MLP の重みを反映した Sparse Autoencoder の初期化手法の提案

菊谷 幹 [1]　北田 俊輔 [1]　原 聡 [1]
[1] 電気通信大学
{k2210202@gl.cc., ka004763@gl.cc., satohara@}uec.ac.jp

概要

Sparse Autoencoder (SAE) は LLM 内部の処理を解釈するためのツールとして広く用いられている。しかし、異なる初期値で訓練した SAE では異なる “特徴” が学習されることが報告されている。この問題を解決して SAE の学習を安定化させるためには、SAE の良い初期値を設定する必要がある。本研究では SAE の初期値として Transformer 層の MLP のデコーダを用いる手法として SAE-MD を提案する。実験の結果、SAE-MD は既存の SAE と同等の性能を有し、学習の安定性を向上できることを確認した。

1　はじめに

大規模言語モデル (LLM) の発展に伴い、LLM の出力に対する安全性の追求やさらなる改良のため、モデル内部での情報の処理方法を明らかにする試みが数多くなされている [1, 2]。そのような試みの一つが Sparse Autoencoder (SAE) である [3, 4, 5]。SAE は、LLM 内部の情報を “特徴”（≈ 人間の解釈可能な概念）としてニューロン一つ一つに対応付けることを目的とした手法である。SAE のニューロンの発火の様子を見ることで、SAE が獲得した “特徴” を用いて LLM の内部の処理を解釈できるようになる。

しかし、異なる初期値で訓練した SAE では異なる “特徴” が学習されることが報告されている [6]。この事実は SAE がいくつかの “特徴” を捉えきれない、あるいは逆にノイズなどの無関係な “特徴” を捉えてしまっている可能性を示唆している。このような不安定な SAE を用いて LLM の挙動を解釈しても、その解釈結果が不正確である懸念が残る。SAE が LLM の主要な解釈手段の一つとなっている現在において、その不安定性の解消は喫緊の課題である。

SAE の初期値への強い依存性を解消し、“特徴” の学習を安定化させるためには、SAE の良い初期値を設定する必要がある。そこで、本研究では SAE の初期値として Transformer 層の MLP のデコーダを用いることを提案する。本研究の着眼点は、MLP のデコーダのベクトルの和として記述されている LLM の埋め込み表現を、SAE は改めて複数のベクトルの和として記述し直していると解釈する点にある。この事実は MLP のデコーダが SAE のデコーダとして有用であることを示唆している。実験の結果、MLP のデコーダを初期値に用いた SAE は既存の SAE と同等の性能を有しつつも、学習の安定性を大きく向上できることを確認した。

2　Sparse Autoencoder (SAE)

SAE は入力ベクトルをスパースなベクトルに変換し、そのベクトルを使って入力ベクトルを再構成するように学習するモデルである [3]。SAE に LLM 内部の埋め込み表現を入力することでスパースなベクトルに変換できる。このとき、スパースなベクトルの各要素が “特徴” に対応する。ベクトルの i 番目の要素が発火し非ゼロであれば、その要素に対応する “特徴” が埋め込み表現中に存在すると解釈できる。

様々な構造の SAE が提案されており、代表的な SAE の一つが活性化関数に JumpReLU を用いた JumpReLU SAE である [7]。$\boldsymbol{x} \in \mathbb{R}^d$ を SAE への入力ベクトルとしたとき、n 次元の中間層を持つ JumpReLU SAE は式 (1), (2), (3) で表される。

$$\boldsymbol{z} = \mathrm{JumpReLU}_\theta(\boldsymbol{W}^{\mathrm{enc}}(\boldsymbol{x} - \boldsymbol{b})), \tag{1}$$

$$\hat{\boldsymbol{x}} = \boldsymbol{W}^{\mathrm{dec}}\boldsymbol{z} + \boldsymbol{b} = \sum_i z_i \boldsymbol{w}_i^{\mathrm{dec}} + \boldsymbol{b}, \tag{2}$$

$$\mathrm{JumpReLU}_\theta(z) = z \ (\text{if } z > \theta), \quad 0 \ (\text{otherwise}). \tag{3}$$

