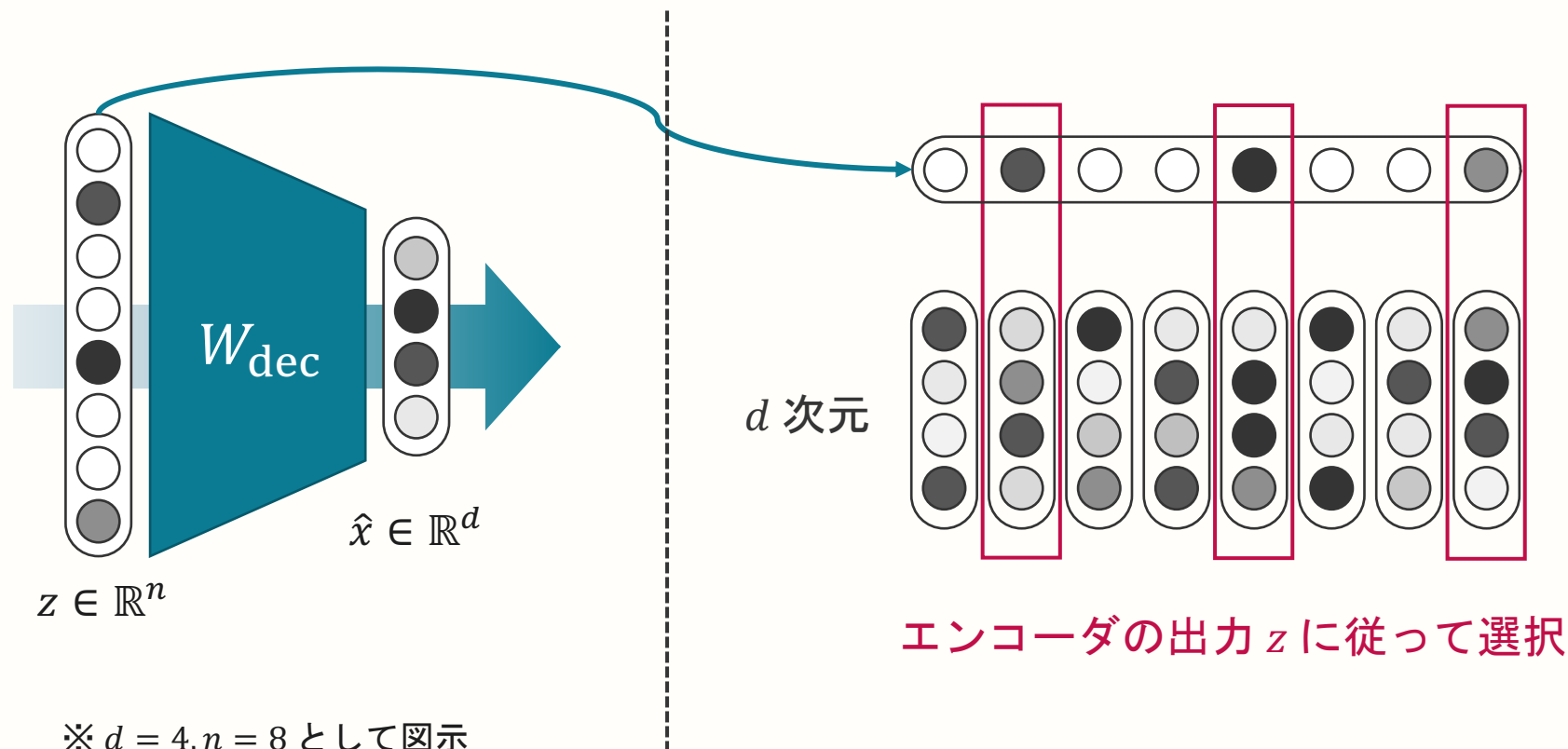


※  $d = 4, n = 8$  として図示



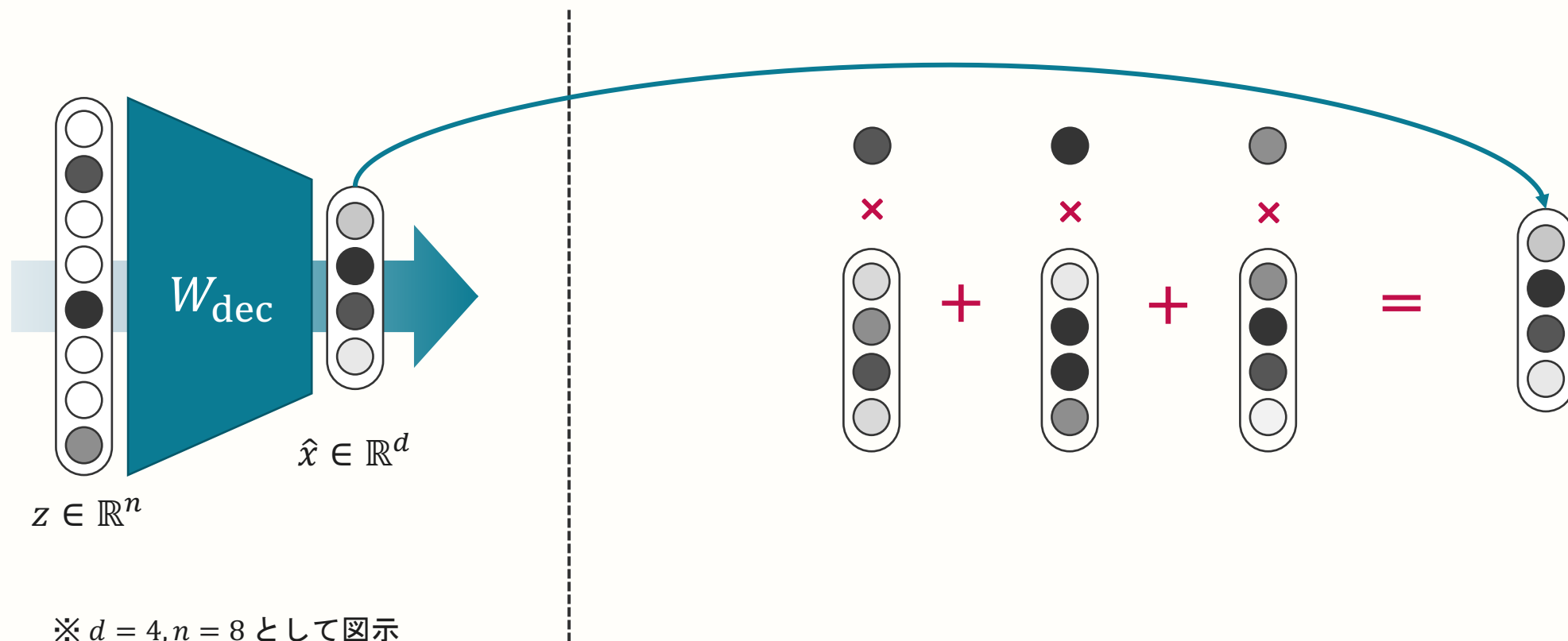
# スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

- $W_{\text{dec}} \in \mathbb{R}^{d \times n}$  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



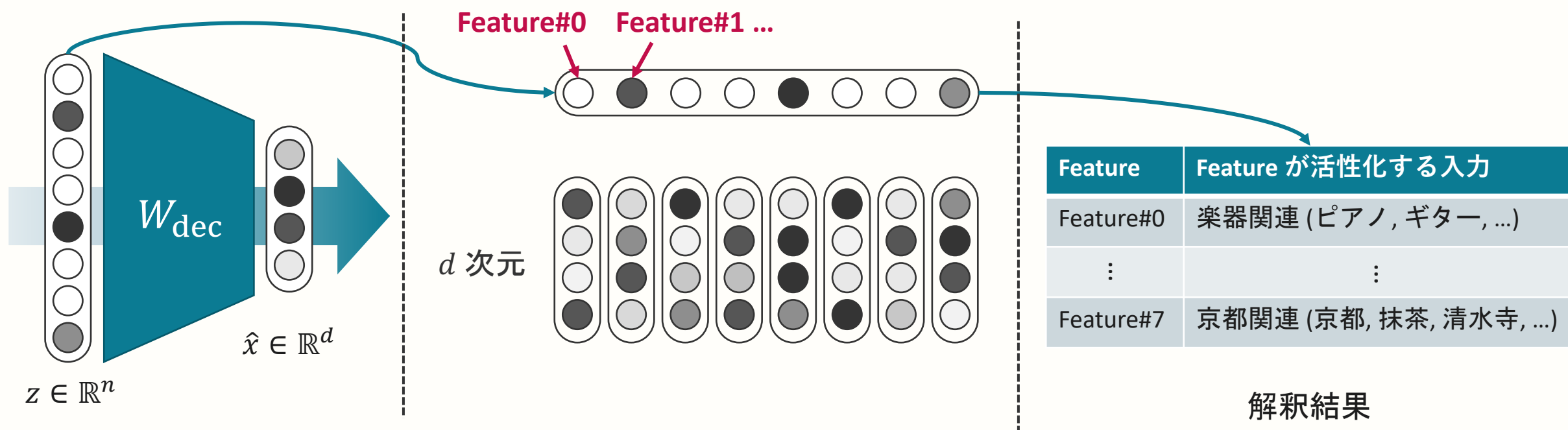
# スパースオートエンコーダ (SAE) [Olshausen+, 97; Huben+, 23]

