二値化スパースオートエンコーダ

趙羽風[1]　楊昊霖[2]　Brian M. Kurkoski[1]　井之上直也[1,3]
[1]北陸先端科学技術大学院大学　[2]シカゴ大学　[3]理化学研究所　yfzhao@jaist.ac.jp

概要

LLM の内部状態から原子的特徴を抽出しその動作原理を解釈する試みが数多く行われてきたが，従来法は単一サンプルに対して正則化を課したオートエンコーダに依存し，サンプル間での**全体的**スパース性を保証できない．その結果，不活性・高頻度特徴が生成され，スパース性が損なわれる．本研究では，特徴のスパース性を高めるため，隠れ活性のミニバッチ単位でエントロピーを最小化する新しい自己符号化器を提案する．効率化のため，活性をステップ関数で 1 ビットに離散化し，勾配推定によって逆伝播を実現する．この手法を二値化スパースオートエンコーダ（BAE）と呼び，特徴集合のエントロピー削減とスパース分解への有効性を示す[1)]．

1　はじめに

LLM の内部表現から数値的要素（**特徴**）を個別に抽出・分離するための既存の手法として，スパースオートエンコーダ（SAE） [1] のように，訓練時に L_1 正則化などの制約を課して隠れ表現の活性を抑え，特徴を**サンプルごとに**自動的に原子化する方法が一般的である．しかし，このような手法は**全体的**なスパース性を保証しないため，頻繁に活性化される（dense）特徴と，全く活性化されない（dead）特徴が混在する傾向がある [2, 3, 4]．これは機械論的解釈可能性研究におけるスパース性仮定 [5] と矛盾し，サンプル間で広範に活性化する特徴は一貫した意味解釈を困難にし，さらに dead 特徴の存在によってパラメータ効率も低下する．

したがって本研究では，上述の問題を解決するために，訓練インスタンスの**ミニバッチ**間における情報理論的制約を導入した二値化スパースオートエンコーダ（BAE）を提案する．図 1 に示すように，我々はミニバッチ全体の隠れ層のエントロピーを最小化する目的関数を設計し，特徴間の共変動を抑制しつつ，全体的なスパース性を強制することで頻繁に活性化される特徴を抑える．しかし，一般的な隠れ層は実数ベクトルであり，そのエントロピーの計算には高次元の数値積分が必要となるため，計算複雑性が指数的に増大してしまう [6]．そこで我々は隠れ層を二値化して丸め，この二値ベクトル上でエントロピーを計算することにより，計算コストを大幅に削減し，さらに勾配推定法 [7] を用いて，この丸め操作に対しても逆伝播を可能にする．

本研究では BAE の有効性を応用的観点から実証的に示す．本研究の貢献は次の二点に要約される．

(1) 隠れ状態集合からのエントロピーの効率的推定．入力集合を再構成するためのエントロピーを，隠れ層のエントロピーとして大幅に低コストで推定でき，これは LLM 内部の動作を理解する上で重要な指標となる．異なる真値エントロピーをもつ人工データセット上の実験により，この推定の精度を確認した．さらに，通常の言語モデルにおける前向き計算過程の中に存在するエントロピー変動を分析し，各層における情報帯域幅や暗黙的なコンテキストウィンドウを明らかにした．

(2) スパースな特徴抽出．通常の SAE と同様に，オートエンコーダの第 2 層における線形ベクトル（しばしば *dictionary* と呼ばれる）は，入力から抽出された原子的特徴として機能する．SAE では，サンプル単位の正則化に起因して特徴が密に活性化される [2, 3, 4]．これに対し，BAE はミニバッチ単位のエントロピーに基づく学習目的と，各チャネルの情報利得に基づく一貫した活性スケーリングにより，dense 特徴と dead 特徴の両方を効果的に抑制し，より多くの活性で可解釈な特徴を抽出する．さらに，我々は従来の特徴解釈手法 [8, 9] を改良し，LLM による数値トークンの不安定な処理 [10] を回避することで，より頑健な自動特徴解釈を実現した．