$\boldsymbol{W}^{\mathrm{enc}} \in \mathbb{R}^{n \times d}$ と $\boldsymbol{W}^{\mathrm{dec}} \in \mathbb{R}^{d \times n}$ は SAE のエンコーダとデコーダの重み行列、$\boldsymbol{b} \in \mathbb{R}^d$ はバイアスである。また $\boldsymbol{w}_i^{\mathrm{dec}}$ は $\boldsymbol{W}^{\mathrm{dec}}$ の i 列目のベクトルである。

本研究では JumpReLU SAE を研究の対象とし、JumpReLU SAE を SAE と略記する。

3 提案手法: SAE-MD

本研究の目的は SAE の “特徴” の学習を安定化させるために、SAE の学習の良い初期値を設定することである。通常の SAE の学習では $\boldsymbol{W}^{\text{enc}}, \boldsymbol{W}^{\text{dec}}$ は一様分布で初期化される [8]。この初期化方法は勾配の消失・爆発への対策として有効であるが、SAE が獲得する “特徴” の安定化には寄与せず、異なる乱数シードで異なる “特徴” が学習される [9]。

本研究では SAE の学習に有効な初期値として Transformer 層の MLP のデコーダを用いることを提案する。以下では Transformer の内部構造の再解釈を通じてこのアイディアについて説明する。

Transformer 層の再解釈　LLM の第 l 層への入力を $\boldsymbol{x}^{(l)}$ とする。このとき、一般的な Pre-LayerNorm 型の Transformer 層の処理は

$$\boldsymbol{z}^{(l)} = \boldsymbol{x}^{(l)} + \mathrm{MHA}(\mathrm{LN}(\boldsymbol{x}^{(l)})), \tag{4}$$

$$\boldsymbol{x}^{(l+1)} = \boldsymbol{z}^{(l)} + \mathrm{FFN}(\mathrm{LN}(\boldsymbol{z}^{(l)})), \tag{5}$$

と記述される。ここで $\mathrm{MHA}, \mathrm{LN}, \mathrm{FFN}$ はそれぞれ Multi-Head の自己注意機構, レイヤー正規化、そして MLP を用いた非線形変換を表している。さらに式 (5) の第二項を書き下すと

$$\boldsymbol{U}^{\text{dec}}\sigma(\boldsymbol{U}^{\text{enc}}\mathrm{LN}(\boldsymbol{z}^{(l)}) + \boldsymbol{c}^{\text{enc}}) + \boldsymbol{c}^{\text{dec}} = \sum_i \sigma_i \boldsymbol{u}_i^{\text{dec}} + \boldsymbol{c}^{\text{dec}},$$

となる。ただし σ は MLP の活性化関数で、σ_i は $\sigma(\boldsymbol{U}^{\text{enc}}\mathrm{LN}(\boldsymbol{z}^{(l)}) + \boldsymbol{c}^{\text{enc}})$ の第 i 成分である。また、$\boldsymbol{U}^*, \boldsymbol{c}^*$ は MLP のエンコーダおよびデコーダの重み行列とバイアスで、$\boldsymbol{u}_i^{\text{dec}}$ はデコーダの重み行列 $\boldsymbol{U}^{\text{dec}}$ の i 列目のベクトルである。式 (5) において第二項が支配的だと仮定できるとする[1] と

$$\boldsymbol{x}^{(l+1)} \approx \sum_i \sigma_i \boldsymbol{u}_i^{\text{dec}} + \boldsymbol{c}^{\text{dec}}, \tag{6}$$

つまり LLM 内部の埋め込み表現 $\boldsymbol{x}^{(l+1)}$ は MLP のデコーダの重みのベクトル $\boldsymbol{u}_i^{\text{dec}}$ の和で記述できる。

本研究のアイディア　$\boldsymbol{x}^{(l+1)}$ を SAE により分解・再構成すること、つまり式 (1), (2) において $\boldsymbol{x} = \boldsymbol{x}^{(l+1)}$ とすることを考える。式 (6) と式 (2) を見比べると、下記の通り両者ともに $\boldsymbol{x}^{(l+1)}$ を異なる方法でベクトルの和として表現していると解釈できる。

$$\boldsymbol{x}^{(l+1)} \approx \underbrace{\sum_i \sigma_i \boldsymbol{u}_i^{\text{dec}} + \boldsymbol{c}^{\text{dec}}}_{\text{MLP 近似}} \approx \underbrace{\sum_i z_i \boldsymbol{w}_i^{\text{dec}} + \boldsymbol{b}}_{\text{SAE 近似}}. \tag{7}$$