- $W_{\text{dec}} \in \mathbb{R}^{d \times n}$  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際に選ばれるかから、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



※ 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

# 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

$$z = \text{ReLU} \left( W_{\text{enc}}(x - b_{\text{pre}}) \right)$$

$$\hat{x} = W_{\text{dec}}z + b_{\text{pre}}$$

## 損失

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



## 定式化

$$z = \text{TopK} \left( W_{\text{enc}}(x - b_{\text{pre}}) \right)$$

$$\hat{x} = W_{\text{dec}}z + b_{\text{pre}}$$

## 損失

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

# 予備実験: 設定

学習済み LLM-jp-3-1.8B で TopK-SAE のパラメータ調整を行う

## データ

- LLM-jp-corpus v3 の 日本語 wiki (50%) と 英語 wiki (50%)
- 計 165M トークンを 8:1:1 で訓練, 検証, テスト用に
- 65トークン分を LLM に入力  
→ [BOS] トークンを除いた64トークン分の **12 層目**の表現を SAE に **L2正規化**して入力
- バッチサイズ: 32768

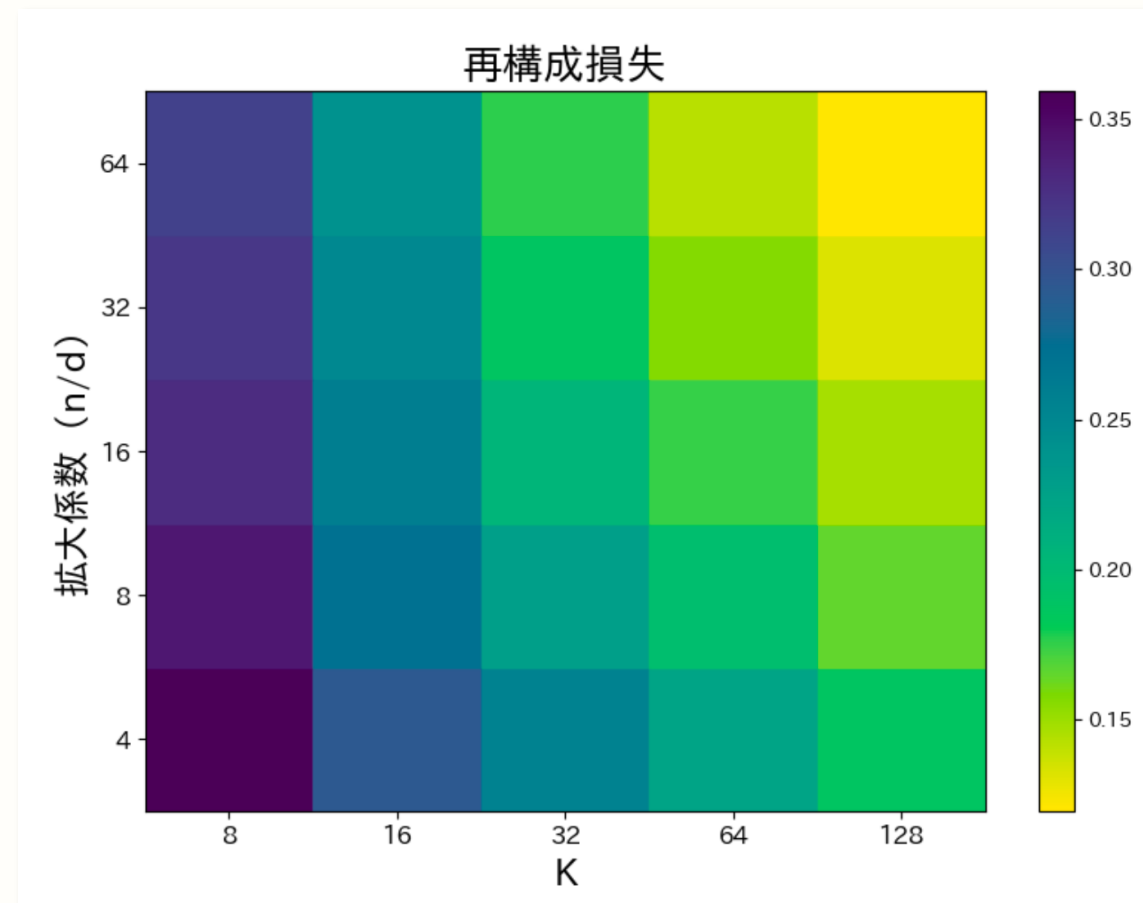
## TopK-SAE のパラメータ

- 拡大係数  $(\frac{n}{d}) = \{4, 8, 16, 32, 64\}$ ,  $K = \{8, 16, 32, 64, 128\}$



# 予備実験: 再構成精度

- 拡大係数が大きく、 $K$ が大きくなるほど再構成精度が向上
- 学習率はグリッドサーチを用いて最適なものを選択
  - 学習率にかなり再構成精度が左右される
- 人にとって解釈しやすい結果が得られることと再構成精度が高いことはまた別の問題  
[Leask+, 25; Menon+, 24]
  - $n, K$  が大きいほど概念が分離
  - 小さいと Polysemantic に