2　二値化スパースオートエンコーダ

本研究では図 1 に示す BAE を提案する．その計算は以下のように行われる．

1）　本論文の完全版は https://arxiv.org/pdf/2509.20997 からアクセスできる．

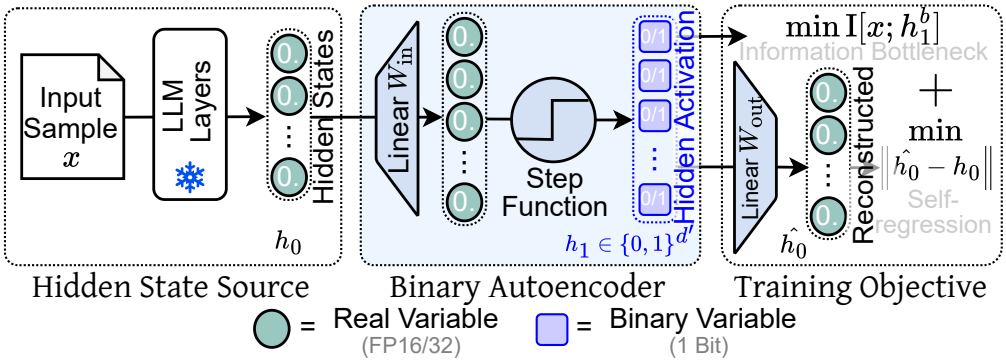

図 1　BAE の前向き計算では，LLM 層の隠れ状態 h_0 を W_{in} で写像し，ステップ関数により二値化して h_1 を得る．その後，W_{out} で $\hat{h}_0$ を復元し，$\hat{h}_0$ を自己回帰損失，h_1 をエントロピー損失に入力する．

前向き計算． LLM から得られる入力隠れ状態 $h_0 \in \mathbb{R}^d$ に対して，BAE モデル $\mathcal{F}$ は出力 $\mathcal{F}(h_0)$（または $\hat{h}_0 = \mathcal{F}(h_0)$ と表記）を次のように計算する．

$$\mathcal{F}(h_0) = \Gamma\left(h_0 W_{\text{in}}\right) W_{\text{out}} + b, \tag{1}$$

ここで，$W_{\text{in}} \in \mathbb{R}^{d \times d'}$ は入力 h_0 を d' 次元に線形分解する行列であり，$b \in \mathbb{R}^d$ は隠れ状態の異方性 [11, 12, 13, 14] を再構成するためのバイアス項である．Γ は二値化関数であり，$\mathbb{R}^{d'}$ を要素ごとに隠れ層 $h_1 \in \{0,1\}^{d'}$ へ写像する．

$$\Gamma\left([x_1, x_2, \ldots, x_{d'}]\right) = \left[\gamma\left(x_1\right), \gamma\left(x_2\right), \ldots, \gamma\left(x_{d'}\right)\right],$$
$$\gamma(x) = \begin{cases} 0, x < 0 \\ 1, x \geqslant 0. \end{cases} \tag{2}$$

目標関数 1：自己回帰． バッチサイズを n_b とする隠れ状態のミニバッチ $H_0 = \left\{h_0^{(1)}, h_0^{(2)}, \ldots, h_0^{(n_b)}\right\}$ が与えられたとき，自己回帰型の学習目的関数は次の L_2 ノルムで計算される：

$$\mathcal{L}_r(H_0) = \frac{1}{n_b} \sum_{h_0 \in H_0} \|h_0 - \mathcal{F}(h_0)\|_2. \tag{3}$$

ここで，$\mathcal{F}(h_0)$ はオートエンコーダによって再構成された隠れ状態を表す．

目標関数 2：エントロピー． 隠れ層（$h_1 = \Gamma\left(h_0 W_{\text{in}}\right)$）を h_0 に対して全体的にスパースな表現へ制約するために，h_1 の周辺エントロピーを最小化する．ここで $h_1 \in \{0,1\}^{d'}$ であるため，この計算は実数空間上の数値積分を必要とせず，微分可能かつ高効率である．さらに，周辺エントロピー制約の効果を最大化するために，h_1 の共分散（対角成分を除く）にもペナルティを与え，周辺エントロピーが同時分布のエントロピーに近づくようにする．すなわち，ミニバッチ H_0 上で，エントロピーに基づく損失項を次のように定義する：

$$\mathcal{L}_e(H_0) = \alpha_e \mathrm{H}[\frac{1}{n_b} \sum_{h_0 \in H_0} \Gamma\left(h_0 W_{\text{in}}\right)] + \alpha_c \mathrm{D}[\Gamma\left(H_0 W_{\text{in}}\right)],$$
$$\text{where } \mathrm{H}[x] = -\sum_{i=1}^{d'} x_i \log_2 x_i,\ \mathrm{D}[X] = \sum_{i,j:i \neq j} |\mathrm{cov}(X)_{i,j}|. \tag{4}$$