1)　式 (6) および以降では式 (5) の第一項 $\boldsymbol{z}^{(l)}$ を無視している。この項の適切な取り扱いは今後の課題である。

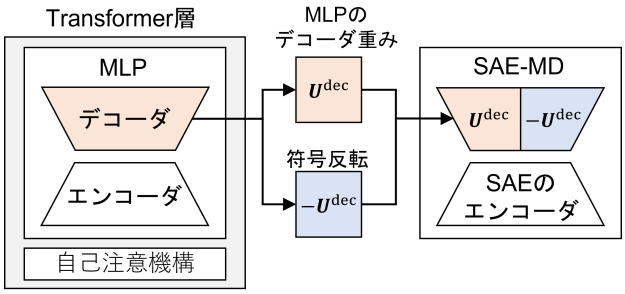


図 1: 提案手法 SAE-MD の概略図。LLM 内部の対応する Transformer 層の MLP のデコーダを SAE のデコーダの初期値として採用する。

この事実は SAE のデコーダのベクトル $\boldsymbol{w}_i^{\text{dec}}$ として、MLP のデコーダのベクトル $\boldsymbol{u}_i^{\text{dec}}$ が有効である可能性を示唆している。もちろん SAE は $\boldsymbol{z}$ にスパース性を要求するため、$\boldsymbol{u}_i^{\text{dec}}$ をそのまま SAE のデコーダに用いることが有効であるとは言い難い。しかし、$\boldsymbol{u}_i^{\text{dec}}$ を $\boldsymbol{w}_i^{\text{dec}}$ の初期値として用いることは有効であると考えられる。

提案手法：SAE-MD　上記のアイディアに基づいて、本研究では MLP のデコーダを SAE のデコーダの初期値に用いることを提案する。提案手法の概要を図 1 に示す。SAE では JumpReLU によって各 z_i が非負であるのに対して、一般の MLP の活性化の値 σ_i は負の値も取りえる。この場合、式 (7) の近似を成り立たせるには $\sigma_i \boldsymbol{u}_i^{\text{dec}} = |\sigma_i| \times \mathrm{sign}(\sigma_i)\, \boldsymbol{u}_i^{\text{dec}} \approx z_i \boldsymbol{w}_i^{\text{dec}}$ が必要であり、つまり $\boldsymbol{u}_i^{\text{dec}}$ だけでなく $-\boldsymbol{u}_i^{\text{dec}}$ も $\boldsymbol{w}_i^{\text{dec}}$ の初期値の候補とする必要がある。そこで提案手法では $\boldsymbol{U}^{\text{dec}}$ とその符号を反転させた $-\boldsymbol{U}^{\text{dec}}$ を結合し、$[\boldsymbol{U}^{\text{dec}}, -\boldsymbol{U}^{\text{dec}}] \in \mathbb{R}^{d \times 2m}$ を SAE のデコーダ $\boldsymbol{W}^{\text{dec}}$ の初期値として用いる。ただし m は MLP の中間層の次元である。このようにデコーダを鏡状に構成することから、提案手法を Mirror Decoder (MD) を用いた SAE として SAE-MD と呼称する。

4 実験

提案手法 SAE-MD が (i) SAE の学習の安定化を実現すること、(ii) SAE としての品質が通常の SAE と同等であること、についての実験結果を報告する。

4.1 SAE の学習手順

LLM として Gemma-2-2B[2] を採用し、第 12 層の埋め込み表現を使い通常の SAE および提案手法 SAE-MD による学習を行った。Gemma-2-2B の MLP

2)　hf.co/google/gemma-2-2b

の中間層の次元は 9,216 のため、本実験の SAE および SAE-MD の中間層の次元は 18,432 とした。SAE の学習には `dictionary_learning` を用いた[3]。SAE の学習データセットには Pile-uncopyrighted[4] を使用した。ハイパーパラメータは付録 A に示す。