# 予備実験: Feature

- 十分に解釈可能性の高い活性化パターンが観察できた
  - トークンレベルや、概念レベル (例: 9番目の定義に関連・21番目の例示に関連) のパターンが見られた
- 日英の対応**をとっている Feature も一定数あった (例: 26番目の dark とダーク等)

F <sub>ckpt=988240</sub> #00009	<ul style="list-style-type: none"><li>ここで言う「都市」には</li><li>, where fluency is defined as linguistic</li><li>"Arbitrary" here means that the</li></ul>
F <sub>ckpt=988240</sub> #00016	<ul style="list-style-type: none"><li>特有の臭気のある白色個体で、</li><li>物で、白色の粉末である</li><li>It is a colorless liquid with a smell reminiscent</li></ul>
F <sub>ckpt=988240</sub> #00006	<ul style="list-style-type: none"><li>about a mile (1.6 km) east of the</li><li>36.6 square miles (94.8 km), of</li><li>Located 4 miles north from Wasilla</li></ul>
F <sub>ckpt=988240</sub> #00007	<ul style="list-style-type: none"><li>旧表記（数え年）にて表記。</li><li>0から数え始め、1</li><li>一つに数えられることがある。</li></ul>
F <sub>ckpt=988240</sub> #00017	<ul style="list-style-type: none"><li>津海道（しんかい-どう）は</li><li>かけての津藩の藩士である。</li><li>は岡山県御津郡にあった村。</li></ul>
F <sub>ckpt=988240</sub> #00021	<ul style="list-style-type: none"><li>for cyclists (e.g. cyclist-only paths</li><li>itself, for example on signage.</li><li>languages spoken, such as Belgium</li></ul>
F <sub>ckpt=988240</sub> #00026	<ul style="list-style-type: none"><li>よりはダーク・ファンタジー</li><li>Darkened Skye is a</li><li>baryonic dark matter is hypothetical dark matter</li></ul>
F <sub>ckpt=988240</sub> #00039	<ul style="list-style-type: none"><li>に上海美術映画作成所より制作された</li><li>The game was developed by Beam Software</li><li>It was part of Mutual Film Corporation's</li></ul>
F <sub>ckpt=988240</sub> #00041	<ul style="list-style-type: none"><li>is located nine kilometers south-west of</li><li>airport located 13 km northwest of</li><li>airport located seventeen miles (</li></ul>

# 本実験: 設定

予備実験の結果を踏まえて、**チェックポイント横断**で実験

- 分析対象は LLM-jp-3-1.8B のチェックポイント 6 箇所 (10, 100, 1000, 10000, 100000, 988240)
- TopK-SAE のパラメータは  $\frac{n}{d} = 16, K = 32$
- その他のデータなどの設定は予備実験と同様

# 本実験: Feature

- (a) ckpt=100, (b) ckpt=10000, (c) ckpt=988240 での活性化パターン例

	Feature番号	Featureが活性化する文章例	言語傾向 (4.3章)	意味粒度 (4.4章)
(a)	F <sub>ckpt=100</sub> #00002	<ul style="list-style-type: none"> <li>・ ) は、「日本の貴婦人</li> <li>・ または HMG-CoA レダクターゼ</li> <li>・ called radiological pollution, is</li> </ul>	日英混合	無関連
	F <sub>ckpt=100</sub> #00004	<ul style="list-style-type: none"> <li>・ から20世紀前半にかけて</li> <li>・ は日本の防衛官僚。</li> <li>・ investigations are performed by geotechnical</li> </ul>	日英混合	無関連
(b)	F <sub>ckpt=10000</sub> #00004	<ul style="list-style-type: none"> <li>・ dorsalis), also known as the scrub</li> <li>・ regnans, known variously as</li> <li>・ nerve) also known as the fourth</li> </ul>	英語	トークンレベル: 「known」
	F <sub>ckpt=10000</sub> #00009	<ul style="list-style-type: none"> <li>・ 石油生産設備から</li> <li>・ 冷暖房設備、冷凍冷蔵設備、動力設備又は</li> <li>・ のプラント設備を</li> </ul>	日本語	トークンレベル: 「設備」
(c)	F <sub>ckpt=988240</sub> #00009	<ul style="list-style-type: none"> <li>・ ここで言う「都市」には</li> <li>・ , where fluency is defined as linguistic</li> <li>・ "Arbitrary" here means that the</li> </ul>	日英混合	概念レベル (同義): 「定義」
	F <sub>ckpt=988240</sub> #00016	<ul style="list-style-type: none"> <li>・ 特有の臭気のある白色個体で、</li> <li>・ 物で、白色の粉末である</li> <li>・ It is a colorless liquid with a smell reminiscent</li> </ul>	日英混合	概念レベル (意味的共通性): 「物質の特性」