ここで，α_e, α_c はハイパーパラメータである．したがって，全体の損失関数は次のように表される：

$$\mathcal{L}(H_0) = \mathcal{L}_r(H_0) + \mathcal{L}_e(H_0). \tag{5}$$

Γ に対する勾配推定． 二値化関数 Γ の導関数はディラックのデルタ関数となるため，損失 $\mathcal{L}_e$ から W_{in} への誤差逆伝播を可能にする必要がある．先行研究 [7, 15] に従い，Γ の勾配を平滑化関数 $x \mapsto (1+e^{-x})^{-1}$（“Sigmoid”）で要素ごとに近似する．したがって，Γ の微分は次のように推定される：

$$\frac{\partial \Gamma(x)}{\partial x} \coloneqq \Gamma(x) \odot (\mathbf{1} - \Gamma(x)), \tag{6}$$

ここで，$\mathbf{1}$ は全要素が 1 のベクトルを表し，$\odot$ はベクトルのアダマール積（要素ごとの積）を示す．

3　隠れ状態のエントロピー推定

BAE によるエントロピー推定． ニューラルネットワークの隠れ状態に対するエントロピー計算は内部機構の理解に有用だが，高次元空間ではエントロピー計算が非常に困難である [6]．これに対して BAE は，元の実数値ベクトルを要素間ができるだけ独立になるように二値ベクトル（h_1）へ分解することで，隠れ層の平均値 $\bar{h}_1$ の周辺エントロピーを用いることができ，効率的なエントロピー推定が可能となる．具体的には，ベクトル集合 $H_0 = \left\{h_0^{(i)}\right\}_{i=1}^n$ を訓練済み BAE によって符号化し，$H_1 = \left\{h_1^{(i)} = \Gamma\left(h_0^{(i)} W_{\text{in}}\right)\right\}_{i=1}^n$ を得る．h_1 は二値化されており，共分散損失によって要素間の相関が最小化

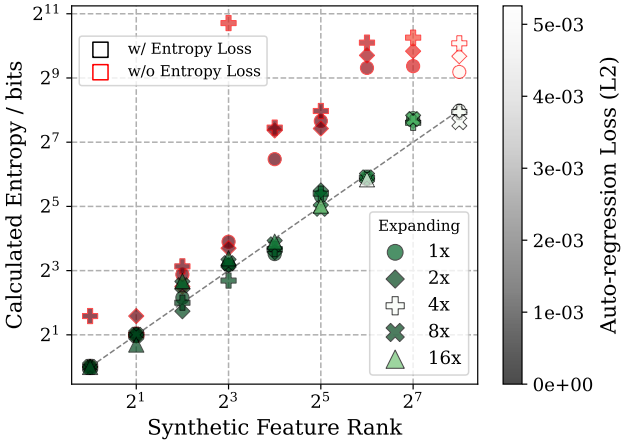

図 2　合成ベンチマーク上における BAE のエントロピー計算．横軸は既知のエントロピー値，縦軸は計算されたエントロピーを示す．緑／赤の色は $\mathcal{L}_e$ が有効かどうかを表し，不透明度は入力全体に対する $\mathcal{L}_r$ の値を示す．

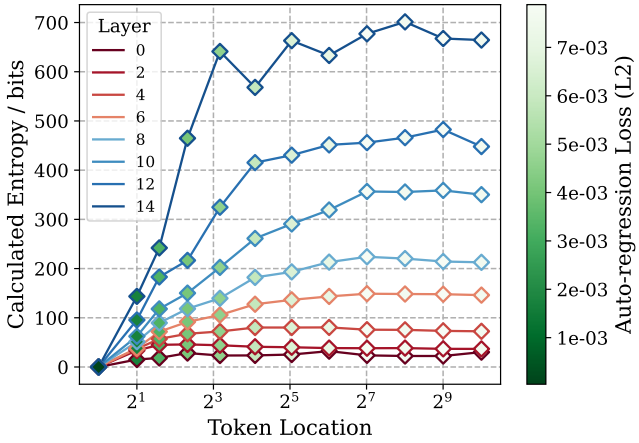

図 3　Pile および Llama 3.2-1B から特定の層とトークン位置で抽出した隠れ状態に基づいて計算されたエントロピー．曲線の色は抽出した層を示し，点の不透明度は入力全体に対する $\mathcal{L}_r$ の値を示す．