実験では SAE, SAE-MD の両者について中間層のスパース性を調整する損失関数を用いて異なるスパース性を有する複数の SAE（非ゼロ要素数 20, 40, 80, 160, 320）を学習した。また、それぞれの学習時の乱数シードは 24 および 42 に固定した二通りを実施した。これにより SAE の初期化および学習データのシャッフルが、SAE および SAE-MD の学習に与える影響を調査した。

4.2 学習の安定性の評価

異なる乱数シード 24 および 42 で学習された SAE（SAE_{24}、SAE_{42}）および SAE-MD（$\mathrm{SAE\text{-}MD}_{24}$、$\mathrm{SAE\text{-}MD}_{42}$）の間で学習された “特徴” がどの程度異なるのかを評価した。

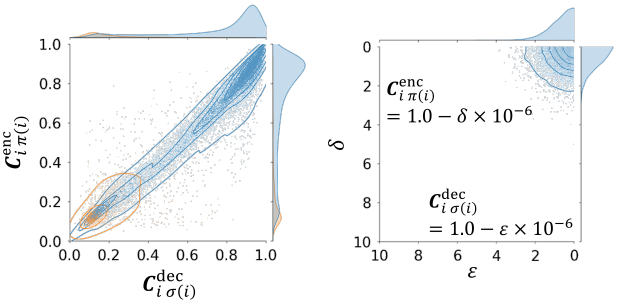
安定性の評価は Paulo らの手法 [6] に従って、ハンガリー法を用いた “特徴” のマッチングを行った。以下では SAE_{24}, SAE_{42} を例に説明する。$\mathrm{SAE\text{-}MD}_{24}$, $\mathrm{SAE\text{-}MD}_{42}$ についても同様の処理を適用する。

1. 類似度行列の作成: SAE_{24}, SAE_{42} についてコサイン類似度 $\boldsymbol{C}^{\mathrm{enc}}_{ij} = \cos(\boldsymbol{w}^{\mathrm{enc}}_{24,i}, \boldsymbol{w}^{\mathrm{enc}}_{42,j})$ および $\boldsymbol{C}^{\mathrm{dec}}_{ij} = \cos(\boldsymbol{w}^{\mathrm{dec}}_{24,i}, \boldsymbol{w}^{\mathrm{dec}}_{42,j})$ を計算する。$\boldsymbol{w}^{\mathrm{enc}}_{a,i}, \boldsymbol{w}^{\mathrm{dec}}_{a,i}$ は SAE_a の $\boldsymbol{W}^{\mathrm{enc}}$ の i 行目のベクトルおよび $\boldsymbol{W}^{\mathrm{dec}}$ の i 列目のベクトルである。

2. ハンガリー法の適用: 類似度行列 $\boldsymbol{C}^{\mathrm{enc}}, \boldsymbol{C}^{\mathrm{dec}}$ のそれぞれに対して類似度を最大化するようにハンガリー法を使ってマッチングを行う。第 i 行にマッチングした列のインデックスをそれぞれ $\pi(i), \sigma(i)$ とする。

上記の手順で “特徴” の比較を行った結果を図 2 に示す。図では SAE および SAE-MD のそれぞれで、各 i について $\boldsymbol{C}^{\mathrm{enc}}_{i\pi(i)}$ を縦軸、$\boldsymbol{C}^{\mathrm{dec}}_{i\sigma(i)}$ を横軸にプロットし、ハンガリー法によりエンコーダとデコーダの両者で同じマッチングがされた場合（$\pi(i) = \sigma(i)$）に青、そうでない場合（$\pi(i) \neq \sigma(i)$）にオレンジで表記している。ハンガリー法で同じマッチングがされ、さらにコサイン類似度 $\boldsymbol{C}^{\mathrm{enc}}_{i\pi(i)}, \boldsymbol{C}^{\mathrm{dec}}_{i\sigma(i)}$ が 1.0 に近いときに、二つの異なる SAE で “同じ特徴” が学習されたと判定できる。図では、通常の SAE ではコサイン類似度が 0.6 以上のある程度類似した “特徴” が

3） github.com/saprmarks/dictionary_learning
4） hf.co/datasets/monology/pile-uncopyrighted



