

Improved Inapproximability of VC Dimension and Littlestone’s Dimension via (Unbalanced) Biclique

Pasin Manurangsi ✉

Google Research, Bangkok, Thailand

Abstract

We study the complexity of computing (and approximating) VC Dimension and Littlestone’s Dimension when we are given the concept class explicitly. We give a simple reduction from Maximum (Unbalanced) Biclique problem to approximating VC Dimension and Littlestone’s Dimension. With this connection, we derive a range of hardness of approximation results and running time lower bounds. For example, under the (randomized) Gap-Exponential Time Hypothesis or the Strongish Planted Clique Hypothesis, we show a tight inapproximability result: both dimensions are hard to approximate to within a factor of $o(\log n)$ in polynomial-time. These improve upon constant-factor inapproximability results from [27].

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases VC Dimension, Littlestone’s Dimension, Maximum Biclique, Hardness of Approximation, Fine-Grained Complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2023.85

Acknowledgements I would like to thank ITCS 2023 reviewers for their helpful comments and suggestions.

1 Introduction

VC Dimension [33] and Littlestone’s Dimension [23] are two of the most fundamental quantities in learning theory; the former governs the sample complexity in PAC model to within a constant factor [3, 14], while the latter governs the sample complexity in online learning [23]. Due to this, it would be extremely useful to have an efficient algorithm for computing or approximating these dimensions for any given concept class $\mathcal{C} \subseteq 2^X$.

Given the importance of the two quantities, it should come as no surprise that this question has been investigated for several decades. Two models have been considered, based on whether the concept class is given explicitly. In the first “implicit” model, the input is a circuit which when given (indices of) $C \in \mathcal{C}$ and $x \in X$, evaluates to $C(x) \in \{0, 1\}$. For this model, both problems are hard: Schaefer proved that computing VC Dimension is Σ_p^3 -complete [31] whereas Littlestone’s Dimension is PSPACE-complete [32]. Furthermore, Mossel and Umans [29] showed that VC Dimension is Σ_p^3 -hard even to approximate to within a factor less than 2, and that it is AM-hard to approximate to within $n^{1-\epsilon}$ -factor for any constant $\epsilon > 0$.

The second model—which is the focus of the remainder of this work—is the “explicit” model, where the concept class is given as an $|X| \times |\mathcal{C}|$ -matrix. It is not hard to see that straightforward algorithms can solve both problems in $n^{O(\log n)}$ -time, because both dimensions are at most $O(\log n)$. Therefore, these problems cannot be NP-hard (unless NP is contained in $DTIME(n^{O(\log n)})$). Papadimitriou and Yannakakis [30] defined a class similar to NP but with “limited non-determinism” called LOGNP, and showed that VC Dimension is complete for this class. Frances and Litman [12] showed that Littlestone’s Dimension is also LOGNP-hard. A consequence of these results is that, assuming the Exponential Time



© Pasin Manurangsi;

licensed under Creative Commons License CC-BY 4.0

14th Innovations in Theoretical Computer Science Conference (ITCS 2023).

Editor: Yael Tauman Kalai; Article No. 85; pp. 85:1–85:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Hypothesis (ETH)¹, VC Dimension and Littlestone's Dimension cannot be solved in $n^{o(\log n)}$ time. More recently, Manurangsi and Rubinfeld [24] extended these lower bounds to rule out even approximation algorithms. Specifically, they showed, assuming ETH, that there is no $(2 - o(1))$ -approximation algorithm for VC Dimension and no $(1 + \epsilon)$ -approximation algorithm for Littlestone's Dimension that runs in $n^{\log^{1-o(1)} n}$ time. Nonetheless, it remains open whether any constant factor approximation is achievable in polynomial time:

► **Question 1.** *Is there polynomial-time $O(1)$ -approximation algorithm for VC Dimension or Littlestone's Dimension?*

As mentioned earlier, VC Dimension characterizes the sample complexity in PAC learning only to within a (large) constant factor [3, 14]. Therefore, Question 1 is not only a natural question but it can also be argued that $O(1)$ -approximation of VC Dimension is essentially as good as computing it exactly for the purpose of approximating the sample complexity.

1.1 Our Contributions

We answer the above question negatively, under certain computational complexity assumptions. Before we state our results, let us briefly recall the assumptions that we rely on. (All assumptions are formalized in Section 2.) In addition to ETH, we will use its strengthening called *Gap-ETH* [7, 26] which (roughly) says that there is no $2^{o(N)}$ -time algorithm even for an approximate version of 3SAT. The *Planted Clique Hypothesis* [19, 18] states that there is no polynomial-time algorithm that can distinguish between an Erdos-Renyi random graph and one with a planted clique of size $N^{\Omega(1)}$. The *Strongish Planted Clique Hypothesis* [28] is a strengthening of Planted Clique Hypothesis that rules out even $N^{o(\log N)}$ -time algorithms.

With all the assumptions in mind, we can now state our results. We divide the results into two groups, based on its inapproximability ratio. In the first group of results, we show, under ETH, Gap-ETH and Strongish Planted Clique Hypothesis, that there is no polynomial-time $\tilde{o}(\log n)$ -approximation algorithm for VC Dimension / Littlestone's Dimension, as stated below.

► **Theorem 2.** *Assuming ETH, there is no polynomial-time algorithm for approximating VC Dimension or Littlestone's Dimension to within $o\left(\frac{\log n}{(\log \log n)^\xi}\right)$ factor for some constant $\xi > 0$.*

► **Theorem 3.** *Assuming Gap-ETH, there is no polynomial-time algorithm for approximating VC Dimension or Littlestone's Dimension to within $o(\log n)$ factor.*

► **Theorem 4.** *Assuming the Strongish Planted Clique Hypothesis, there is no polynomial-time algorithm for approximating VC Dimension or Littlestone's Dimension to within $o(\log n)$ factor.*

Not only do these results answer Question 1, but the inapproximability factors are essentially the best possible: an algorithm that always output one is an $(\log n)$ -approximation algorithm.

If we only want to rule out any constant-factor approximation, then our lower bounds hold not only against polynomial-time algorithms but slightly quasi-polynomial time algorithms as well:

¹ ETH [16, 17] states that 3SAT cannot be solved in $2^{o(N)}$ time where N denotes the number of variables.

► **Theorem 5.** *Assuming the Strongish Planted Clique Hypothesis, there is no $n^{o(\log n)}$ -time algorithm for approximating VC Dimension or Littlestone’s Dimension to within any constant factor.*

► **Theorem 6.** *Assuming Gap-ETH, there is no $n^{o((\log n)^{1/3})}$ -time algorithm for approximating VC Dimension or Littlestone’s Dimension to within any constant factor.*

► **Theorem 7.** *Assuming ETH, there is no $n^{o\left(\frac{(\log n)^{1/3}}{(\log \log n)^\xi}\right)}$ -time algorithm for approximating VC Dimension or Littlestone’s Dimension to within any constant factor for some constant $\xi > 0$.*

Recall that there is also an $n^{O(\log n)}$ -time exact algorithm for computing both dimensions. Therefore, the running time lower bound in Theorem 5 is tight. On the other hand, while Theorems 6 and 7 achieve larger inapproximability factors compared to [27], the running time lower bounds are weaker than that in [27] due to technical reasons which we discuss more in Section 7.

Finally, we also note that under the Planted Clique Hypothesis, we can rule out constant factor approximation but only against polynomial-time algorithms:

► **Theorem 8.** *Assuming the Planted Clique Hypothesis, there is no polynomial-time algorithm for approximating VC Dimension or Littlestone’s Dimension to within any constant factor.*

We remark that our results are also strong enough to rule out any *fixed-parameter tractable algorithms*² with non-trivial approximation ratios. We defer the discussion on this to Appendix A.

Reduction from (Unbalanced) Biclique

All of our results are proved via a simple reduction from an unbalanced version of the Maximum Balanced Complete Bipartite Graph (Maximum Biclique) problem. Indeed, the fact that we can achieve multiple flexible results can be attributed to the rich literature on the Maximum Biclique problem. Furthermore, by identifying this as a hard problem underpinning the approximation of VC Dimension and Littlestone’s dimension, our reduction helps demystify the rather specific reductions in [27], which seem to be tailored towards starting from the Label Cover problem—and do not seem to be applicable e.g. with Planted Clique hypotheses.