されているため，H_0 を再構成するのに必要なエントロピーを $\mathrm{H}[\bar{h}_1]$ として計算できる．ここで $\bar{h}_1$ は H_1 の各行ベクトルの平均を表す．

3.1　BAE によるエントロピー推定の精度

BAE の二値化隠れ層（h_1）のエントロピー推定精度を評価するために，既知のエントロピー値をもつ人工的なランダム方向ベンチマークを構築した（詳細は付録 A）．このベンチマーク上で BAE を訓練し，計算されたエントロピーが真の値と一致するかを確認した．図 2 に示すように，標準的な BAE（緑）は内部次元 d' に依存せず正確に r を推定する一方，α_e と α_c を 0 にした設定（赤）ではエントロピーが高く，制約付き BAE が最小エントロピーを正しく捉えていることが確認された．

3.2　LLM のエントロピーダイナミックス

本節では，隠れ状態のエントロピーを指標として，LLM の前向き計算における動的変化を追跡する．具体的には，Pile [16] から 262144 文をサンプリングし，それらを Llama 3.2-1B [17] に入力する．その後，各層において $2^{0,1,\cdots,10}$ 番目のトークンに対応する隠れ状態を抽出する．次に，各層および各位置ごとに BAE を訓練し（実験の詳細は付録 A を参照），学習済みの BAE から図 3 に示すようにエントロピーを計算し，以下のような観察を得た．

層ごとの帯域幅．特定の層から得られる隠れ状態のエントロピーは，位置インデックスの増加に伴って上昇し，最終的に一定の値で飽和する．この観察結果は次のことを示唆している：特定の層の隠れ空間を，トークン情報が伝達されるチャネル [18] としてみなすと，そのチャネルには固定された帯域幅が存在し，保持できるトークン情報量に限界があるということである．言い換えれば，Transformer は各層に暗黙的なコンテキストウィンドウを有しており，この限界を超えた情報は破棄または上書きされる．その結果として *lost-in-the-middle* 問題 [19, 20, 21, 22, 23] のような歪みが生じることが考えられる．

トークン情報ゲイン．特定のトークン位置において，より深い層の隠れ状態はより多くの情報を保持しており，またエントロピーの飽和もより遅く現れる．このことは，Transformer ブロックが逐次的に文脈化された情報を隠れ状態へ注入していることを示唆している．その結果，深い層ほど広いコンテキストウィンドウをシミュレートする傾向があり，より高次の情報集約や抽象化を伴う下流タスクの処理を容易にする．一方で，浅い層はより狭いコンテキストウィンドウに制約されているため，局所的な言語レベルの特徴に焦点を当て，それらを後段の層へ伝播させてより広範な抽象表現を形成する傾向がある．これは直感とも整合しており，既存の観察結果 [24, 25, 26, 27] とも一致している．

4　BAE による原子的特徴抽出

我々のエントロピー制約が隠れ活性 h_1 のスパース性と非相関性を促進することから，h_0 から h_1 への線形分解は LLM の隠れ状態における原子的特徴を分離できる．まず二値化された隠れ層の値 h_1 を，各特徴の活性強度を反映する連続的な指標へ変換し，各入力に対してどの特徴が活性化しているかを

