On the Computational Complexity of Inverting Generative Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

The objective of generative model inversion is to identify a size-n latent vector that produces a generative model output that closely matches a given target. This operation is a core computational primitive in numerous modern applications involving computer vision and NLP. However, the problem is known to be computationally challenging and NP-hard in the worst case. This paper aims to provide a fine-grained view of the landscape of computational hardness for this problem. We establish several new hardness lower bounds for both exact and approximate model inversion. In exact inversion, the goal is to determine whether a target is contained within the range of a given generative model. Under the strong exponential time hypothesis (SETH), we demonstrate that the computational complexity of exact inversion is lower bounded by $\Omega(2^n)$ via a reduction from k-SAT; this is a strengthening of known results. For the more practically relevant problem of approximate inversion, the goal is to determine whether a point in the model range is close to a given target with respect to the ℓ_p-norm. When p is a positive odd integer, under SETH, we provide an $\Omega(2^n)$ complexity lower bound via a reduction from the closest vectors problem (CVP). Finally, when p is even, under the exponential time hypothesis (ETH), we provide a lower bound of $2^{\Omega(n)}$ via a reduction from Half-Clique and Vertex-Cover.

1 Introduction

1.1 Generative model Inversion

In the last 30 years, recovery of latent vectors generating a target has gained attention. The focus has shifted away from "linear" generative models such as sparse models (1; 2; 3) and towards nonlinear generative models such as convolutional neural networks (4; 5), pre-trained generative priors (6), or untrained deep image priors (7; 8; 9). While there has been significant practical progress in compressed sensing with generative models, theoretical progress has been more modest. The seminal work of (6) established the first statistical upper bounds for this field, and (10) showed these bounds are nearly optimal, but only restrictive cases have provable algorithmic upper bounds for generative inversion. The paper (11) proves the convergence of projected gradient descent for compressed sensing with generative priors, but only under the assumption that the range of the generative model admits a polynomial-time projection oracle.

Several works establish upper bounds. The paper (12) proves the convergence of gradient descent for shallow generative priors whose weights obey a distributional assumption. (13) shows the correctness of a layer-wise inversion algorithm for sufficiently expansive networks, and establishes NP-hardness lower bounds for exact calculation. (14) and (15) show that under certain structural assumptions on G, some methods converge to a neighborhood of true solution. However, these assumptions are somewhat hard to verify in practice. Recent work has proposed using invertible generative models for image sampling (16; 17), using non-volume preserving transformations and being compared to a different approach proposed by (18). Our focus is on more general families of generative neural networks with ReLU activations, which are not necessarily invertible.

Problem Statement. A 1-layer ReLU network $G_1 : \mathbb{R}^n \to \mathbb{R}^{m_1}$ can be defined as $G_1(z) = \text{ReLU}(W_1 z + b_1)$ with weight matrix $W_1 \in \mathbb{R}^{m_1 \times n}$ and bias $b_1 \in \mathbb{R}^{m_1}$, where m_1 is the number of hidden neurons. An L-layer neural network $G := G_L$ can be expressed with the following recurrence relation: $G_l(z) = \text{ReLU}(W_l G_{l-1}(z) + b_l)$ for $l \in \mathbb{Z}^+,\ l = 2, 3, \dots, L$, where $W_l \in \mathbb{R}^{m_l \times m_{l-1}}$ are the layer-wise weight matrices, and $b_l \in \mathbb{R}^{m_l}$ are layer-wise bias vectors. We assume that the network width, $\max_l m_l$, is bounded as $O(n)$ unless otherwise specified.

In the generative inversion problem, we are given a ReLU network G and an observation x. Then, the purpose is to determine the closest point of the range of the neural network to the input x. In other words, we want to find z^* that satisfies the following under a given norm: $z^* = \arg\min_z \|G(z) - x\|$.

Stated as a decision problem, the goal of exact recovery is to distinguish between two cases: either there exists a z^* such that $G(z^*) = x$, or for all z we have $G(z) \neq x$. On the other hand, in many practical cases finding a close enough point suffices. Therefore, we investigate the hardness of the decision problem where the goal is to distinguish between the two cases for a parameter $\delta > 0$: there exists a point z^* such that $\|G(z^*) - x\| < \delta$, or for all z we have $\|G(z) - x\| \geq \delta$. We prove several new hardness results when $\|\cdot\|$ corresponds to ℓ_p-norm depending on the parity of p.