To define the Maximum Biclique problem, we need a few additional terminologies: we use $K_{a,b}$ where $a, b \in \mathbb{N}$ to denote the bipartite complete graph (biclique) with a vertices on one side and b vertices on the other. We say that a bipartite graph $G = (A, B, E)$ *contains* $K_{a,b}$ if there exist a -size $S \subseteq A$ and b -size $T \subseteq B$ such that S, T induces a complete bipartite subgraph. Otherwise, if no such S, T exist, we say that G is $K_{a,b}$ -free.

Most work in the literature has considered the case of *balanced* biclique, i.e. $a = b$. The Maximum Balanced Biclique problem then asks for the maximum a such that the input graph G contains $K_{a,a}$. For the purpose of stating its inapproximability, it is useful to define the following “gap” version³, as stated below. Note that if (q_1, q_2) -Gap Biclique Problem cannot be efficiently solved, then there is no efficient (q_2/q_1) -approximation algorithm for Maximum Balanced Biclique.

² Please refer to e.g. [9, 5] for background on parameterized complexity.

³ We provide an additional discussion on gap problems in Section 2.2.

► **Definition 9** ((q_1, q_2) -Gap Biclique Problem). *Given a bipartite graph $G = (A, B, E)$, distinguish between the following two cases:*

- (YES) G contains K_{q_1, q_1} as a subgraph.
- (NO) G is K_{q_2, q_2} -free.

The Maximum Balanced Biclique problem turns out to be a challenging research question from hardness of approximation point of view. Despite the well-understood status of its non-bipartite counterpart [15, 21, 34], it is not even known if Maximum Balanced Biclique problem is NP-hard to approximate to within a factor of 1.0001. Fortunately for us, several hardness results are known under stronger assumptions [10, 2, 20, 24, 25, 4, 28].

As alluded to earlier, the exact problem we will use is not the balanced version above. We need an unbalanced version where one side of the biclique has t vertices and the other side has 2^t vertices, as defined below.

► **Definition 10** ((t_1, t_2) -Gap Exponential Biclique Problem). *Given a bipartite graph $G = (A, B, E)$, distinguish between the following two cases:*

- (YES) G contains $K_{t_1, 2^{t_1}}$ as a subgraph.
- (NO) G is $K_{t_2, 2^{t_2}}$ -free.

It should be noted that Gap Exponential Biclique is quite different from Gap Biclique. For example, to get a constant gap—say $t_1 \geq 2t_2$ —in the former, we need the number of vertices on one side to be 2^{t_1} in the YES case versus $2^{t_1/2} = \sqrt{2^{t_1}}$ in the NO case. This seems closer to the setting $q_2 = \sqrt{q_1}$ (i.e. $\sqrt{q_1}$ gap) in the Gap Biclique problem. Furthermore, Gap Exponential Biclique must have $t_1, t_2 \leq \log n$, meaning that the problem is solvable in time $n^{O(\log n)}$. Indeed, in our hardness results below, we have to resort to literature from *parameterized* hardness of Gap Biclique, which also considers the case $q_1, q_2 \ll n$. In some cases, we have to adapt the reductions slightly to get the parameters sufficiently strong for Gap Exponential Biclique.

Let us now turn our attention back to the main reduction, which is from Gap Exponential Biclique to the question of approximating VC / Littlestone's Dimension. In fact, our reduction proves the hardness of approximating *both* VC and Littlestone's Dimension simultaneously. Specifically, we show that the following gap problem is hard⁴:

► **Definition 11** ((d_1, d_2) -VC-Ldim Problem). *Given a concept class \mathcal{C} as a binary matrix, distinguish between the following two cases:*

- (YES) The VC Dimension of \mathcal{C} is at least d_1 .
- (NO) The Littlestone's Dimension of \mathcal{C} is less than d_2 .

With all the components defined, we can now state the properties of our reduction:

► **Theorem 12.** *If there is an $T(n)$ -time algorithm for the (d_1, d_2) -VC-Ldim Problem, there is an $O(T(n))$ -time (randomized) algorithm for the $(d_1/2, 2d_2)$ -Gap Exponential Biclique Problem.*

That is, any hardness of Gap Exponential Biclique with gap α can be translated to hardness of VC-Ldim Problem with gap $\alpha/4$. This allows us to easily translate the hardness of approximating of biclique problems to that of approximating VC / Littlestone's Dimension.

⁴ Recall that the Littlestone's Dimension of any class is no less than its VC Dimension.

Organization

The rest of this paper is organized as follows. We provide some additional definitions in Section 2. We then present the main reduction (Theorem 12) in Section 3. The subsequent three sections then establish concrete hardness results (stated in Section 1.1). Finally, we conclude with some discussion and open questions in Section 7.

2 Preliminaries

All approximation problems considered in this work are maximization problem. For these problems, we say that an algorithm⁵ has an approximation ratio (or factor) r if its output always lie between OPT/r and OPT , where OPT denote the true optimum for the input instance.

2.1 VC Dimension and Littlestone's Dimension

For a domain X , a concept c is simply a function from X to $\{0, 1\}$. A concept class \mathcal{C} is a set of concepts. For a subset $S \subseteq X$ and a concept c , let $c|_S$ denote the restriction of c on S , i.e. $c|_S : S \rightarrow \{0, 1\}$ such that $c|_S(x) = c(x)$ for all $x \in S$. Furthermore, define $\mathcal{C}|_S := \{c|_S \mid c \in \mathcal{C}\}$.

We now recall the definition of VC Dimension and Littlestone's Dimension.

► **Definition 13** (VC Dimension [33]). *The VC dimension of a concept class \mathcal{C} , denoted by $\text{VC}(\mathcal{C})$, is defined as the size of the largest subset $S \subseteq X$ such that $\mathcal{C}|_S$ contains all boolean functions on S .*

► **Definition 14** (Online Algorithm & Mistake Bound). *An online learning algorithm \mathbb{A} for a concept class \mathcal{C} is an algorithm that, at each step i , is given a sample x_i , the algorithm has to make a prediction z_i and afterwards the algorithm is told the correct label y_i . The mistake bound of \mathbb{A} for a sequence⁶ $(x_1, y_1), \dots, (x_T, y_T)$ is defined as the number of incorrect predictions (i.e. $\sum_{i \in [T]} \mathbf{1}[z_i \neq y_i]$).*

The mistake bound of \mathbb{A} for a concept class \mathcal{C} is defined as the maximum mistake bound of \mathbb{A} across all sequences $(x_1, y_1), \dots, (x_T, y_T)$ that are realizable by \mathcal{C} (i.e. there exists $c \in \mathcal{C}$ such that $y_i = c(x_i)$ for all $i \in [T]$).

► **Definition 15** (Littlestone's Dimension [23]). *The Littlestone's Dimension of \mathcal{C} , denoted by $\text{Ldim}(\mathcal{C})$, is defined as the minimum mistake bound for \mathcal{C} across all online algorithms \mathbb{A} .*

We note that the Littlestone's Dimension is sometimes defined in terms of mistake trees, but it was shown in [23] that this is equivalent to the above mistake bound definition. We choose to work with the mistake bound definition because it is more convenient in our proofs.

For convenience, we also extend these biclique-related terminologies to concept classes $\mathcal{C} \subseteq 2^X$, where the graph G is define as (X, \mathcal{C}, E) where $(x, c) \in E$ iff $c(x) = 1$.

⁵ We assume for simplicity that these algorithms are deterministic, although it is trivial to extend the formalism and proofs to randomized approximation algorithms.

⁶ Throughout this work, we assume w.l.o.g. that x_1, \dots, x_T are distinct.

2.2 Gap Problems

Gap problems we consider in this work (e.g. Definitions 9–11) belong to the class of *promise problems*. (See e.g. [13] for a more thorough treatment on the topic.) A promise problem consists of two disjoint languages L_{YES} and L_{NO} . A randomized algorithm \mathbb{A} is said to solve the promise problem (L_{YES}, L_{NO}) iff

- for all inputs $x \in L_{YES}$, $\Pr[\mathbb{A}(x) = YES] \geq 2/3$, and,
- for all inputs $x \in L_{NO}$, $\Pr[\mathbb{A}(x) = NO] \geq 2/3$.