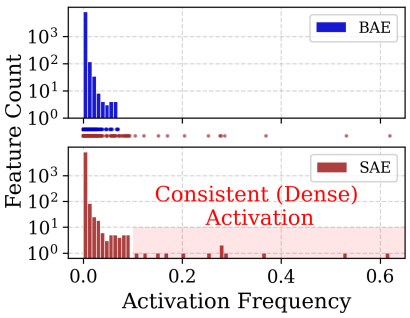

図 4　第 11 層における特徴活性頻度分布（他の層については完全版論文を参照）．

判断する．その後，これらの活性状況に基づき，改良版の手法を用いて各特徴に意味的な解釈を付与し，その信頼性を評価する．

4.1　二値化した h_1 の活性強度の推定

特定の入力で活性化した特徴を判断するためには活性強度が必要だが，BAE はそれを明示的に出力しない[2]．そのため図 5 のように，各チャネルのバースティネスを計算して二値化した h_1 を強度 β に変換する．具体的には，隠れ層活性値集合 $H_1 = \{h_1^{(i)}\}_{i=1}^{n}$ の各インスタンス（インデックス i）に対して，事前分布[3] $\bar{h}_1$ との距離 $\beta^{(i)} = \log_2 |h_1^{(i)} - \bar{h}_1|$ を計算する．ここで $\log_2$ は要素ごとの対数を意味する．この $\beta^{(i)}$ は $h_1^{(i)}$ におけるチャネルごとの活性強度を示し，値 $\beta_j^{(i)}$ が大きいチャネル j ほど，対応する特徴がより強く活性化しているとみなす．より大きい $\beta_j^{(i)}$ を持つ特徴は h_0 の再構成により多くの情報を運び，h_0 をより代表する特徴となる．$\beta^{(i)}$ の top-10 に対応する特徴を，その入力 $h_1^{(i)}$ における活性特徴と定義する．

4.2　BAE によるスパースな特徴抽出

上記の方法で得られた活性特徴に基づき，各特徴に対して LLM-as-a-judge [28] 手法を用いて自然言語による解釈を付与し，その解釈がどの程度の汎化サンプルを正しく説明できるかを可解釈スコアとして評価する．我々は従来手法を改良し，LLM による数値トークン処理の問題を回避した．この手法を ComSem と呼び，詳細は付録 B に記す．

BAE は dead 特徴を削減できる． BookCorpus [29] 上で ComSem を実行し，Pile [16] で学習された SAE 系列モデルと比較して BAE を評価する（詳細は付録 A 参照）．結果を表 1 に示す．全てのベースラインと比較して，BAE は LLM の隠れ状態から最も多くの活性特徴と完全に解釈可能な特徴を抽出できることが確認された．すなわち，BAE は SAE における dead 特徴問題を解決できる．

表 1　2 つのバックエンド LLM 上での BAE およびベースラインの評価（$d'/d = 4$）．**Feature Activated**：対応するチャネル上で，十分な数の h_0 インスタンス（8 以上，付録 B 参照）が顕著な活性強度（top-10 方法）を示した特徴数．**IS**> 0 #：ComSem スコアが 0 より大きい特徴の数．**IS**= 1 #：ComSem スコアが 1 である特徴の数．

Feat. Source	Model	Feature Activated	ComSem$_{4.1\text{-mini}}$ IS > 0 #	ComSem$_{4.1\text{-mini}}$ IS = 1 #	ComSem$_{4.1}$ IS > 0 #	ComSem$_{4.1}$ IS = 1 #
Llama 3.2-1B Layer 11 $d = 2048$	ReLU SAE	2065	1177	0.232	1380	0.260
	Top-k SAE	3417	2540	0.440	2684	0.452
	Gated ReLU SAE	1226	976	**0.531**	1026	**0.557**
	TransCoder	1794	979	0.218	1090	0.239
	BAE (ours)	**5464**	**3882**	0.360	**4140**	0.382
Llama 3.2-1B Layer 14 $d = 2048$	ReLU SAE	2528	1423	0.195	1600	0.217
	Top-k SAE	2702	1900	0.389	2004	0.418
	Gated ReLU SAE	2948	2095	**0.412**	2250	**0.435**
	TransCoder	3401	1931	0.237	2166	0.267
	BAE (ours)	**6120**	**3963**	0.324	**3971**	0.323
Llama 3.2-3B Layer 20 $d = 3072$	ReLU SAE	1923	1183	0.289	1289	0.312
	Top-k SAE	3234	2286	0.402	2508	0.425
	Gated ReLU SAE	4628	3271	**0.402**	3580	**0.437**
	TransCoder	5508	3001	0.233	3424	0.257
	BAE (ours)	**9308**	**5956**	0.308	**6805**	0.348