(a) SAE　　(b) SAE-MD

図 2: 異なる乱数シードで学習された SAE の特徴の一致度合い（中間層の非ゼロ要素数 320 の場合）。縦軸が $\boldsymbol{C}^{\mathrm{enc}}_{i\pi(i)}$、横軸が $\boldsymbol{C}^{\mathrm{dec}}_{i\sigma(i)}$ で、ハンガリー法によりエンコーダとデコーダの両者で同じマッチングがされた場合（$\pi(i) = \sigma(i)$）に青、そうでない場合（$\pi(i) \neq \sigma(i)$）にオレンジで表記している。

多く見られつつも、0.2 前後の小さな類似度が見られる “特徴” も一定数存在する。この結果は先行研究 [6] の報告とも整合的であり、SAE が乱数シードによって異なる “特徴” を無視できない割合学習していることを示している。他方、提案手法 SAE-MD ではコサイン類似度はほぼ 1.0 であり乱数シードが学習される “特徴” に影響を及ぼしていないこと、つまり SAE の学習の安定化に強く寄与していることが確認できる。

4.3 SAE としての品質の評価

前節では SAE-MD が学習の安定化に強く寄与することを確認した。本節では SAE-MD が従来の SAE と比較してその品質も同等であることを確認する。SAE の品質の包括的なベンチマークとして SAEBench を使用した [10]。SAEBench は、SAE の主要な能力である**再構成**、**解釈性**、**概念検出**、**因果的介入・分離**について 8 つの指標で評価を行う。下記に各指標の概要を示す。なお、どの指標ともに「大きい方が良い結果を示す」ように設計されている。

再構成

Loss Recovered: LLM の損失が、SAE による再構成後にどの程度復元されるかの割合。

解釈性

Automated Interpretability: SAE の各 “特徴” z_i の説明に基づき、未知のデータでの z_i の発火をどの程度正確に予測できるか。