Note that the algorithm are allowed to output arbitrarily for input $x \notin L_{YES} \cup L_{NO}$.

We say that an algorithm is a randomized reduction from a promise problem (L_{YES}, L_{NO}) to (L'_{YES}, L'_{NO}) if it takes in x and outputs x' such that

- If $x \in L_{YES}$, then $\Pr[x' \in L'_{YES}] \geq 2/3$.
- If $x \in L_{NO}$, then $\Pr[x' \in L'_{NO}] \geq 2/3$.

It is simple to see that if the reduction runs in $F(n)$ time (which also implies that it produces x' of size $N \leq F(n)$) and there is a randomized algorithm for solving (L'_{YES}, L'_{NO}) in $T(N)$ time, then there is a randomized algorithm for solving (L_{YES}, L_{NO}) in $O(T(F(n)))$ time.

Since most of the biclique hardness results we use are through randomized reductions, we will henceforth drop the “randomized” prefix and assume that any discussions related to promise problems are for randomized algorithms and reductions.

2.3 Exponential Time Hypotheses

Given a 3CNF formula Φ , we let $\text{SAT}(\Phi)$ denote the fraction of clauses that can be simultaneously satisfied by an assignment. A degree of variable with respect to a formula Φ is defined as the number of clauses it appears in. Furthermore, we define the $(1, 1 - \mu)$ -Gap 3SAT problem to be the problem of distinguishing between (YES) $\text{SAT}(\Phi) = 1$, and (NO) $\text{SAT}(\Phi) < 1 - \mu$.

ETH and Gap-ETH can be formulated as follows.

► **Assumption 16** (Exponential Time Hypothesis (ETH) [16, 17]). *No $2^{o(N)}$ -time algorithm can decide whether any given N -variable 3CNF formula Φ is satisfiable (i.e. $\text{SAT}(\Phi) = 1$).*

► **Assumption 17** (Gap-Exponential Time Hypothesis (Gap-ETH) [7, 26]). *For some constant $d, \mu > 0$, no $2^{o(N)}$ -time algorithm can solve the $(1, 1 - \mu)$ -Gap 3SAT problem on N -variable 3CNF formulae with maximum degree at most d .*

The nearly-linear size PCP of [6] gives a lower bound for Gap 3SAT problem under ETH that is slightly weaker than that of Gap-ETH:

► **Theorem 18** (Nearly-Linear Size PCP [6]). *Assuming ETH, there exists constant $v, d, \mu > 0$ such that no $2^{o(N/(\log N)^v)}$ -time algorithm can solve the $(1, 1 - \mu)$ -Gap 3SAT problem on N -variable 3CNF formulae with maximum degree at most d .*

2.4 Planted Clique Hypotheses

Let $\mathcal{G}(N, p)$ denote the Erdos-Renyi random graph distribution where there are N vertices and an edge exists between each pair of vertices with probability p . Furthermore, let $\mathcal{G}(N, p, \kappa)$ denote the distribution of a graph sampled from $\mathcal{G}(N, p)$ but afterwards get planted with a Clique of size κ .

► **Assumption 19** (Planted Clique Hypothesis (e.g. [19, 18])). *There exists $\delta > 0$ such that no $N^{O(1)}$ -time algorithm \mathbb{A} satisfies both of the following:*

- (Completeness) $\Pr_{G \sim \mathcal{G}(N,p,\lceil n^\delta \rceil)}[\mathbb{A}(G) = 1] \geq 2/3$.
- (Soundness) $\Pr_{G \sim \mathcal{G}(N,p)}[\mathbb{A}(G) = 1] \leq 1/3$.

► **Assumption 20** (Strongish Planted Clique Hypothesis [28]). *There exists $\delta > 0$ such that no $N^{o(\log N)}$ -time algorithm \mathbb{A} satisfies both of the following:*

- (Completeness) $\Pr_{G \sim \mathcal{G}(N,p,\lceil n^\delta \rceil)}[\mathbb{A}(G) = 1] \geq 2/3$.
- (Soundness) $\Pr_{G \sim \mathcal{G}(N,p)}[\mathbb{A}(G) = 1] \leq 1/3$.

3 Reducing Biclique to VC/Littlestone's Dimension

In this section, we present our main reduction (Theorem 12). Before we do so, let us describe the high-level overview of the reduction. Let $G = (A, B, E)$ be an input instance of (d_1, d_2) -Gap Exponential Biclique. Perhaps the simplest reduction one can attempt is to simply let the class \mathcal{C} be corresponding to the adjacency matrix of G , i.e. $X = A, \mathcal{C} = B$ and $c(a) = 1$ iff $(a, c) \in E$.

This simple reduction actually “almost works”. Specifically, the soundness (i.e. NO case) actually holds. This can easily be seen for VC Dimension, because if $\mathcal{C}|_S$ consists of all boolean functions on S , then it must contain a $K_{\lfloor |S|/2 \rfloor, 2^{\lfloor |S|/2 \rfloor}}$. This means that in the NO case, the VC Dimension must be less than $2d_2$. Below, in Lemma 21, we show that the same bound holds even against Littlestone's Dimension.

While the soundness holds, the completeness (i.e. YES case) of the reduction clearly fails: even if G is a complete bipartite graph, the corresponding \mathcal{C} has VC Dimension of just zero. The problem here lies in the fact that “there are too many ones” in the adjacency matrix. It turns out that this is very simple to fix: just flip each entry in the adjacency matrix to zero w.p. $1/2$. This means that the submatrix induced by the biclique is now a random matrix; it is easy to show that such a matrix has high VC Dimension with large probability (Lemma 22). This completes the high-level overview of the reduction.

3.1 Some Helpful Lemmas

Before we formalize the reduction, let us state a few necessary lemmas. First is the aforementioned bound for the Littlestone's Dimension for biclique-free concept class, which will be used in the soundness proof.

► **Lemma 21.** *Let $t \in \mathbb{N}$ and \mathcal{C} be any $K_{t,2^t}$ -free concept class. Then, we must have $\text{Ldim}(\mathcal{C}) < 2t$.*

To prove the above lemma, it is convenient to define $\mathcal{C}[(x_1, y_1), \dots, (x_T, y_T)]$ for any sequence $(x_1, y_1), \dots, (x_T, y_T)$ as the class of concepts in \mathcal{C} that is consistent with the sequence. More formally,

$$\mathcal{C}[(x_1, y_1), \dots, (x_T, y_T)] := \{c \in \mathcal{C} \mid \forall i \in [T], c(x_i) = y_i\}.$$

Proof of Lemma 21. To give the desired upper bound on $\text{Ldim}(\mathcal{C})$, it suffices to give an online algorithm \mathbb{A} that achieves a mistake bound of less than $2t$ on \mathcal{C} . After receiving $(x_1, y_1), \dots, (x_{i-1}, y_{i-1})$ and x_i , the algorithm gives prediction z_i based on the following rule:

1. If $|\mathcal{C}[(x_1, y_1), \dots, (x_{i-1}, y_{i-1})]| \geq 2^t$, the algorithm outputs 0.
2. Otherwise, the algorithm outputs $\arg\max_{z \in \{0,1\}} |\mathcal{C}[(x_1, y_1), \dots, (x_{i-1}, y_{i-1}), (x_i, z)]|$ (tie broken arbitrarily).

Note that since $\mathcal{C}[(x_1, y_1), \dots, (x_{i-1}, y_{i-1})]$ is non-increasing, the algorithm will first be in the first stage (in the first case) and then moves to the second stage (the second case).