# 本実験: 活性化パターンの分析

Feature の傾向  $\equiv$  LLM 内部の表現が含む傾向のある知識・情報

- なぜなら、Feature は LLM 内部表現の再構成に使われるので、その表現が含みがちな知識・情報ほど Feature として頻出する

そこで、Feature の傾向を**言語傾向**と**意味粒度**の2軸で分析し、LLM の内部表現が学習経過でどのような知識・情報を含むようになっていくかを分析する

# 本実験: 活性化パターンの分析

## 言語傾向: 自動で全 Feature を分類

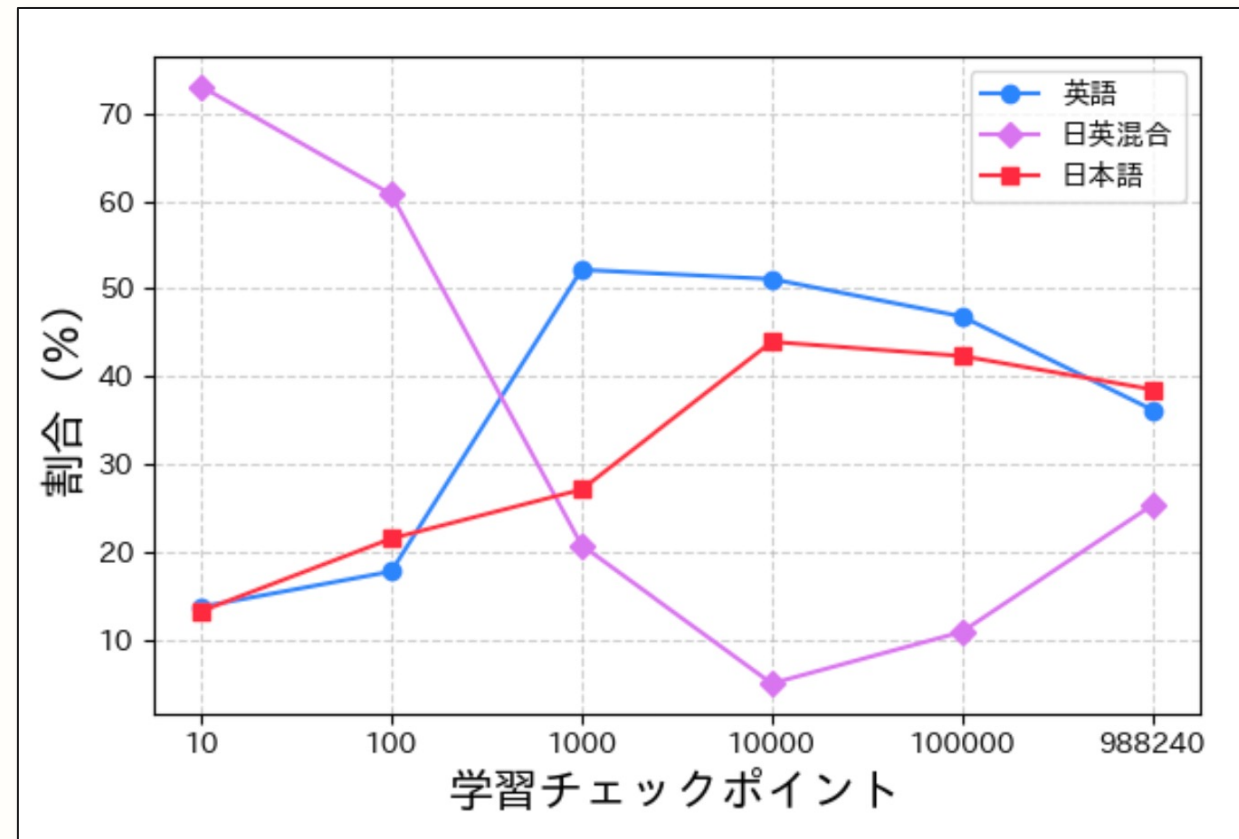
- 日本語 or 英語 or 日英混合 のどの言語に活性化するか

## 意味粒度: 手動で 100 Feature ずつ分類

- トークンレベル: 同一のトークンに活性化 (例: **猫**と**猫**)
- 概念レベル (同義): 同一の意味を表すトークンまたは文に活性化 (例: **猫**と**ネコ**)
- 概念レベル (意味的共通性): 意味的共通性を持つトークンまたは文に活性化 (例: **猫**と**犬**)
- 無関係: 解釈可能な関係性が見られない時