1.2 Fine-grained complexity.

Classical complexity theory has attempted to delineate the boundary between problems that admit efficient (polynomial-time) algorithms and problems that do not. A fine(r) grained picture of the landscape of polynomial-time has begun to emerge over the last decade. In particular, the focus has shifted towards pinning down the exponent, c, of a problem that can be solved in polynomial time $\tilde{O}(n^c)$. Most of these newer results are conditional and rely on reductions from popular (but plausible) conjectures such as the Strong Exponential Time Hypothesis (SETH) (19), (20). See the relevant surveys (21), (22), (23), and (24) for comprehensive overviews of this emerging area. This approach provides conditional lower bounds on well-known problems such as edit distance (25), Frechet distance (26), dynamic time warping (27), longest common subsequence (LCS) (28), and string matching (29). In the context of machine learning, reductions from SETH applied to clustering (30), kernel PCA (31), sparse linear regression (32), Gaussian kernel density estimation (33), and approximate nearest neighbors (34) problems. In recent work, this approach has also been shown to imply an $\Omega(n^2)$-lower bound for transformer models with input size n (35).

(Strong) Exponential Time Hypothesis and k-SAT. The k-SAT problem involves a given SAT formula on n variables, with each clause of size k, and requires us to determine whether the formula is satisfiable or not. Despite decades of effort, no one has invented a faster-than-exponential ($O(2^n)$) time algorithm for this problem. Unless $P = NP$, no polynomial-time algorithm exists. The Strong Exponential Time Hypothesis (SETH) is a strengthening of this statement (19): for every $\varepsilon > 0$, there is no algorithm that solves k-SAT in $2^{(1-\varepsilon)n}$ time. The Exponential Time Hypothesis (ETH) is a (slightly) weaker conjecture: there exists $\delta > 0$ such that 3-SAT cannot be solved in time $2^{\delta n}$.

Closest Vector Problem. We leverage SETH-hardness of the closest vector problem (CVP), defined as follows: Consider a lattice $\mathcal{L} = \mathcal{L}(\mathbf{B}) = \{\mathbf{B}z \mid z \in \mathbb{Z}^n\}$, where $B = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n)$ and all the vectors $\mathbf{v}_i \in \mathbb{R}^d$. Given a target vector $\mathbf{t}$, define $dist_p(\mathcal{L}, \mathbf{t}) = \min_{\mathbf{x} \in \mathcal{L}(\mathbf{B})} \|\mathbf{x} - \mathbf{t}\|_p$. The goal of the CVP$_p$ problem is to distinguish between two cases: YES ($dist_p(\mathcal{L}, \mathbf{t}) \leq r$) and NO ($dist_p(\mathcal{L}, \mathbf{t}) > r$). A result of (36) states that assuming SETH, then CVP$_p$ cannot be solved in $O(2^{(1-\epsilon)n})$ time for odd integers $p \geq 1, p \neq 2\mathbb{Z}$. As follow-up work, (37) proved the same hardness for the so-called (0,1)-CVP$_p$ problem: for any $1 \leq p \leq \infty$, distinguish between the two cases: YES ($\|By^* - t\| \leq r$ for some $y^* \in \{0, 1\}^n$) and NO ($\|By - t\| > r$ for all $y \in \mathbb{Z}$).

Half-Clique and Vertex Cover. We also consider two graph optimization problems: Vertex Cover and Half-Clique. The goal of Vertex Cover is to find a set of vertices of the minimum size that touch every edge in a given graph; the decision version is to decide whether or not a vertex cover of a given size exists. The Half-Clique problem is a special case of the k-CLIQUE problem (for $k = n/2$), where the goal is to decide whether or not there exists a clique with k-vertices. Both decision problems are classical, NP-complete, and ETH-hard (22).

2 Hardness Results

The constructions of the reductions from the problems whose hardness is known under some hypotheses are given in this part. For the sake of brevity, we leave the analysis to the full version.

2.1 Hardness of Exact Inversion

The work (13) shows the NP-hardness of the exact inversion problem. We strengthen this to show SETH-hardness below in 1. First, we provide hardness results when $z \in \{1, -1\}^n$, and we generalize these results for $z \in \mathbb{R}^n$.