Since \mathcal{C} is $K_{t, 2^t}$ -free, \mathbb{A} will make at most t mistakes in the first stage⁷. Once \mathbb{A} reaches the second stage, the size $|\mathcal{C}[(x_1, y_1), \dots, (x_{i-1}, y_{i-1})]|$ reduces by a factor of (at least) two with each mistake. Since it starts off with value less than 2^t , \mathbb{A} must make less than t mistakes in the second stage. Therefore, in total \mathbb{A} will make less than $2t$ mistakes. ◀

The second lemma we need, which will be used in the completeness argument, is that a class consisting of independent random concepts have large VC Dimensions:

► **Lemma 22.** *Let S be any t -size set and $\mathcal{C} \subseteq \{0, 1\}^S$ be a concept class of size 2^t , where each concept is independently uniformly sampled at random from $\{0, 1\}^S$. If $t \geq 4$, then $\text{VC}(\mathcal{C}) \geq t/2$ with probability at least $2/3$.*

Proof. Consider any subset $S' \subseteq S$ of size $t' := \lceil t/2 \rceil$. We have

$$\begin{aligned} \Pr[\text{VC}(\mathcal{C}) < t/2] &\leq \Pr[\exists f \in \{0, 1\}^{S'}, f \notin \mathcal{C}|_{S'}] \\ &\leq \sum_{f \in \{0, 1\}^{S'}} \Pr[f \notin \mathcal{C}|_{S'}] \\ &= \sum_{f \in \{0, 1\}^{S'}} (1 - 1/2^{t'})^{|C|} \\ &\leq 2^{t'} \cdot e^{-|C|/2^{t'}} = 2^{\lceil t/2 \rceil} \cdot e^{-2^{\lfloor t/2 \rfloor}} < 1/3, \end{aligned}$$

where the second inequality is due to the union bound and the first equality is due to the fact that each function in \mathcal{C} is picked independently u.a.r. from $\{0, 1\}^S$. ◀

3.2 The Reduction

We are now ready to prove Theorem 12.

Proof of Theorem 12. We will provide a reduction from the (d_1, d_2) -Gap Exponential Biclique problem to the $(d_1/2, 2d_2)$ -VC-Ldim problem such that the problem size remains the same, which implies the theorem statement. (See discussion in Section 2.2.)

Given an instance $G = (A, B, E)$ for the (d_1, d_2) -Gap Exponential Biclique Problem. Let $X = A$ and, for every $b \in B$, create a concept c_b in \mathcal{C} where $c_b(a)$ for each $a \in A$ is defined as follows.

- If $(a, b) \notin E$, let $c_b(a) = 0$.
- If $(a, b) \in E$, let $c_b(a) \in \{0, 1\}$ be drawn uniformly at random.

(Completeness) We may assume w.l.o.g. that $d_1 \geq 4$; otherwise, the problem can be trivially solved in polynomial time. Suppose that G contains $K_{d_1, 2^{d_1}}$, i.e. there exists $S \subseteq A, T \subseteq B$ of sizes $d_1, 2^{d_1}$ respectively such that $(s, t) \in E$ for all $s \in S, t \in T$. This, together with the definition of \mathcal{C} , means that $c_t|_S$ for each $t \in T$ is drawn independently uniformly at random among $\{0, 1\}^S$. Thus, we may apply Lemma 22 to conclude that $\text{VC}(\mathcal{C}) \geq \text{VC}(\mathcal{C}|_S)$ is at least $d_1/2$ with probability $2/3$.

(Soundness) If G is $K_{d_2, 2^{d_2}}$ -free, then \mathcal{C} is also $K_{d_2, 2^{d_2}}$ -free (because the corresponding graph of \mathcal{C} is a subgraph of G). Therefore, Lemma 21 ensures that $\text{Ldim}(\mathcal{C}_i) \leq 2d_2$. ◀

⁷ Otherwise, x_1, \dots, x_{i-1} and the points in $\mathcal{C}[(x_1, y_1), \dots, (x_{i-1}, y_{i-1})]$ would induce $K_{t, 2^t}$.

4 (Gap-)ETH-Hardness of Exponential Biclique

Having presented the reduction, we will next prove concrete hardness results, starting with those based on Gap-ETH and ETH. To prove these, we start by making a straightforward observation: there is the trivial reduction from (d_1, d_2) -Gap Biclique to $(\lfloor \log d_1 \rfloor, d_2)$ -Gap Exponential Biclique (i.e. keeping the input graph exactly the same). This yields the following result.

► **Observation 23.** *If there exists an $T(n)$ -time algorithm for the $(\lfloor \log d_1 \rfloor, d_2)$ -Gap Exponential Biclique Problem, there exists an $T(n)$ -time algorithm for the (d_1, d_2) -Gap Biclique Problem.*

We will also need the following reduction from Gap 3SAT to Gap Biclique due to [4] (which is in turn a modification of that from [24]).

► **Theorem 24** ([4, Theorem 5.13]). *For any constants $d, \mu > 0$, there exists a constant $\gamma > 0$ and a reduction from $(1, 1 - \mu)$ -Gap 3SAT problem on n -variable 3CNF formulae with maximum degree at most d to $(n^{\gamma/\sqrt{r}}, r)$ -Gap Biclique on graphs of size n for any sufficiently large r . Furthermore, $n = 2^{\Theta_{d,\mu}(N/\sqrt{r})}$ and the reduction runs in $n^{O(1)}$ time.*

4.1 Gap-ETH-Hardness

We will now prove the Gap-ETH hardness results (Theorems 3 and 6) by plugging in an appropriate value of r in Theorem 24 and then apply our reduction (Theorem 12) afterwards. For the $o(\log n)$ factor inapproximability result, we set r to be slowly growing function of N , as formalized below.

Proof of Theorem 3. Let $\alpha(n)$ denote any function⁸ such that $\alpha(n) = o(\log n)$. We will prove below that there is no polynomial-time algorithm for $(4\alpha(n)d, d)$ -Gap Exponential Biclique, where d is a parameter to be set below. By Theorem 12, this immediately implies that there is no polynomial-time $\alpha(n)$ -approximation algorithm for VC dimension or Littlestone's dimension.

Let Φ be the input to the $(1, 1 - \mu)$ -Gap 3SAT problem. Let $r = \lceil \sigma \cdot \sqrt{N/\alpha(2^N)} \rceil$ where $\sigma > 0$ is a sufficiently small constant to be chosen later. Note here that $r = \omega(1)$ because $\alpha(2^N) = o(\log(2^N)) = o(N)$. We apply the reduction in Theorem 24 to produce an input graph G' of size $n = 2^{\Theta(N/\sqrt{r})} \leq 2^{o(n)}$ for $(n^{\gamma/\sqrt{r}}, r)$ -Gap Biclique. Observation 23 also implies that this is an instance for $(\lfloor \log(n^{\gamma/\sqrt{r}}) \rfloor, r)$ -Gap Exponential Biclique. Let $t_1 = \lfloor \log(n^{\gamma/\sqrt{r}}) \rfloor = \Theta(N/r)$ and $d = t_2 = r$. From our choice of r , we have $t_1/t_2 = \Theta(1/\sigma^2) \cdot \alpha(2^N)$. Thus, by taking σ to be a sufficiently small constant, the ratio between t_1, t_2 is at least $4\alpha(2^N)$, which is in turn no less than $4\alpha(n)$ for any sufficiently large N .

Thus, if there were a polynomial-time algorithm for $(4\alpha(n)d, d)$ -Gap Exponential Biclique Problem, then we could apply the above reduction and run it on the resulting graph G' to solve the $(1, 1 - \mu)$ -Gap 3SAT problem in time $2^{o(N)}$. This would violate Gap-ETH. ◀

For Theorem 6, we instead choose $r = \Theta(\sqrt{N})$.

⁸ Henceforth, we assume w.l.o.g. that the approximation ratio (denoted by $\alpha(n)$) is a non-decreasing function of the problem size n ; this is w.l.o.g. as otherwise we can always consider $\tilde{\alpha}(n) := \max_{n' \leq n} \alpha(n')$ instead.

Proof of Theorem 6. Let $C > 1$ be any constant. We will prove below that there is no $n^{o((\log n)^{1/3})}$ -time algorithm for $(4Cd, d)$ -Gap Exponential Biclique, where d is a parameter to be set below. By Theorem 12, this immediately implies that there is no $n^{o((\log n)^{1/3})}$ -time C -approximation algorithm for VC dimension or Littlestone's dimension.