# 本実験: 活性化パターンの言語傾向推移

- 学習初期は日英混合 Feature が大半
  - 無作為なトークンに活性化
- 日英混合 Feature は学習経過で一旦減り学習後期に再び増加
  - 日英間で同一の意味を持つトークンや文章に対して活性化
- 英語Feature と日本語Feature は学習中期に増加
  - 同一言語内における類似した意味を持つトークンに対して活性化



# 本実験: 活性化パターンの言語傾向推移

- 学習初期は日英混合 Feature が大半
  - 無作為なトークンに活性化
- 日英混合 Feature は学習経過で一旦減り学習後期に再び増加
  - 日英間で同一の意味を持つトークンや文章に対して活性化
- 英語Feature と日本語Feature は学習中期に増加
  - 同一言語内における類似した意味を持つトークンに対して活性化

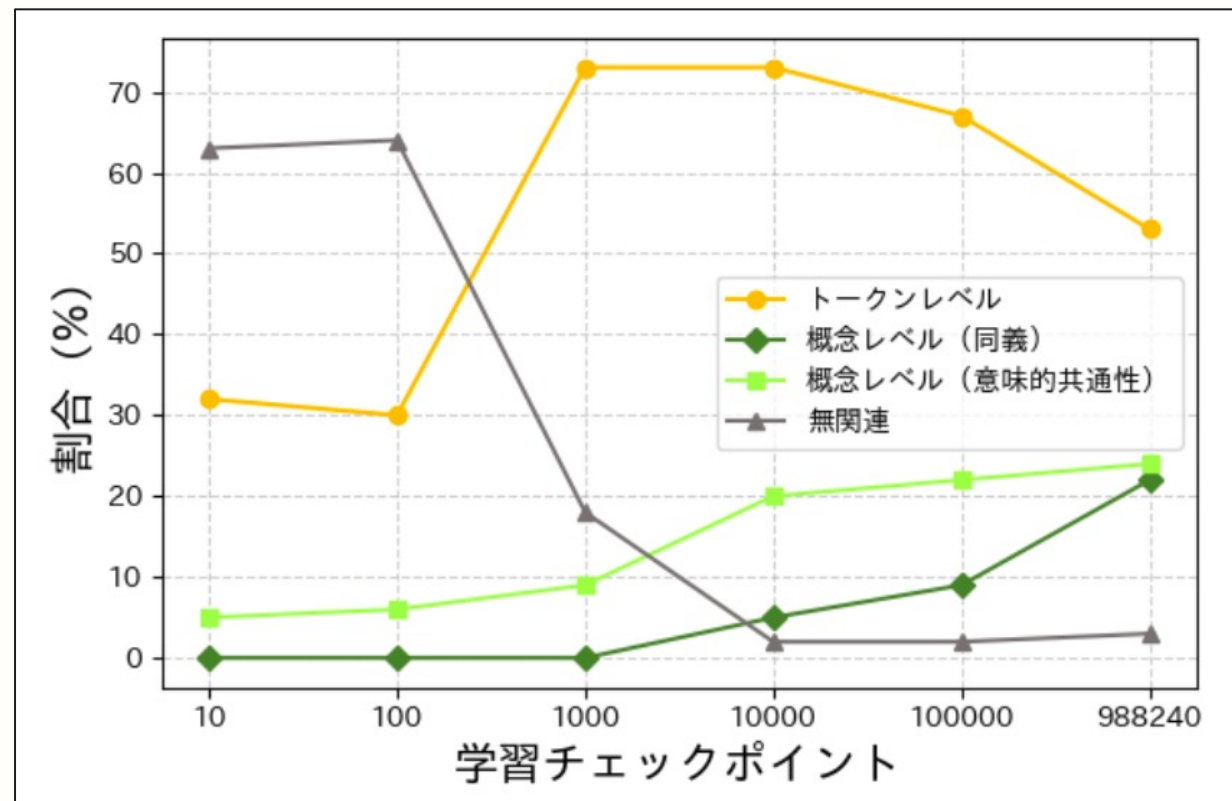


1. **学習初期から中期**にかけて**言語別**にトークンや文章の意味を習得
2. **学習中期から後期**にかけてトークンや文章の**言語間での対応関係**を習得