BAE は dense 特徴を削減できる． 学習済みの BAE および SAE における各特徴の活性化頻度分布を図 4 に可視化した．この可視化から，BAE の特徴は左寄りの分布を示しつつスパースに活性化している一方で，典型的な SAE では一部のチャネルが高い活性化頻度を保ち，long-tail 分布を示していることがわかる．これは dense な活性化の存在を示唆している [2, 3]．これらの結果は，ミニバッチ単位のエントロピー損失が入力インスタンス間で広範に活性化する特徴を抑制できるという我々の仮説を支持するものである．他の設定における結果については完全版論文を参照．

5　結論

本研究では，LLM の解釈性可能性研究のためのツールとして BAE を提案した．BAE はミニバッチ単位の二値化隠れ活性に対してエントロピー最小化を行い，スパースで原子的な特徴を抽出する．実験により，BAE が特徴集合のエントロピーを高精度に推定し，LLM の隠れ状態から明確な特徴を効果的に分離できることを示した．さらに，BAE は隠れ状態ベクトルの高効率な圧縮も可能であり，隠れ状態ベクトルの保存やストリーミングへの応用を促進する．詳細は本論文の完全版を参照されたい．

2)　典型的な SAE では，隠れ活性（本研究の h_1 に相当）がそのまま活性強度として利用できる．

3)　$\bar{h}_1$ は学習中に保存された付随成分，または H_1 全体の平均値として得られる．

謝辞

本研究は，JST 創発的研究支援事業 JPMJFR232K，および中島記念国際交流財団の助成を受けたものです．

参考文献

[1] Dong Shu, et al. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. **arXiv preprint arXiv:2503.05613**, 2025.

[2] Alessandro Stolfo, et al. Antipodal pairing and mechanistic signals in dense SAE latents. In **ICLR 2025 Workshop on Building Trust in Language Models and Applications**, 2025.

[3] Senthooran Rajamanoharan, et al. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. **arXiv preprint arXiv:2407.14435**, 2024.

[4] Xiaoqing Sun, et al. Dense sae latents are features, not bugs. **arXiv preprint arXiv:2506.15679**, 2025.

[5] Nelson Elhage, et al. Toy models of superposition. 2022.

[6] Kristjan Greenewald, et al. High-dimensional smoothed entropy estimation via dimensionality reduction. In **2023 IEEE International Symposium on Information Theory (ISIT)**, pages 2613–2618. IEEE, 2023.

[7] Itay Hubara, et al. Binarized neural networks. **Advances in neural information processing systems**, 29, 2016.

[8] Steven Bills, et al. Language models can explain neurons in language models, 2023.

[9] Robert Huben, et al. Sparse autoencoders find highly interpretable features in language models. In **The Twelfth International Conference on Learning Representations**, 2024.

[10] Ofir Press, et al. Measuring and narrowing the compositionality gap in language models. In **Findings of the Association for Computational Linguistics: EMNLP 2023**, pages 5687–5711, 2023.

[11] Jun Gao, et al. Representation degeneration problem in training natural language generation models. In **International Conference on Learning Representations**, 2019.

[12] Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, 2019.

[13] Daniel Biś, et al. Too much in common: Shifting of embeddings in transformer language models and its implications. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, June 2021.

[14] Nathan Godey, et al. Anisotropy is inherent to self-attention in transformers. In **Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)**, March 2024.

[15] Edwin Vargas, et al. Biper: Binary neural networks using a periodic function. In **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**, pages 5684–5693, 2024.

[16] Leo Gao, et al. The pile: An 800gb dataset of diverse text for language modeling. **arXiv preprint arXiv:2101.00027**, 2020.

[17] Aaron Grattafiori, et al. The llama 3 herd of models. **arXiv preprint arXiv:2407.21783**, 2024.

[18] Nelson Elhage, et al. A mathematical framework for transformer circuits. **Transformer Circuits Thread**, 2021.

[19] Cheng-Ping Hsieh, et al. RULER: What's the real context size of your long-context language models? In **First Conference on Language Modeling**, 2024.

[20] Nelson F. Liu, et al. Lost in the middle: How language models use long contexts. 12, 2024.

[21] Junqing He, et al. Never lost in the middle: Mastering long-context question answering with position-agnostic decompositional training. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, 2024.

[22] Shengnan An, et al. Make your llm fully utilize the context. **Advances in Neural Information Processing Systems**, 37:62160–62188, 2024.

[23] Jiaheng Liu, et al. A comprehensive survey on long context language modeling, 2025.