Let Φ be the input to the $(1, 1 - \mu)$ -Gap 3SAT problem. Let $r = \lceil \sigma \cdot \sqrt{N} \rceil$ where $\sigma > 0$ is a sufficiently small constant to be chosen later. We apply the reduction in Theorem 24 to produce an input graph G' of size $n = 2^{\Theta(N/\sqrt{r})} = 2^{\Theta(N^{3/4})}$ for $(n^{\gamma/\sqrt{r}}, r)$ -Gap Biclique. Observation 23 also implies that this is an instance for $(\lfloor \log(n^{\gamma/\sqrt{r}}) \rfloor, r)$ -Gap Exponential Biclique. Let $t_1 = \lfloor \log(n^{\gamma/\sqrt{r}}) \rfloor = \Theta(N/r)$ and $d = t_2 = r$. From our choice of r , we have $t_1/t_2 = \Theta(1/\sigma^2)$. Thus, by taking σ to be a sufficiently small constant, the ratio between t_1, t_2 is at least $4C$ as desired.

Thus, if there were a $n^{o((\log n)^{1/3})}$ -time algorithm for $(4Cd, d)$ -Gap Exponential Biclique Problem, then we could apply the above reduction and run it on the resulting graph G' to solve the $(1, 1 - \mu)$ -Gap 3SAT problem in time $(2^{\Theta(N^{3/4})})^{o(\log(2^{\Theta(N^{3/4})})^{1/3})} = 2^{o(N)}$. This would violate Gap-ETH. \blacktriangleleft

4.2 ETH-Hardness

The proofs of Theorems 2 and 7 are nearly identical to those of Theorems 3 and 6, except that we have to set r to be larger than the corresponding Gap-ETH-based results by a polylogarithmic in N factor. This is to compensate for the weaker running time lower bound we have under ETH from Theorem 18.

Proof of Theorem 2. Let $\xi = 4v$ where v is the constant from Theorem 18. Let $\alpha(n)$ denote any function such that $\alpha(n) = o\left(\frac{\log n}{(\log \log n)^\xi}\right)$. We will prove below that there is no polynomial-time algorithm for $(4\alpha(n)d, d)$ -Gap Exponential Biclique, where d is a parameter to be set below. By Theorem 12, this immediately implies that there is no polynomial-time $\alpha(n)$ -approximation algorithm for VC dimension or Littlestone's dimension.

Let Φ be the input to the $(1, 1 - \mu)$ -Gap 3SAT problem. Let $r = \lceil \sigma \cdot \sqrt{N/\alpha(2^N)} \rceil$ where $\sigma > 0$ is a sufficiently small constant to be chosen later. Note here that $r = \omega((\log N)^{2v})$ because $\alpha(2^N) = o\left(\frac{\log(2^N)}{(\log \log(2^N))^\xi}\right) = o\left(\frac{N}{(\log N)^\xi}\right)$. We apply the reduction in Theorem 24 to produce an input graph G' of size $n = 2^{\Theta(N/\sqrt{r})} \leq 2^{o(N/(\log N)^v)}$ for $(n^{\gamma/\sqrt{r}}, r)$ -Gap Biclique. Observation 23 also implies that this is an instance for $(\lfloor \log(n^{\gamma/\sqrt{r}}) \rfloor, r)$ -Gap Exponential Biclique. Let $t_1 = \lfloor \log(n^{\gamma/\sqrt{r}}) \rfloor = \Theta(N/r)$ and $d = t_2 = r$. From our choice of r , we have $t_1/t_2 = \Theta(1/\sigma^2) \cdot \alpha(2^N)$. Thus, by taking σ to be a sufficiently small constant, the ratio between t_1, t_2 is at least $4\alpha(2^N)$, which is in turn no less than $4\alpha(n)$ for any sufficiently large N .

Thus, if there were a polynomial-time algorithm for $(4\alpha(n)d, d)$ -Gap Exponential Biclique Problem, then we could apply the above reduction and run it on the resulting graph G' to solve the $(1, 1 - \mu)$ -Gap 3SAT problem in time $2^{o(N/(\log N)^v)}$. From Theorem 18, this would violate ETH. \blacktriangleleft

Proof of Theorem 7. Let $\xi = v$ be the constant from Theorem 18. Let $C > 1$ be any constant. We will prove below that there is no $n^{o\left(\frac{(\log n)^{1/3}}{(\log \log n)^\xi}\right)}$ -time algorithm for $(4Cd, d)$ -Gap Exponential Biclique, where d is a parameter to be set below. By Theorem 12, this immediately implies that there is no $n^{o\left(\frac{(\log n)^{1/3}}{(\log \log n)^\xi}\right)}$ -time C -approximation algorithm for VC dimension or Littlestone's dimension.

Let Φ be the input to the $(1, 1 - \mu)$ -Gap 3SAT problem. Let $r = \lceil \sigma \cdot \sqrt{N} \rceil$ where $\sigma > 0$ is a sufficiently small constant to be chosen later. We apply the reduction in Theorem 24 to produce an input graph G' of size $n = 2^{\Theta(N/\sqrt{r})} = 2^{\Theta(N^{3/4})}$ for $(n^{\gamma/\sqrt{r}}, r)$ -Gap Biclique. Observation 23 also implies that this is an instance for $(\lfloor \log(n^{\gamma/\sqrt{r}}) \rfloor, r)$ -Gap Exponential Biclique. Let $t_1 = \lfloor \log(n^{\gamma/\sqrt{r}}) \rfloor = \Theta(N/r)$ and $d = t_2 = r$. From our choice of r , we have $t_1/t_2 = \Theta(1/\sigma^2)$. Thus, by taking σ to be a sufficiently small constant, the ratio between t_1, t_2 is at least $4C$ as desired.

Thus, if there were a $n^{o\left(\frac{(\log n)^{1/3}}{(\log \log n)^\xi}\right)}$ -time algorithm for $(4Cd, d)$ -Gap Exponential Biclique Problem, then we could apply the above reduction and run it on the resulting graph G' to solve the $(1, 1 - \mu)$ -Gap 3SAT problem in time $(2^{\Theta(N^{3/4})})^{o\left(\frac{\log(2^{\Theta(N^{3/4})})^{1/3}}{(\log \log(2^{\Theta(N^{3/4})})^\xi)}\right)} = 2^{o(N/(\log N)^v)}$. From Theorem 18, this would violate ETH. \blacktriangleleft

5 Hardness from Planted Clique Hypothesis

We next move on to prove hardness result based on the Planted Clique Hypothesis (Theorem 8). We start by recalling that the Planted Clique Hypothesis by itself already implies a fairly strong hardness of Gap Biclique:

► **Lemma 25** (Folklore). *Assuming the Planted Clique Hypothesis, for some constants $\delta \in (0, 1), \zeta > 1$, there is no polynomial-time algorithm for $(\lceil N^\delta \rceil, \zeta \lceil \log N \rceil)$ -Gap Biclique.*

The above result is folklore; for completeness, we provide a proof sketch in Appendix B.

While Lemma 25 is indeed a strong result, it is not yet enough for us. Specifically, if we want to apply Observation 23 from the previous section, then we would now only reduce to the $(\delta \log N, \zeta \log N)$ -Gap Exponential Biclique problem. However, this is trivial because $\delta < 1 < \zeta$.

5.1 One-Sided Graph Product

The above problem is due to the fact that the YES case is too small to apply the observation directly. To boost the YES case, we use the following “one-sided” graph product, which helps boost one side of the biclique in the YES case while not increasing the NO case. The reduction and its main properties are described below.

One-Sided Graph Product

Input: Bipartite Graph $G = (A, B, E)$ where $|A|, |B| = N$, positive integers ℓ_R .

Output: Graph $G' = (A', B', E')$.

The graph G' is constructed as follows.

1. Let $A' = A$.
2. Let $B' = B^{\ell_R}$ where we view each element of B' as a multiset.
3. For every $a \in A', T \in B'$, include (a, T) in E' iff $\{a\}, T$ induces a biclique in G .

■ **Figure 1** One-Sided Graph Product.

► **Lemma 26.** *Let $\delta \in (0, 1)$ be a constant and N be sufficiently large (depending on δ). Let G, G' be as in Figure 1. Then, we have*

- (Completeness) *If G contains K_{q_1, q_1} , $q_1 \geq t_1$ and $\ell_R \log(q_1) \geq t_1$, then G' contains $K_{t_1, 2^{t_1}}$.*
- (Soundness) *If G is K_{q_2, q_2} -free, $q_2 \leq t_2$ and $\ell_R \log(q_2) \leq t_2$, then G' is $K_{t_2, 2^{t_2}}$ -free.*

Proof.