Theorem 1. *Suppose SETH holds. Then for any $\epsilon > 0$ there is a 2-layer, $O(n)$-width ReLU network, $G_2(z)$ and a target x for which no $O(2^{(1-\epsilon)n})$ time algorithm can demonstrate the existence of $z \in \{1, -1\}^n$ that satisfies $G_2(z) = x$.*

*Reduction.*The reduction is from k-SAT, with m-clauses, n-literals. Let $z = [z_1, z_2, \dots, z_n]^T \in \{1, -1\}^n$ where z_i represents the assignment of i^{th} literal, 1 for TRUE and -1 for FALSE. We construct the first-layer weight matrix $W_1 \in \mathbb{R}^{m \times n}$ as, if i^{th} literal appears positively or negatively (or does not appear) in j^{th} clause then $(W_1)_{ji} = -1$ or 1(or 0). Also, the first layer bias vector $b_1 \in \mathbb{R}^m$ is constructed that all entries are $-(k-1)$. For the second layer, $W_2 \in \mathbb{R}^{1 \times m}$ is all 1 matrix, while $b_2 = 0$. Lastly, the given observation $x \in \mathbb{R}$ is 0. It can be shown that determining whether there is a z for which $G_2(z) = x$ can be reduced from k-SAT by this construction.

Theorem 2. *Suppose SETH holds. Then for any $\epsilon > 0$, there is a 4-layer, $O(n)$-width ReLU network G_4 and an observation x such that there is no $O(2^{(1-\epsilon)n})$ time algorithm to determine whether there exists $z \in \mathbb{R}^n$ satisfying $G_4(z) = x$.*

*Reduction.*The reduction from k-SAT. The input is $z = [z_1, z_2, \dots, z_n]^T \in \mathbb{R}^n$. The first 2 layers are used to map all the values into $[-1, 1]$. The first layer output is $\max\{z_i, -1\}$ for all $i \in [n]$, and the second layer output is $v \in \mathbb{R}^n$ such that $v_i = \min\{\max\{z_i, -1\}, 1\}$ for all $i \in [n]$. The third layer output is $u \in R^{m+2}$ where first m nodes will be defined as the first layer of Theorem 1 with same weight and bias, and we will add 2 more nodes $u_{m+1} = \sum_{i=1}^{n} \max\{v_i, 0\}$ and $u_{m+2} = \sum_{i=1}^{n} - \min\{v_i, 0\}$. The last layer has 2 output nodes, the first output node $G_4(z)_1 = \sum_{i=1}^{n} u_i$, and the second output node $G_4(z)_2 = u_{m+1} + u_{m+2}$. Also, the given observation $x = [0, n]^T \in \mathbb{R}^2$.

2.2 Hardness of Approximate Inversion

We first lower bound the complexity of the inverse generative model under the ℓ_p-norm for positive odd numbers p by reducing it from CVP$_p$. This gives a lower bound of $O(2^n)$ by assuming SETH. For positive even numbers p, we reduce from the Half-Clique and Vertex Cover Problems. This gives a lower bound of $2^{\Omega(n)}$ by assuming ETH.

2.2.1 Reduction from CVP

In Theorem 3, we present hardness results on the input $z \in \{0, 1\}^n$, which are derived by applying a reduction from $(0, 1)$-CVP$_p$. Our results are inspired by the proofs of the hardness of the sparse linear regression problem established in (32). We further extend our findings by proving that the results also hold for $z \in \mathbb{R}^n$ in Theorem 4, with a reduction from the binary case.