[24] Ganesh Jawahar, et al. What does BERT learn about the structure of language? In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, 2019.

[25] Yixiong Chen, et al. Which layer is learning faster? a systematic exploration of layer-wise convergence rate for deep neural networks. In **The Eleventh International Conference on Learning Representations**, 2023.

[26] Lean Wang, et al. Label words are anchors: An information flow perspective for understanding in-context learning. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pages 9840–9855, 2023.

[27] Da Xiao, et al. MUDDFormer: Breaking residual bottlenecks in transformers via multiway dynamic dense connections. In **Forty-second International Conference on Machine Learning**, 2025.

[28] Jiawei Gu, et al. A survey on llm-as-a-judge. **arXiv preprint arXiv:2411.15594**, 2024.

[29] Yukun Zhu, et al. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In **Proceedings of the IEEE international conference on computer vision**, pages 19–27, 2015.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

[31] Timo Schick, et al. Toolformer: Language models can teach themselves to use tools. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2023.

[32] Janice Ahn, et al. Large language models for mathematical reasoning: Progresses and challenges. In **Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop**, pages 225–237, 2024.

[33] Xin Xu, et al. Can LLMs solve longer math word problems better? In **The Thirteenth International Conference on Learning Representations**, 2025.

[34] Zihao Zhao, et al. Calibrate before use: Improving few-shot performance of language models. In **Proceedings of the 38th International Conference on Machine Learning**, Proceedings of Machine Learning Research, 2021.

[35] Jiahui Geng, et al. A survey of confidence estimation and calibration in large language models. In **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, 2024.

[36] Yu Fei, et al. Mitigating label biases for in-context learning. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, 2023.

A　実験の詳細

デフォルトハイパーパラメータ．デフォルトでは，$d' = 4d$，$\alpha_e = 10^{-7}$，$\alpha_c = 10^{-7}$ に設定する．最適化には Adam [30] を用い，学習率 5×10^{-4}，モーメンタム係数 $\alpha_1 = 0.9$，$\alpha_2 = 0.999$，ミニバッチサイズ $n_b = 512$ として，2000 エポック学習を行う．また，最初の 500 エポックでは $\alpha_e = 0$ とする．

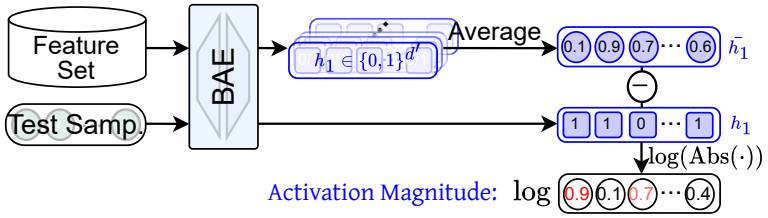

図 5　バースティネスに基づく活性化強度の計算プロセス（§4.1）．

人工ランダム方向ベンチマーク．BAE の二値化された隠れ活性（h_1）におけるエントロピー推定を評価するために，人工的なランダム方向ベンチマークを構築する．手順は次の通りである．**(1)** 次元 d の r 階直交基底 $M \in \mathbb{R}^{r \times d}$ をサンプリングする．**(2)** r 個の二値係数 $c \in \{0,1\}^r$ をサンプリングする．**(3)** c の中で値が 1 である要素に対応する M の基底を和として足し合わせ，インスタンス cM を生成する．(2) と (3) を n 回繰り返すことで，n 個のサンプルからなる人工ランダム方向データセットを得る．直感的には，このデータセットのエントロピーは r である．なぜなら，固定された基底のもとで，ランダム性は r 個の独立したベルヌーイ係数にのみ由来するからである．

§3.1 の BAE 訓練パラメータ．上記のプロセスに従い，全 65536 個のサンプルを生成し，このうち $n = 52428$ 個を学習用サンプル，残りを検証用サンプルとした．特に言及のないハイパーパラメータについては，すべてデフォルト値を用いた．学習後，二値符号化された h_1 を得るために，改めて全 65536 個のサンプルに対して BAE を適用し，式 4 に示したとおり，これらの h_1 に対する周辺エントロピーを計算した．