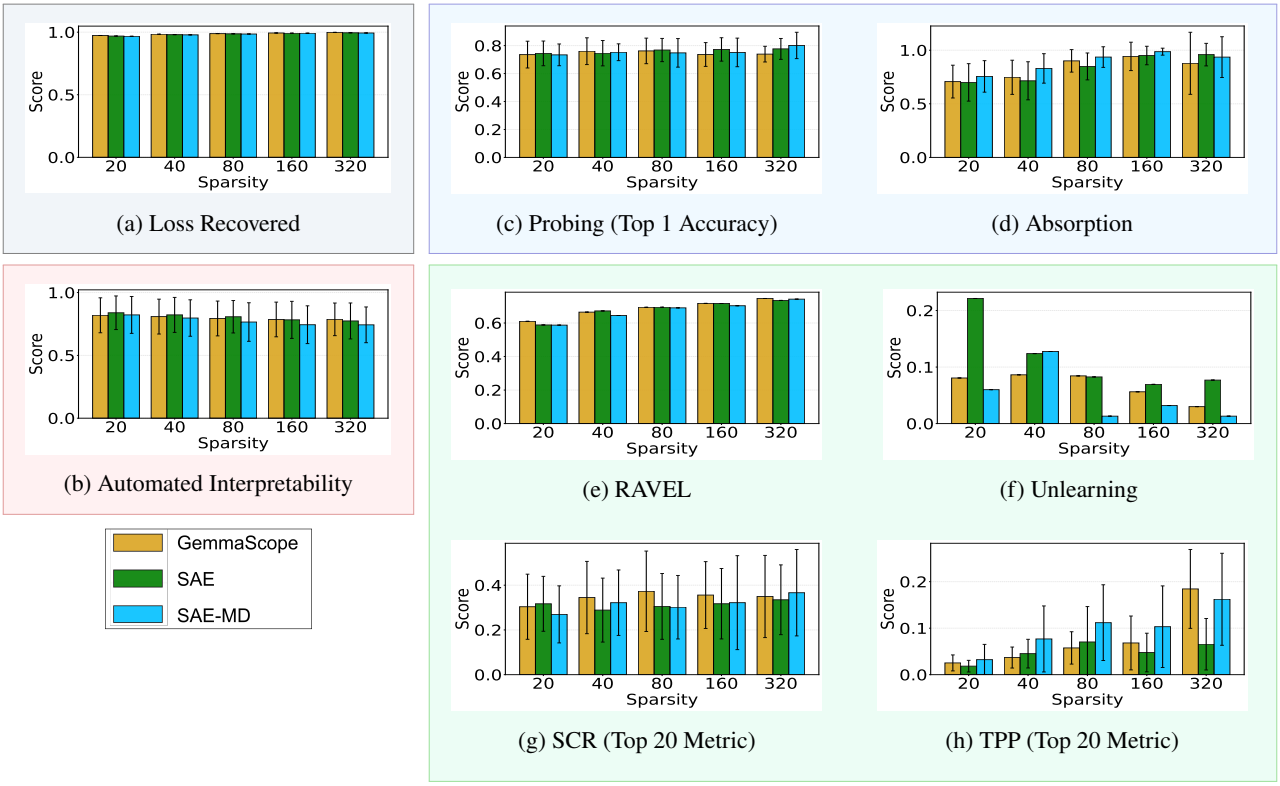


図 3: SAEBench の結果（平均と標準偏差）

概念検出

k-Sparse Probing: 上位 k 個の z_i のみを用いた線形分類器による、特定概念の識別精度。

Absorption: 下位概念が上位概念の特徴量に「吸収」され、概念の識別が困難になっている度合い。

因果的介入・分離

RAVEL: 属性間の相関を排除し、特定の属性のみを独立して変更できる操作の正確性。

Unlearning: 標的とする知識の消去能力と、無関係なタスクにおける性能維持のバランス。

Spurious Correlation Removal (SCR): 介入によって「偽の相関」を除去し因果関係を抽出する能力。

Targeted Probe Perturbation (TPP): 多クラス設定において、特定クラスの特徴量への介入が他クラスの予測に与える影響（因果的寄与度）。

図 3 に SAEBench による評価の結果を示す。本実験で用いた SAE および SAE-MD の性能を客観的に評価するため、Gemma Scope[5] [11] を比較対象として含めた。なお、乱数シードによる性能差はほぼ見られなかったため、ここでは乱数シードが 42 の結果を報告する。乱数シードの違いによる結果の差異は付録 B に示す。Loss Recovered と Automated Interpretability では SAE と SAE-MD がほぼ同等の値となっており、再構成と解釈性の面では両者は同等と言える。同様に、概念検出を測る指標 (k-Sparse Probing、Absorption) や、特徴の解離を測る指標 (RAVEL、SCR) についても、平均に小さな差はあれど標準偏差を考慮すると SAE と SAE-MD は同等の性能と言える。なお Unlearning と TPP は例外的にどの手法ともに 0.1 前後とスコアが低く、SAEBench [10] でのオリジナルの評価と整合的である。これらの結果は、提案手法 SAE-MD が安定性を大きく向上した上で、SAE の品質としては従来のもとの比較して遜色ないことを示している。

5 おわりに

本研究では SAE の学習を安定化させる方法として、SAE の初期値に Transformer 層の MLP のデコーダを用いる手法 SAE-MD を提案した。実験の結果、SAE-MD は既存の SAE と同等の性能を有し、学習の安定性を向上できることを確認した。より詳細な実験評価および式 (6) で無視した式 (5) の第一項 $z^{(l)}$ の適切な取り扱いが今後の課題である。

5） hf.co/google/gemma-scope-2b-pt-res

謝辞

本研究は、JST 経済安全保障重要技術育成プログラム JPMJKP24C3 の支援を受けたものです。

参考文献

[1] Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety – a review. **arXiv preprint arXiv:2404.14082**, 2024.

[2] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. **ACM Transactions on Intelligent Systems and Technology**, Vol. 15, No. 2, 2024.

[3] Andrew Y. Ng. Sparse autoencoder. **CS294A Lecture Notes, Stanford University**, pp. 251–258, 2011.

[4] Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In **International Conference on Learning Representations**, 2024.

[5] Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In **International Conference on Representation Learning**, 2025.

[6] Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. **arXiv preprint arXiv:2501.16615**, 2025.

[7] Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. **Google DeepMind**, 2024.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015.

[9] Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In **International Conference on Representation Learning**, 2025.

[10] Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, et al. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In **International Conference on Machine Learning**, 2025.