(Completeness) Let $P \subseteq A, Q \subseteq B$ be a K_{q_1, q_1} -biclique in G . Observe that P and Q^{ℓ_R} induces a biclique in G' . We also have $|Q^{\ell_R}| = q_1^{\ell_R} \geq 2^{t_1}$ where the inequality follows from the second assumption on the parameters. Thus, we can conclude that G' contains $K_{t_1, 2^{t_1}}$ as desired.

(Soundness) Suppose for the sake of contradiction that G' contains $K_{t_2, 2^{t_2}}$, i.e. there exists $P \subseteq A', \mathcal{T} \subseteq B'$ of sizes $t_2, 2^{t_2}$ respectively that induces a biclique. By definition, this also means that P and $Q := (\bigcup_{T \in \mathcal{T}} T) \subseteq B$ induce a biclique in the graph G . Since $\mathcal{T} \subseteq Q^{\ell_R}$, we must have $Q \geq |\mathcal{T}|^{1/\ell_R} \geq q_2$, where the second inequality follows from our assumption that $\ell_R \log(q_2) \leq t_2$. Recall also that we assume that $t_2 \geq q_2$. As a result, we can conclude that G contains K_{q_2, q_2} . ◀

5.2 Proof of Theorem 8

We can now prove the desired hardness (Theorem 8) via the one-sided graph product where we pick ℓ_R to be a sufficiently large constant.

Proof of Theorem 8. Let $C > 1$ be any constant. We will prove below that there is no polynomial-time algorithm for $(4Cd, d)$ -Gap Exponential Biclique, where d is a parameter to be set below. By Theorem 12, this immediately implies that there is no polynomial-time C -approximation algorithm for VC dimension or Littlestone's dimension.

Let G be the input to the $(\lceil N^\delta \rceil, \zeta \lceil \log N \rceil)$ -Gap Biclique problem. Let $q_1 = \lceil N^\delta \rceil, d = q_2 = \lceil \zeta \log N \rceil$. We use the reduction in Figure 1 with $t_1 = 4Cd, t_2 = d, \ell_R = \lceil t_1 / \log(q_1) \rceil$. The reduction runs in polynomial time (because $\ell_R \leq O(1)$). We now check the completeness and soundness:

(Completeness) By our setting of parameters, $\ell_R \log(q_1) \geq t_1$. Furthermore, for any sufficiently large N , we have $q_1 = \Theta(N^\delta)$ is at least $t_1 = \Theta(\log N)$. Thus, applying the completeness of Lemma 26, we conclude that, if G contains K_{q_1, q_1} , then the G' contains $K_{t_1, 2^{t_1}}$.

(Soundness) By our setting of parameters, $q_2 = t_2$. Furthermore, $\ell_R \log(q_2) = O(\log \log N)$ must be less than $t_2 = \Theta(\log N)$ for any sufficiently large N . Thus, applying the completeness of Lemma 26, we can conclude that, if G is K_{q_2, q_2} -free, then the G' is $K_{t_1, 2^{t_1}}$ -free.

Thus, if there were a polynomial-time algorithm for $(4Cd, d)$ -Gap Exponential Biclique Problem, then we could apply the above reduction and run it on the resulting graph G' to solve the $(\lceil N^\delta \rceil, \zeta \lceil \log N \rceil)$ -Gap Biclique problem in polynomial time. From Lemma 25, this would violate the Strongish Planted Clique Hypothesis. ◀

6 Hardness from Strongish Planted Clique Hypothesis

Finally, we will prove the hardness results based on the Strongish Planted Clique Hypothesis (Theorems 4 and 5). Again, we start by recalling that the Strongish Planted Clique Hypothesis already implies a fairly strong hardness of Gap Biclique, as stated below. (Proof sketch in Appendix B.)

► **Lemma 27** (Folklore). *Assuming the Strongish Planted Clique Hypothesis, for some constants $\delta \in (0, 1), \zeta > 1$, there is no $N^{o(\log N)}$ -time algorithm for $(\lceil N^\delta \rceil, \lceil \zeta \cdot \log N \rceil)$ -Gap Biclique.*

The proof of Theorem 5 is exactly the same as Theorem 8 except that we start with an $N^{\Omega(\log N)}$ running time lower bound (in Lemma 27) instead of just polynomial running time lower bound (in Lemma 25), so we end up with an $n^{\Omega(\log n)}$ running time lower bound as well.

For the remainder of this section, we focus on proving the tight inapproximability ratio (Theorem 4).

6.1 Two-Sided Randomized Graph Product

Again, we employ graph products, but this time the product is two-sided and furthermore is randomized. This “two-sided randomized graph product” has been used several times in literature (e.g. [20]) but we will consider different parameter regimes compared to previous works. Therefore, we state the randomized graph product and its properties in full below.

Two-Sided Randomized Graph Product [1]

Input: Bipartite Graph $G = (A, B, E)$ where $|A|, |B| = N$, positive integers n, ℓ .

Output: Graph $G' = (A', B', E')$.

The graph G' is constructed as follows.

1. For each $i \in [n]$, independently sample $S_i \sim A^\ell$. Then, let $A' = \{S_1, \dots, S_n\}$.
2. For each $j \in [n]$, independently sample $T_j \sim B^\ell$. Then, let $B' = \{T_1, \dots, T_n\}$.
3. For every $i, j \in [n]$, include (S_i, T_j) in E' iff S_i, T_j induces a biclique in G .

■ **Figure 2** Two-Sided Randomized Graph Product.

► **Lemma 28.** *Let $\delta \in (0, 1)$ be a constant and N be sufficiently large (depending on δ). Furthermore, suppose that $n \geq 102^{t_1} \cdot (N/q_1)^\ell$, $n \leq 1000N^{(1-0.5\delta)^\ell}$, $20 \leq \ell$ and $0.005\delta \cdot t_2 \cdot \ell \geq q_2$. Then, the reduction in Figure 2 is a reduction from (q_1, q_2) -Gap Biclique to $(2^{t_1}, t_2)$ -Gap Biclique.*

To prove Lemma 28, we require a lemma showing that $S_1, \dots, S_n, T_1, \dots, T_n$ (when viewed as sets) are “dispersers”, meaning that a union of a certain number of them is sufficiently large. [28] also used such a property and their lemma (stated below) will be sufficient for us.

► **Lemma 29** ([28, Lemma 7]). *Let $\gamma > 0$, and suppose $n \leq 1000N^{(1-\gamma)^\ell}$, $20 \leq \ell$. Let $S_1, \dots, S_n, T_1, \dots, T_n$ be as sampled as in Figure 2. Then, with probability at least 0.9, the following event occurs: for every $M \subseteq [n]$ with $|M| \leq N^{0.99\gamma}/\ell$, we have $|\bigcup_{i \in M} S_i| \geq 0.01\gamma|M|\ell$ and $|\bigcup_{i \in M} T_i| \geq 0.01\gamma|M|\ell$.*

Proof of Lemma 28.

(Completeness) Let $P \subseteq A, Q \subseteq B$ be a K_{q_1, q_1} -biclique in G . Observe that $(A' \cap P^\ell)$ and $(B' \cap Q^\ell)$ induce a biclique in G' . Notice also that each S_i belongs to P^ℓ with probability $(q_1/N)^\ell$. Therefore, we have $\mathbb{E}[|A' \cap P^\ell|] = n \cdot (q_1/N)^\ell \geq 102^{t_1}$, where the inequality follows from our assumption on parameters. Applying standard concentration bounds, we can conclude that $\Pr[|A' \cap P^\ell| \geq 2^{t_1}] \geq 0.95$. An analogous argument shows that $\Pr[|B' \cap Q^\ell| \geq 2^{t_1}] \geq 0.95$. Applying the union bound, G' contains $K_{2^{t_1}, 2^{t_1}}$ with probability at least 0.9.