§3.2 の BAE 訓練パラメータ．Pile-train 分割から $n = 209{,}715$ 個のデータサンプルを生成し，さらに 52,429 個のデータサンプルを検証用として用いた．特に言及していないハイパーパラメータについては，すべてデフォルト設定を使用した．また，すべての位置における隠れ状態集合の長さを揃えるため，Llama 3.2-1B への入力文のうち，トークン長が 1024 未満の文をすべて除外した．学習後，先に用いた合計 262,144 個のサンプルすべてに対して再度 BAE を適用し，h_1 の二値符号化を行った上で，式 4 に従い，これらの h_1 に対する周辺エントロピーを算出した．

§4.2 の BAE・ベースライン訓練パラメータ．Pile-train 分割において，Llama 3.2-1B の特定の層から $n = 8{,}243{,}323$ 個の隠れ状態ベクトルをサンプリングし，そのうち 6,594,658 個を BAE/SAE の学習用サンプルとして使用し，残りを検証用とした．オートエンコーダは 200 エポックにわたり学習を行い，最初の 50 エポックにおいては $\alpha_e = 0$ と設定した．

B　ComSem: 共通意味に基づく特徴解釈と評価

Revisiting Current Automatic Feature Interpretation and Evaluation. 本節では，既存の特徴に対する自動解釈および評価手法を再検討する [8, 9]．ここでは，W_{out} における特徴（線形ベクトル）に対応する h_1 のチャネルを対象とする．**(Step 1)** ある入力テキストが与えられると，当該チャネルにおける活性化強度（例：SAE における h_1 の該当チャネル値）が各トークンごとに計算される．続いて **(Step 2)**，文と全トークンに対する活性化強度の組を，プロンプト（例：「以下の活性化に基づいて，この特徴の説明を予測せよ」）とともに LLM に入力し，当該特徴を単一のフレーズ（例：「自由に関連する語彙」）として解釈させる．**(Step 3)** テスト入力テキストが与えられると，生成された解釈に基づいて，各トークンにおける活性化強度のシミュレーションを LLM に問い合わせる．最後に，このシミュレーションされた活性化強度と，SAE により計算された活性化強度との相関係数を，当該特徴の解釈可能性スコアとして扱う（詳細は [8] を参照されたい）．

上記の手順における Step 2 は，多数の数値トークンを正確に取り扱い，信頼できる説明フレーズを生成する LLM の能力に依存している．また Step 3 では，数値トークンを通じて活性化強度を忠実にシミュレーションすることが求められ，これは LLM の数学的推論能力および出力キャリブレーションに対して高い要件を課すものである．しかしながら，既存研究により，LLM は言語的タスクと比較して数値推論における能力が相対的に弱いことが報告されている [10, 31, 32, 33]．さらに，出力が特定のトークンに対して暗黙的なバイアスを含む可能性も指摘されており [34, 35]，このような点から，上述のパイプラインは信頼性に欠ける側面を有し，その堅牢性および信憑性を向上させるための再検討が必要である．

Common Semantics-based Feature Interpretation and Evaluation (ComSem). したがって，LLM に数値トークンを直接処理させることへの依存を避けるため，抽出された特徴を解釈する際に，LLM が有する言語的意味認識能力の強みを活用する新たなパイプラインとして ComSem を提案する．具体的には，オートエンコーダの h_1 における特定のチャネル（特徴）を対象とし，テスト文集合が与えられたとき，**(Step 1)** 当該チャネルにおいて**強い活性化強度**を示す隠れ状態をもつすべてのトークン（BAE に関する詳細は §4.1 参照）を取り出し，それらを対応する文とともに収集する．**(Step 2)** これらのトークンが文脈中で示す共通性を，バックエンドの LLM に問い合わせ，その結果を当該チャネルに対応する特徴の解釈とする．**(Step 3)** 当該チャネルにおいて有意に活性化したトークン–文の組からなるホールドアウト集合に対し，Step 2 で生成された解釈によって各トークンが説明可能か否かを LLM に判定させる．「Yes」と判断された比率を，当該特徴の解釈可能性スコアとして算出する．

ComSem は，数学的・数値的推論に関する懸念を回避できるとともに，真偽値出力に対して単純な出力キャリブレーションを適用する可能性を提供する[4)]．以上の理由から，我々は BAE の評価に ComSem を用いる．

4)　例えば [34, 36] を参照されたい．本研究では，このような出力キャリブレーションは使用していない．