# 本実験結果: 活性化パターンの意味粒度推移

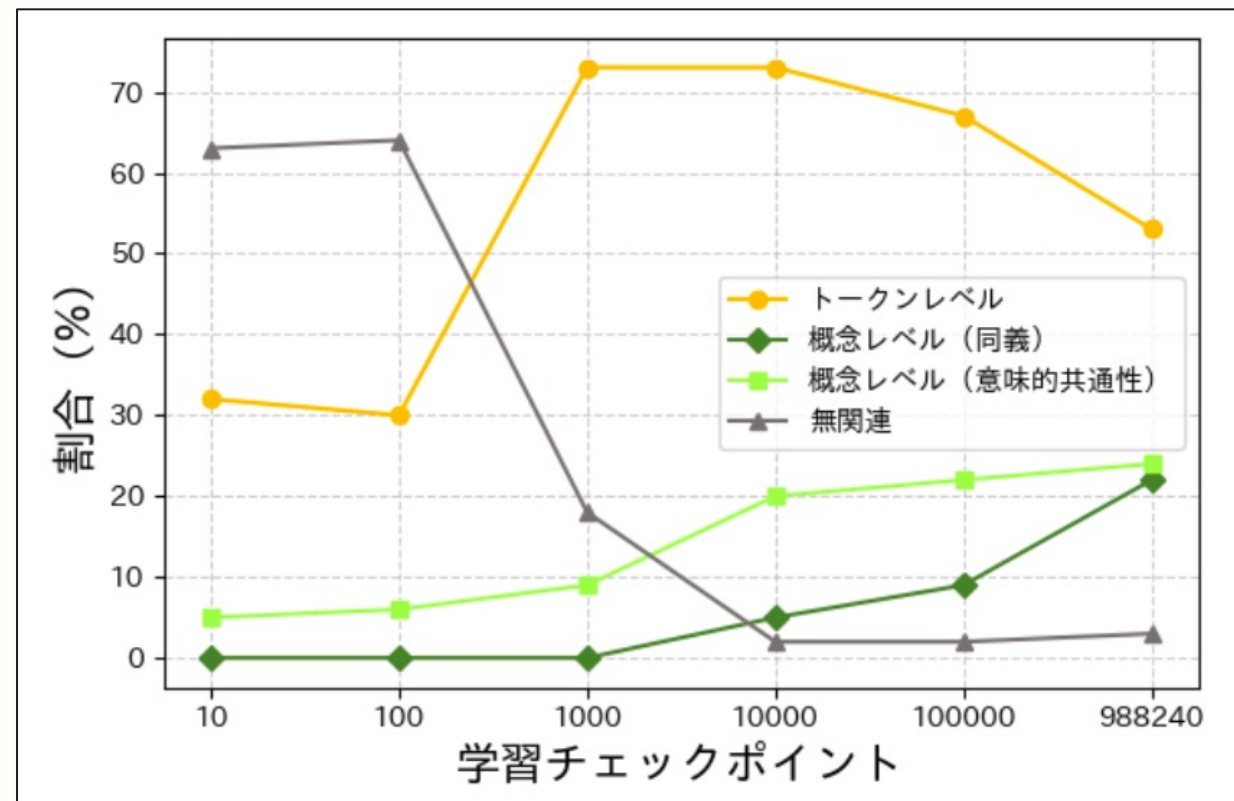
- 学習初期は無関連 Feature が大半
- 学習中期にトークンレベル Feature が増加
- 学習後期にかけ概念レベル Feature 増加



# 本実験結果: 活性化パターンの意味粒度推移

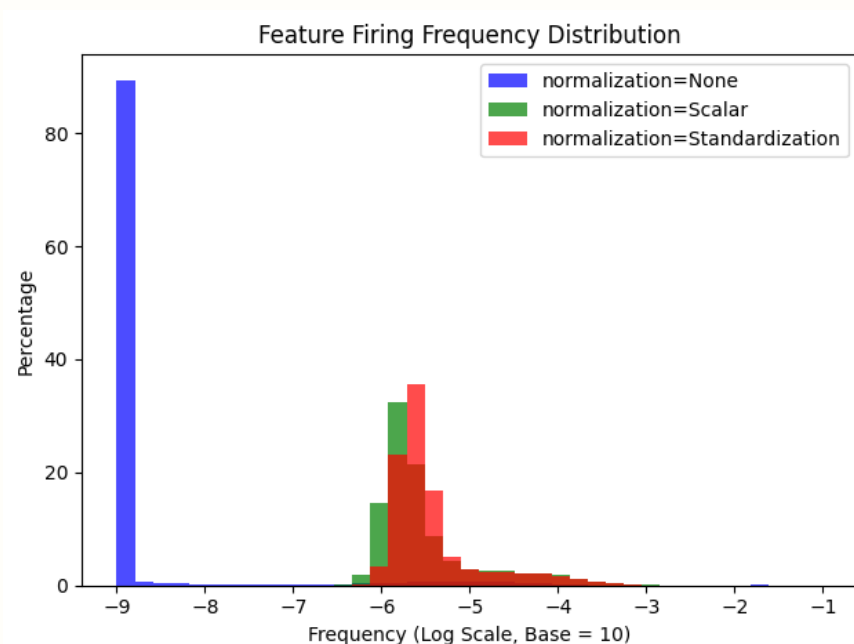
- 学習初期は無関連 Feature が大半
- 学習中期にトークンレベル Feature が増加
- 学習後期にかけ概念レベル Feature 増加