(Soundness) From our assumption on the parameters, we can apply Lemma 29 (with $\gamma = 0.5\delta$), which guarantee that with probability 0.9, the following holds:

$$\forall M \in \binom{[n]}{t_2}, \quad \left| \bigcup_{i \in M} S_i \right| \geq q_2 \text{ and } \left| \bigcup_{i \in M} T_i \right| \geq q_2. \quad (1)$$

We will prove that, when this holds, if G is K_{q_2, q_2} -free, then G' is K_{t_2, t_2} -free.

Suppose contrapositively that G' contains K_{t_2, t_2} , i.e. there exists $\mathcal{P} \subseteq A'$, $\mathcal{Q} \subseteq B'$ each of size t_2 respectively that induces a biclique in G' . By definition, this also means that $P := (\bigcup_{S \in \mathcal{P}} S) \subseteq A$ and $Q := (\bigcup_{T \in \mathcal{Q}} T) \subseteq B$ induce a biclique in the graph G . By (1), we must have $|P|, |Q| \geq q_2$, meaning that G contains K_{q_2, q_2} . ◀

6.2 Proof of Theorem 4

We can now prove Theorem 4 by applying the above two-sided randomized graph product with appropriate parameters (e.g. ℓ is $o(\log N)$).

Proof of Theorem 4. Let $\alpha(n)$ denote any function such that $\alpha(n) = o(\log n)$. We will prove below that there is no polynomial-time algorithm for $(4\alpha(n)d, d)$ -Gap Exponential Biclique, where d is a parameter to be set below. By Theorem 12, this immediately implies that there is no polynomial-time $\alpha(n)$ -approximation algorithm for VC dimension or Littlestone's dimension.

Let G denote the input to the $(\lceil N^\delta \rceil, \zeta \lceil \log N \rceil)$ -Gap Biclique problem. Let $q_1 = \lceil N^\delta \rceil$, $q_2 = \lceil \zeta \log N \rceil$ and $d = \lceil (\log N)^{2/3} / \alpha(N^{\log N})^{1/3} \rceil$. Note that $d = \omega(1)$ because $\alpha(N^{\log N}) = o(\log(N^{\log N})) = o(\log^2 N)$. We use the reduction in Figure 2 with $t_1 = \alpha(N^{\log N}) \cdot d$, $t_2 = d$, $\ell = \max \left\{ 20, \left\lceil \frac{200q_2}{t_2^\delta} \right\rceil \right\}$, $n = \lceil 102^{t_1} \cdot (N/q_1)^\ell \rceil$. Note that $\ell = O(q_2/t_2) = o(\log N)$. Furthermore, $t_1/\ell = O(\alpha(N^{\log N}) \cdot d^2 / \log N) = O(\alpha(N^{\log N})^{1/3} \cdot (\log N)^{1/3}) = o(\log N)$. Therefore, we have $n = O(2^{t_1} \cdot N^{(1-\delta)\ell}) = N^{(1-\delta+o(1))\ell} \leq N^{o(\log N)}$. The second-to-last inequality also implies that $n \leq N^{(1-0.5\delta)\ell}$ for any sufficiently large N . In other words, for any sufficiently large N , the parameters satisfy conditions in Lemma 28.

Thus, the reduction runs in $n^{O(1)} = N^{o(\log N)}$ time and produces an instance to the $(2^{t_1}, t_2)$ -Gap Biclique Problem. By Observation 23, this is also an instance for the (t_1, t_2) -Gap Exponential Biclique problem. By definition, $t_1/t_2 = \alpha(N^{\log N}) \geq \alpha(n)$ for any sufficiently large N .

Thus, if there were a polynomial-time algorithm for $(4\alpha(n)d, d)$ -Gap Exponential Biclique Problem, then we could apply the above reduction and run it on the resulting graph G' to solve the $(\lceil N^\delta \rceil, \zeta \lceil \log N \rceil)$ -Gap Biclique problem in time $N^{o(\log N)}$. From Lemma 27, this would violate the Strongish Planted Clique Hypothesis. ◀

7 Conclusion and Discussion

In this work, we establish several hardness of approximation results and running time lower bounds for approximating VC Dimension and Littlestone's Dimension. For polynomial-time algorithms, we rule out $o(\log n)$ -approximation Theorems 2 and 4, which is tight. For any constant factor approximation, we rule out algorithms that runs in $n^{o(\log n)}$ time under the Strongish Planted Clique Hypothesis but only $n^{\delta(\log^{1/3} N)}$ time under ETH/Gap-ETH. The latter is not a coincidence: the best running time lower bounds for finding k balanced biclique known under ETH/Gap-ETH is only $n^{\Omega(\sqrt{k})}$ even for the exact version of the problem [22] while the trivial algorithm runs in $n^{O(k)}$ time. Closing this gap is a well-known open problem

in parameterized complexity. Due to our reduction, improved running time lower bounds for finding k balanced biclique may also lead to improved running time lower bounds for VC Dimension and Littlestone’s Dimension.

In addition to closing the gap in the time lower bounds, another interesting question—originally posed in [12]—is whether there is an efficient online learner for any given concept class with approximately optimal mistake bound (as defined in Definition 14). Ostensibly, this problem is very similar to that of approximating Littlestone’s Dimension. In fact, in the *exact* setting, [12] showed that there is an efficient online learner with *exactly* optimal mistake bound iff there is an efficient algorithm for *exactly* computing Littlestone’s Dimension. However, this reduction breaks down for the approximate setting. In particular, our results do not rule out the fact that an efficient online learner with mistake bound at most, say, twice the optimal exists. This remains an interesting open question. On this front, we remark that the learner used in our reduction is indeed efficient (given in Lemma 21); instead, the computational burden falls to the adversary / nature who has to choose the “hard” input sequence that induces a large biclique.

References

- 1 Piotr Berman and Georg Schnitger. On the complexity of approximating the independent set problem. In *STACS*, pages 256–268, 1989. doi:10.1007/BFb0028990.
- 2 Amey Bhangale, Rajiv Gandhi, Mohammad Taghi Hajiaghayi, Rohit Khandekar, and Guy Kortsarz. Bicovering: Covering edges with two small subsets of vertices. In *ICALP*, pages 6:1–6:12, 2016. doi:10.4230/LIPIcs.ICALP.2016.6.
- 3 Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989. doi:10.1145/76359.76371.
- 4 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-exponential time hypothesis to fixed parameter tractable inapproximability: Clique, dominating set, and more. *SIAM J. Comput.*, 49(4):772–810, 2020. doi:10.1137/18M1166869.
- 5 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 6 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. doi:10.1145/1236457.1236459.
- 7 Irit Dinur. Mildly exponential reduction from gap 3sat to polynomial-gap label-cover. *Electron. Colloquium Comput. Complex.*, page 128, 2016. URL: <https://eccc.weizmann.ac.il/report/2016/128>, arXiv:TR16-128.
- 8 Rodney G. Downey, Patricia A. Evans, and Michael R. Fellows. Parameterized learning complexity. In *COLT*, pages 51–57, 1993. doi:10.1145/168304.168311.
- 9 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.
- 10 Uriel Feige and Shimon Kogan. Hardness of approximation of the balanced complete bipartite subgraph problem. Technical report, Weizmann Institute of Science, Rehovot, Israel, 2004.
- 11 Andreas Emil Feldmann, Karthik C. S., Euiwoong Lee, and Pasin Manurangsi. A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms*, 13(6):146, 2020. doi:10.3390/a13060146.
- 12 Moti Frances and Ami Litman. Optimal mistake bound learning is hard. *Inf. Comput.*, 144(1):66–82, 1998. doi:10.1006/inco.1998.2709.
- 13 Oded Goldreich. On promise problems: A survey. In Oded Goldreich, Arnold L. Rosenberg, and Alan L. Selman, editors, *Theoretical Computer Science, Essays in Memory of Shimon Even*, volume 3895 of *Lecture Notes in Computer Science*, pages 254–290. Springer, 2006. doi:10.1007/11685654_12.