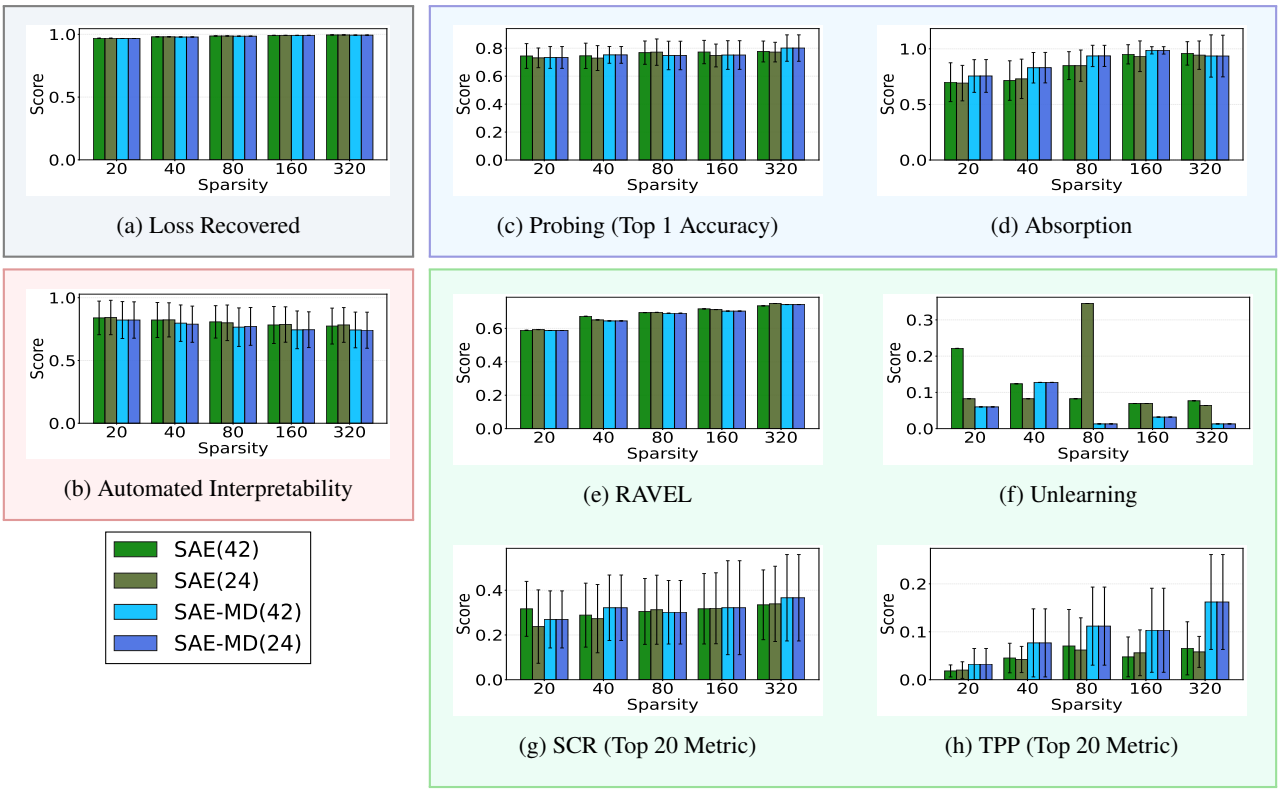
[11] Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. **arXiv preprint arXiv:2408.05147**, 2024.

A　SAE 学習のハイパーパラメータ

表 1: これらの ハイパーパラメータ は Karvonen ら [10] の設定を参考としている。

Hyperparameter	Value
Tokens processed	500M
Learning rate	3×10^{-4}
Learning rate warmup (from 0)	1,000 steps
Sparsity penalty warmup (from 0)	5,000 steps
Learning rate decay (to 0)	Last 20% of training
Dataset	Pile-uncopyrighted
Batch size	2.048
LLM context length	1,024

B　乱数シードによる SAEBench の結果の差異



(a) Loss Recovered　(b) Automated Interpretability　(c) Probing (Top 1 Accuracy)　(d) Absorption　(e) RAVEL　(f) Unlearning　(g) SCR (Top 20 Metric)　(h) TPP (Top 20 Metric)

図 4: SAEBench の結果（平均と標準偏差）。括弧内の数字は乱数シードの値を示す。