Theorem 3. *Assume SETH. Then for any $\epsilon > 0$, there is a 1-layer, $O(n)$-width ReLU network G_1 and an observation x such that there is no $O(2^{(1-\epsilon)n})$ time algorithm to determine whether there exists $z \in \{0, 1\}^n$ that satisfies $\|G_1(z) - x\|_p < \delta$ for a given $\delta > 0$ and any positive odd number p.*

Reduction. The reduction is from $(0, 1)$-CVP$_p$. The first layer weight and bias are constructed as $W_1 = \begin{bmatrix} \overline{W}_1 \\ -\overline{W}_1 \end{bmatrix}$ and $b_1 = \begin{bmatrix} \bar{b}_1 \\ -\bar{b}_1 \end{bmatrix}$ where for some positive real number $\alpha > \delta = r$,

$$\overline{W}_1 = \begin{bmatrix} \mathbf{v}_1 & \vec{0} & \mathbf{v}_2 & \vec{0} & \dots & \mathbf{v}_n & \vec{0} \\ \alpha & \alpha & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \alpha & \alpha & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & \alpha & \alpha \end{bmatrix}, \ \bar{b}_1 = \begin{bmatrix} -\mathbf{t} \\ -\alpha \\ -\alpha \\ -\alpha \end{bmatrix}$$

Also, the given observation x is the all-zeros vector.

Theorem 4. *Assume SETH. Then for any $\epsilon > 0$, there is a 5-layer ReLU network G_5 and an observation x such that there is no $O(2^{(1-\epsilon)n})$ time algorithm to determine whether there exists $z \in \mathbb{R}^n$ that satisfies $\|G_5(z) - x\|_p < \delta$ for a given $\delta > 0$ and any positive odd number p.*

In the case of exact inversion, the transition from $\{0, 1\}^n$ to $\mathbb{R}^n$ was accomplished by adding two extra layers. When we try the same trick in the approximate case, instead of having exact binary values $\{0, 1\}$, we result in points clustered in a small interval around 0 or around 1. To overcome this issue, we introduce two additional layers that map the points around 0 to 0 and those around 1 to 1. The following construction is for $\delta < 1/4$. It can be done with slight changes for any $\delta > 0$.

Reduction. The first 2 layers are designed to make the entries z in $[0, 1]$, so z_i is mapped to $v_i = \min\{\max\{z_i, 0\}, 1\}$ for all $i \in [n]$. In the 3^{rd} layer, the first n nodes are defined by $u_{[n]} = \text{ReLU}(W_3 v + b_3)$ where $W_3 = -I_n$ and $b_3 = \frac{1}{2} \cdot \vec{1} \in \mathbb{R}^n$. Also, introduce 2 more nodes that $u_{n+1} = \sum_{i=1}^{n} \max\{v_i, 1/2\}$ and $u_{n+2} = \sum_{i=1}^{n} - \min\{v_i, 1/2\}$. In the 4^{th} layer, the first n nodes are defined by $t_{[n]} = \text{ReLU}(W_4 u_{[n]} + b_4)$ where $W_4 = -4 \cdot I_n$, and $b_4 = \vec{1} \in \mathbb{R}^n$. Also, we add another node $t_{n+1} = u_{n+1} + u_{n+2} = \sum_{i=1}^{n} \max\{v_i, 1/2\} + \sum_{i=1}^{n} - \min\{v_i, 1/2\}$. In the last layer, construction is similar to Theorem 3 with an addition of one more node given by $s_{m+1} = t_{n+1}$ so that the output is $s \in \mathbb{R}^{m+1}$. And, the given output x is $x = [\vec{0}, n/2] \in \mathbb{R}^{m+1}$.

2.2.2 Reduction from Half-Clique

To handle the case of even p, we cannot reduce from CVP in (37). Instead, we achieve this through two reductions: to the binary case from the Half-Clique and from the Vertex Cover problem. To transition from binary to real numbers, the same approach used in Theorem 4 can be followed. Both of these approaches enable us to show ETH hardness (22). We demonstrate a $2^{\Omega(n)}$ lower bound for binary inputs via a reduction from Half-Clique and Vertex Cover.

Theorem 5. *Assume ETH. There is a 1-layer, $O(n^2)$-width ReLU network G_1 and an observation x such that computational complexity to determine whether there exists a $z \in \{0, 1\}^n$ with $\|G_1(z) - x\|_p \leq \delta$ is $2^{\Omega(n)}$ for a given $\delta > 0$ and any positive even number p.*