- 14 Steve Hanneke. The optimal sample complexity of PAC learning. *J. Mach. Learn. Res.*, 17:38:1–38:15, 2016. URL: <http://jmlr.org/papers/v17/15-389.html>.
- 15 Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *FOCS*, pages 627–636, 1996. doi:10.1109/SFCS.1996.548522.
- 16 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.
- 17 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.
- 18 Mark Jerrum. Large cliques elude the metropolis process. *Random Struct. Algorithms*, 3(4):347–360, 1992. doi:10.1002/rsa.3240030402.
- 19 Richard Karp. Probabilistic analysis of some combinatorial search problems. *Algorithms and Complexity: New Directions and Recent Results*, 1976.
- 20 Subhash Khot. Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM J. Comput.*, 36(4):1025–1071, 2006. doi:10.1137/S0097539705447037.
- 21 Subhash Khot and Ashok Kumar Ponnuswami. Better inapproximability results for maxclique, chromatic number and min-3lin-deletion. In *ICALP*, pages 226–237, 2006. doi:10.1007/11786986_21.
- 22 Bingkai Lin. The parameterized complexity of k -biclique. In Piotr Indyk, editor, *SODA*, pages 605–615, 2015. doi:10.1137/1.9781611973730.41.
- 23 Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Mach. Learn.*, 2(4):285–318, 1987. doi:10.1007/BF00116827.
- 24 Pasin Manurangsi. Almost-polynomial ratio hardness of approximating densest k-subgraph. In *STOC*, pages 954–961, 2017. doi:10.1145/3055399.3055412.
- 25 Pasin Manurangsi. Inapproximability of maximum edge biclique, maximum balanced biclique and minimum k-cut from the small set expansion hypothesis. In *ICALP*, pages 79:1–79:14, 2017. doi:10.4230/LIPIcs.ICALP.2017.79.
- 26 Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense csps. In *ICALP*, pages 78:1–78:15, 2017. doi:10.4230/LIPIcs.ICALP.2017.78.
- 27 Pasin Manurangsi and Aviad Rubinfeld. Inapproximability of VC dimension and littlestone's dimension. In *COLT*, pages 1432–1460, 2017. URL: <http://proceedings.mlr.press/v65/manurangsi17a.html>.
- 28 Pasin Manurangsi, Aviad Rubinfeld, and Tselil Schramm. The strongish planted clique hypothesis and its consequences. In *ITCS*, pages 10:1–10:21, 2021. doi:10.4230/LIPIcs.ITCS.2021.10.
- 29 Elchanan Mossel and Christopher Umans. On the complexity of approximating the VC dimension. *J. Comput. Syst. Sci.*, 65(4):660–671, 2002. doi:10.1016/S0022-0000(02)00022-3.
- 30 Christos H. Papadimitriou and Mihalis Yannakakis. On limited nondeterminism and the complexity of the V-C dimension. *J. Comput. Syst. Sci.*, 53(2):161–170, 1996. doi:10.1006/jcss.1996.0058.
- 31 Marcus Schaefer. Deciding the Vapnik-Cervonenkis dimension is Σ_3^P -complete. *J. Comput. Syst. Sci.*, 58(1):177–182, 1999. doi:10.1006/jcss.1998.1602.
- 32 Marcus Schaefer. Deciding the k-dimension is pspace-complete. In *CCC*, pages 198–203, 2000. doi:10.1109/CCC.2000.856750.
- 33 V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971. doi:10.1137/1116025.
- 34 David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory Comput.*, 3(1):103–128, 2007. doi:10.4086/toc.2007.v003a006.

A Parameterized Hardness of Approximation

In this section, we briefly discuss the implications of our results for parameterized algorithms. Fixed-parameter algorithms (FPT algorithms) are those that run in $f(k) \cdot n^{O(1)}$ -time where k denote the parameter (specified as part of the input) and, as is standard, n denote the input size⁹. In our case, we say that an algorithm is an α -approximation FPT algorithm for VC Dimension (resp. Littlestone's Dimension) iff it is an FPT algorithm that can decide between $\text{VC}(\mathcal{C}) \geq k$ and $\text{VC}(\mathcal{C}) < k/\alpha$ (resp. $\text{Ldim}(\mathcal{C}) \geq k$ and $\text{Ldim}(\mathcal{C}) < k/\alpha$). We remark that an α -approximation FPT algorithm is only non-trivial iff $\alpha = o(k)$, since there is a trivial $n^{k/\alpha}$ -time α -approximation algorithm. We show that, unfortunately, no non-trivial FPT approximation algorithm exists for VC / Littlestone's Dimensions, as stated below. Previously, W[1]-hardness against *exact* FPT algorithms was known [8], but we are not aware of any result ruling out FPT approximation algorithms (even for small approximation ratio e.g. 1.001).

► **Corollary 30.** *Assuming Gap-ETH, there is no $o(k)$ -approximation FPT algorithm for VC Dimension or Littlestone's Dimension.*

► **Corollary 31.** *Assuming the Strongish Planted Clique Hypothesis, there is no $o(k)$ -approximation FPT algorithm for VC Dimension or Littlestone's Dimension.*

The above corollaries are direct consequences of the following lemma, which shows that any non-trivial FPT approximation algorithm can be used to obtain $o(\log n)$ -approximation in polynomial-time, together with our $o(\log n)$ -factor hardness results from Theorem 3 and Theorem 4.

► **Lemma 32.** *If there is an $o(k)$ -approximation FPT algorithm for VC Dimension (resp. Littlestone's Dimension), then there also exists a polynomial-time $o(\log n)$ -approximation for VC Dimension (resp. Littlestone's Dimension).*

Proof. We only prove the statement for VC Dimension; the proof for Littlestone's Dimension is analogous. Suppose that there exists an $\alpha(k)$ -approximation algorithm \mathbb{A} that runs in $f(k) \cdot n^{O(1)}$ -time for some $\alpha(k) = o(k)$. Let $g : \mathbb{N} \rightarrow \mathbb{N}$ be defined as $g(n) := \min\{\max\{k \in \mathbb{N} \mid f(k) \leq n\}, \lfloor \sqrt{\log n} \rfloor\}$. (If the set is empty, let $g(n) = 0$.) Note that $\lim_{n \rightarrow \infty} g(n) = \infty$, i.e. $g = \omega(1)$. Let us now consider the following algorithm (where \mathcal{C} denote the input concept class):

- Run \mathbb{A} on \mathcal{C} with $k = g(n)$.
- If \mathbb{A} returns YES, then output $k/\alpha(k)$. Otherwise, output 1.

By definition of $g(n)$, the algorithm runs in polynomial time.

To analyze the approximation ratio, consider two cases:

- \mathbb{A} returns YES. Then, we must have $\text{VC}(\mathcal{C}) \geq k/\alpha(k)$ and we output $k/\alpha(k)$. Therefore, the approximation ratio is at most $\frac{\log n}{k/\alpha(k)} \leq o(\log n)$, where the second inequality follows from $k = g(n) = \omega(1)$ and $\alpha(k) = o(k)$.
- \mathbb{A} returns NO. Then, we must have $\text{VC}(\mathcal{C}) \leq k$ and we output 1. The approximation ratio here is at most $k = g(n) \leq \sqrt{\log n}$.

Thus, in both cases, the approximation ratio is $o(\log n)$ as desired. ◀

⁹ For more detail on FPT approximation algorithms and hardness of approximation results, please refer e.g. to the survey [11].

B Proof Sketch of Lemma 25 and Lemma 27

The reduction from the Planted Clique graph $G = (V, E)$ to an input $G' = (A', B', E')$ to Gap Biclique is as follows¹⁰:

- Let A', B' be copies of V .
- Add an edge between $a \in A', b \in B'$ iff $a = b$ or (a, b) belongs to E .

In the YES case where G contains $\lceil N^\delta \rceil$ -clique, then clearly G' also contains $K_{\lceil N^\delta \rceil, \lceil N^\delta \rceil}$. On the other hand, if G is a random $G(N, 1/2)$ graph, a standard union bound argument shows that it does not contain any $K_{\lceil 3 \log N \rceil, \lceil 3 \log N \rceil}$ w.h.p. It is not hard to see (see e.g. [4, Lemma 5.17]) that this implies that G' does not contain $K_{\lceil 6 \log N \rceil, \lceil 6 \log N \rceil}$. This completes the proof sketch.

¹⁰This reduction is quite standard and is also used e.g. in [4].