- 
1. 学習初期から中期にかけて**トークンレベル**の知識を習得
  2. 学習中期から後期にかけて**概念レベル**の知識を習得



# 余談: 正規化は妥当か

- 今回の設定では LLM の中間表現を SAE に入力する前に正規化している
- ノルムの大きさも何かしらの意味 (例: 意味強度) を持つはずで、それを平らにしてしまうのは妥当なのか

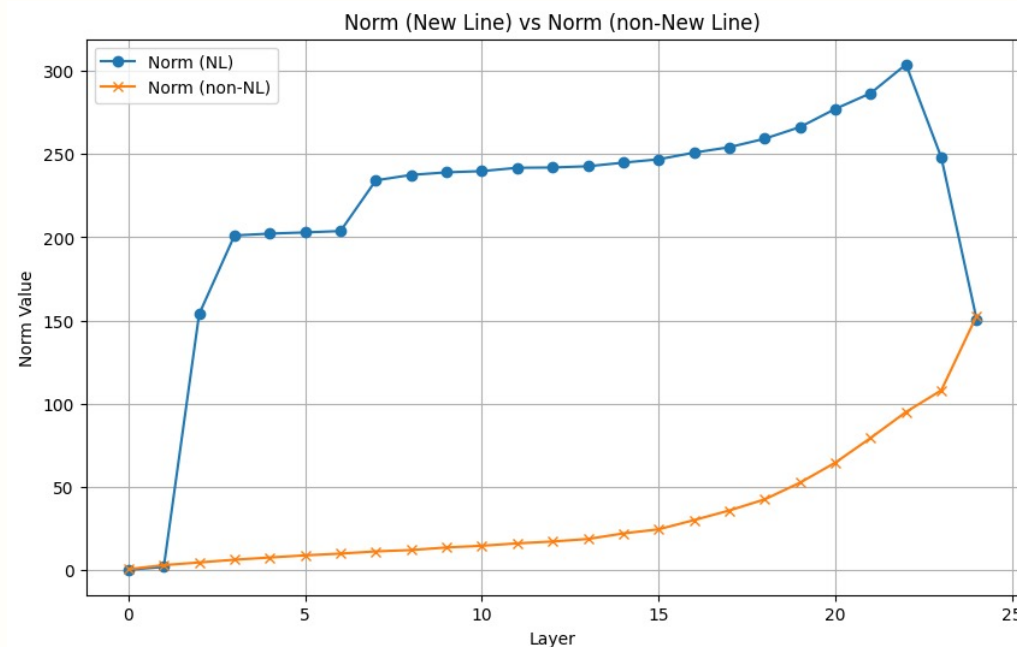


- None: 正規化なし
- Scalar: 長さ1
- Standardization: 平均0、分散1

正規化なしだと Dead Feature が多量に出現する  
→ ノルムの大きい Feature が学習を支配している？

# 余談: 正規化は妥当か

- 今回の設定では LLM の中間表現を SAE に入力する前に正規化している
- ノルムの大きさも何かしらの意味 (例: 意味強度) を持つはずで、それを平らにしてしまうのは妥当なのか
- 改行トークン (“\n”, Newline; NL) はノルムが異常に大きくなる
- レイヤー間でもノルムに差がある
  - 正規化をすると、レイヤー間での比較に影響
  - 今回はレイヤー間での比較を行っていないからよかったが、考慮する必要がありそう



# まとめ

- スパースオートエンコーダを用いてチェックポイント横断で LLM を分析した
- その結果以下の可能性が示唆された
  - LLM は個別言語でのトークンや文章の意味を習得し，その後言語間の対応関係を理解すること
  - トークンレベルの知識を習得した後に概念レベルの知識体系を構築している

## 今後の展望

- (1) 意味粒度判別の自動化
- (2) 他の評価軸の検討
- (3) 言語モデルの様々な能力（例：多言語能力，多段推論能力等）が発現するタイミングに焦点を当てた分析
- (4) 異なるモデルサイズ，多言語モデルでの実験