Reduction. The reduction is from the Half-Clique problem on a positive edge-weighted graph $G(V, E)$ where $|V| = n$. The Half-Clique problem is to determine if a half-clique with a total weight less than M exists. We use the same trick $G_1(z) = \text{ReLU}(W_1 z + b_1)$ where $W_1 = \begin{bmatrix} W \\ -W \end{bmatrix}$ and $b_1 = \begin{bmatrix} b \\ -b \end{bmatrix}$. And, we consider the problem definition to be $\|Wz + b\|_p \leq \delta$ when the target is $x = \vec{0}$. Firstly, label the vertices by $1, \ldots, n$. Construct the matrix $C \in \mathbb{R}^{\binom{n}{2} \times n}$ and the vector $c \in \mathbb{R}^{\binom{n}{2}}$ as follows. For an edge $e(i, j) \in E$ with edge weight w_e, $C_{ek} = 2\sqrt[p]{w_e}$ if $k \in \{i, j\}$, $C_{ek} = 0$ otherwise, and $c_e = -\sqrt[p]{w_e}$. For a non-edge $e(i, j) \notin E$, $C_{ek} = 2\alpha$ if $k \in \{i, j\}$, $C_{ek} = 0$ otherwise, and $c_e = -\alpha$, here α is a large constant. The matrix $W \in \mathbb{R}^{(\binom{n}{2}+1) \times n}$ is constructed as the concatenation of C and $\beta \cdot \vec{1} \in \mathbb{R}^{1 \times n}$, and vector $b \in \mathbb{R}^{n+1}$ is defined by $b_{[n]} = c$ and $b_{n+1} = -(n/2)\beta$, here β is a large constant. Let Z be the number of non-edges, and the given value $\delta = \sqrt[p]{\sum_{e \in E} w_e + \alpha^p Z + (3^p - 1)M}$.

2.2.3 Reduction from Vertex Cover

Here, we give the following construction for Theorem 5 by reduction from Vertex Cover.

Reduction. The Vertex Cover problem asks if $G(V, E)$ has a vertex cover of size q. Let Z denote the number of edges. Construct a matrix $C \in \mathbb{R}^{\binom{n}{2} \times n}$ and a vector $c \in \mathbb{R}^{\binom{n}{2}}$ as follows: For an edge $e(i, j) \in E$, $C_{ei} = C_{ej} = 2\alpha$, and $c_e = -\alpha$. All the other entries are 0 in C. Matrix $W \in \mathbb{R}^{(\binom{n}{2}+1) \times n}$ is constructed as the concatenation of C and $\beta \cdot \vec{1} \in \mathbb{R}^{1 \times n}$, and vector $b \in \mathbb{R}^{n+1}$ is defined by $b_{[n]} = c \in \mathbb{R}^n$ and $b_{n+1} = -(n - q)\beta$, here β is a large constant. Let the given value $\delta = \sqrt[p]{Z\alpha^p}$.

References

[1] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[2] D. Needell and J. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[3] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, pp. 1982–2001, 2010.

[4] J. Chang, C. Li, B. Póczos, B. Kumar, and A. Sankaranarayanan, "One network to solve them all—solving linear inverse problems using deep projection models," in *International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 5889–5898.

[5] A. Mousavi and R. Baraniuk, "Learning to invert: Signal recovery via deep convolutional networks," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2272–2276.

[6] A. Bora, A. Jalal, E. Price, and A. Dimakis, "Compressed sensing using generative models," in *International Conference on Machine Learning (ICML)*, 2017, pp. 537–546.

[7] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9446–9454.

[8] G. Jagatap and C. Hegde, "Algorithmic guarantees for inverse imaging with untrained network priors," in *Neural Information Processing Systems (NeurIPS)*, 2019.

[9] T. L. Y. Wu, M. Rosca, "Deep compressed sensing," in *International Conference on Machine Learning (ICML)*, 2019.

[10] Z. Liu and J. Scarlett, "Information-theoretic lower bounds for compressive sensing with generative models," *IEEE Journal on Selected Areas in Information Theory*, 2020.

[11] V. Shah and C. Hegde, "Solving linear inverse problems using gan priors: An algorithm with provable guarantees," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4609–4613.

[12] P. Hand and V. Voroninski, "Global guarantees for enforcing deep generative priors by empirical risk," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 401–418, 2019.

[13] Q. Lei, A. Jalal, I. S. Dhillon, and A. G. Dimakis, "Inverting deep generative models, one layer at a time," in *Advances in Neural Information Processing Systems*, 2019, pp. 13 910–13 919.

[14] F. Latorre, A. Eftekhari, and V. Cevher, "Fast and provable admm for learning with generative priors," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 004–12 016.

[15] T. Nguyen, G. Jagatap, and C. Hegde, "Provable compressed sensing with generative priors via langevin dynamics," *ArXiv preprint arXiv:2102.12643*, 2021.

[16] J. Whang, Q. Lei, and A. Dimakis, "Compressed sensing with invertible generative models and dependent noise," *arXiv preprint arXiv:2003.08089*, 2020.

[17] M. Asim, A. Ahmed, and P. Hand, "Invertible generative models for inverse problems: mitigating representation error and dataset bias," in *International Conference on Machine Learning (ICML)*, 2020.

[18] E. M. Lindgren, J. Whang, and A. G. Dimakis, "Conditional sampling from invertible generative models with applications to inverse problems," *arXiv preprint arXiv:2002.11743*, 2020.

[19] R. Impagliazzo and R. Paturi, "On the complexity of k-sat," *Journal of Computer and System Sciences*, vol. 62, no. 2, pp. 367–375, 2001.

[20] R. Impagliazzo, R. Paturi, and F. Zane, "Which problems have strongly exponential complexity?" *Journal of Computer and System Sciences*, vol. 63, no. 4, p. 512–530, 2001.

[21] P. Indyk, "Beyond p vs. np: quadratic-time hardness for big data problems," in *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures*, 2017, pp. 1–1.

[22] D. Lokshtanov, D. Marx, S. Saurabh *et al.*, "Lower bounds based on the exponential time hypothesis," *Bulletin of EATCS*, vol. 3, no. 105, 2013.

[23] A. Rubinstein and V. V. Williams, "Seth vs approximation," *ACM SIGACT News*, vol. 50, no. 4, pp. 57–76, 2019.

[24] K. Bringmann, "Fine-grained complexity theory: Conditional lower bounds for computational geometry," in *Conference on Computability in Europe*. Springer, 2021, pp. 60–70.

[25] A. Backurs and P. Indyk, "Edit distance cannot be computed in strongly subquadratic time (unless seth is false)," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, 2015, pp. 51–58.

[26] K. Bringmann, "Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless seth fails," in *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE, 2014, pp. 661–670.

[27] K. Bringmann and M. Künnemann, "Quadratic conditional lower bounds for string problems and dynamic time warping," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, 2015, pp. 79–97.

[28] A. Abboud, A. Backurs, and V. V. Williams, "Tight hardness results for lcs and other sequence similarity measures," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, 2015, pp. 59–78.

[29] ——, "If the current clique algorithms are optimal, so is valiant's parser," *SIAM Journal on Computing*, vol. 47, no. 6, pp. 2527–2555, 2018.

[30] A. Abboud, V. Cohen-Addad, and H. Houdrougé, "Subquadratic high-dimensional hierarchical clustering," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[31] A. Backurs, P. Indyk, and L. Schmidt, "On the fine-grained complexity of empirical risk minimization: Kernel methods and neural networks," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[32] A. Gupte and V. Vaikuntanathan, "The fine-grained hardness of sparse linear regression," *arXiv preprint arXiv:2106.03131*, 2021.

[33] A. Aggarwal and J. Alman, "Optimal-degree polynomial approximations for exponentials and gaussian kernel density estimation," in *Proceedings of the 37th Computational Complexity Conference*, 2022, pp. 1–23.

[34] A. Rubinstein, "Hardness of approximate nearest neighbor search," in *Proceedings of the 50th annual ACM SIGACT symposium on theory of computing*, 2018, pp. 1260–1268.

[35] F. D. Keles, P. M. Wijewardena, and C. Hegde, "On the computational complexity of self-attention," in *International Conference on Algorithmic Learning Theory*. PMLR, 2023, pp. 597–619.

[36] H. Bennett, A. Golovnev, and N. Stephens-Davidowitz, "On the quantitative hardness of cvp," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2017, pp. 13–24.

[37] D. Aggarwal, H. Bennett, A. Golovnev, and N. Stephens-Davidowitz, "Fine-grained hardness of cvp (p)—everything that we can prove (and nothing else)," in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2021, pp. 1816–1835.