# Computing Optimal Epsilon-Nets Is as Easy as Finding an Unhit Set

**Nabil H. Mustafa**
Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, ESIEE Paris, France
mustafan@esiee.fr

## Abstract

Given a set system $(X, \mathcal{R})$ with VC-dimension $d$, the celebrated result of Haussler and Welzl (1987) showed that there exists an $\epsilon$-net for $(X, \mathcal{R})$ of size $O\left(\frac{d}{\epsilon}\log\frac{1}{\epsilon}\right)$. Furthermore, the algorithm is simple: just take a uniform random sample from $X$! However, for many geometric set systems this bound is sub-optimal and since then, there has been much work presenting improved bounds and algorithms tailored to specific geometric set systems.

In this paper, we consider the following natural algorithm to compute an $\epsilon$-net: start with an initial random sample $N$. Iteratively, as long as $N$ is not an $\epsilon$-net for $\mathcal{R}$, pick *any* unhit set $S \in \mathcal{R}$ (say, given by an Oracle), and add $O(1)$ randomly chosen points from $S$ to $N$.

We prove that the above algorithm computes, in expectation, $\epsilon$-nets of asymptotically optimal size for all known cases of geometric set systems. Furthermore, it makes $O\left(\frac{1}{\epsilon}\right)$ calls to the Oracle. In particular, this implies that computing optimal-sized $\epsilon$-nets are as easy as computing an unhit set in the given set system.

**2012 ACM Subject Classification** Theory of computation → Sketching and sampling

**Keywords and phrases** $\epsilon$-nets, Geometric Set Systems

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2019.87

**Category** Track A: Algorithms, Complexity and Games

**Funding** This research was supported by the grant ANR SAGA (JCJC-14-CE25-0016-01).

**Acknowledgements** We thank the reviewers for insightful feedback that helped the content and presentation of this work.

## 1 Introduction

Let $X$ be a set of $n$ base elements, and $\mathcal{R}$ a collection of $m$ sets over $X$. Given a parameter $\epsilon > 0$, an $\epsilon$-net for $(X, \mathcal{R})$ is a set $N \subseteq X$ such that for all $S \in \mathcal{R}$ with $|S| \geq \epsilon n$, we have $N \cap S \neq \emptyset$.

The notion of $\epsilon$-nets has been heavily studied across several disciplines, including discrete and computational geometry, machine learning, statistics, convex geometry and randomized algorithms. We refer the reader to the books [3, 7, 14, 16, 20] as well as recent surveys [19, 21].

For general set systems $(X, \mathcal{R})$, it is easy to see that there exists an $\epsilon$-net of size $O\left(\frac{1}{\epsilon}\log|\mathcal{R}|\right)$. For more constrained set systems – for example, those arising in geometric configurations – one can show the existence of $\epsilon$-nets of considerably smaller size. This was realized in the 1980s with the seminal work of Clarkson [8] and Haussler-Welzl [11]. In particular, when the VC-dimension of $(X, \mathcal{R})$, denoted by VC-dim$(\mathcal{R})$, is at most $d$, then there exist $\epsilon$-nets of size $O\left(\frac{d}{\epsilon}\log\frac{1}{\epsilon}\right)$ – of size *independent* of $|X|$ or $|\mathcal{R}|$. Furthermore, to construct a net of this expected size, the algorithm is simple: just take a uniform random sample.

**Haussler-Welzl Net-Finder Algorithm.**

**Input:** base elements $X$, a set system $\mathcal{R}$ on $X$, a parameter $\epsilon > 0$.

$N$: pick each $x \in X$ independently with probability $\Theta\left(\frac{d}{\epsilon|X|}\log\frac{1}{\epsilon}\right)$.

**return** $N$.

It has been observed over the past 30 years that improvements to the Clarkson and Haussler-Welzl bounds are possible for a variety of geometric set systems – e.g., $O(\frac{1}{\epsilon})$-sized nets exist for subsets induced by disks in the plane, half-spaces in $\mathbb{R}^2$ and $\mathbb{R}^3$, subsets induced by pseudo-disks, dual set systems of linear union complexity and so on. More recently, the work of Varadarajan [23], Aronov et al. [4] and Chan et al. [6] has settled the question on sizes of $\epsilon$-nets for many basic geometric set systems precisely in terms of their so-called shallow-cell complexity – in particular, there exist $\epsilon$-nets of size $O\left(\frac{1}{\epsilon}\log\varphi_{\mathcal{R}}\left(O\left(\frac{1}{\epsilon}\right), O(1)\right)\right)$. A set system $(X, \mathcal{R})$ has shallow-cell complexity $\varphi_{\mathcal{R}}(\cdot,\cdot)$ if for any $Y \subseteq X$, the number of subsets in $\mathcal{R}|_Y$ of size at most $l$ is $|Y| \cdot \varphi_{\mathcal{R}}(|Y|, l)$ (here $\mathcal{R}|_Y = \{S \cap Y : S \in \mathcal{R}\}$ is the projection of the set system $\mathcal{R}$ on $Y$). Bounds for $\varphi_{\mathcal{R}}(\cdot,\cdot)$ has been well-studied and by now we know asymptotically optimal bounds for the basic geometric set systems (see [19]).

However, all the above algorithms for constructing $\epsilon$-nets either have efficient implementations but then only work for very specific set systems (e.g., near-linear time algorithm for disks in $\mathbb{R}^2$ [5], half-spaces in $\mathbb{R}^3$ [15]), or are inefficient if they work for general set systems. Consider these algorithms from recent work:

1. Chan et al. [6] construction gives optimal-sized nets as a function of the shallow-cell complexity of the set system; however the algorithm is inefficient. It has to enumerate over each set of $\mathcal{R}$ – there can be $\Omega(n^d)$ such sets for some constant $d$ – to compute a representative of each set of $\mathcal{R}$. Furthermore, it needs to have access to all the sets of $\mathcal{R}$ at the beginning to be able to compute these representatives.
2. Aronov et al. [4] and Varadarajan [23] algorithms can be implemented to work efficiently, but they work for special cases of set systems (so-called dual set systems induced by geometric objects in the plane) and further also need certain spatial decompositions (for the complement of the union) which are specific to the types of geometric objects.
3. Mustafa et al. [18] also give general bounds in terms of the shallow-cell complexity of a set system. However, the algorithm needs to first select a special maximal subset of $\mathcal{R}$ called a packing. This packing can have large size, and furthermore, computing this packing is inefficient, taking $\Omega(n^2)$ time.

## 2 Our Result

Our main insight is that the cause of inefficiency – the careful construction of sets needed for the "alterations" – can be avoided altogether. By extending the ideas present in the work in 1. and 3. above, we present a simple algorithm that

a) computes an $\epsilon$-net of expected size matching the current-best bounds for known geometric set systems, and
b) avoids any pre-computation, hierarchical subdivisions, representation-computation, or partitioning.

Here is our algorithm – as it turns out, a slight addition of the **Haussler-Welzl Net-Finder Algorithm**.

**General Net-Finder Algorithm.**

**Input:** base elements $X$, a set system $\mathcal{R}$ on $X$, a parameter $\epsilon > 0$.

$N$: pick each $x \in X$ independently with probability $p$ to be fixed later (Section 3).

**while** *there exists a set* $S \in \mathcal{R}$, $|S| \geq \epsilon|X|$, *not hit by* $N$ **do**
  update $N$ by adding $O(1)$ uniformly chosen elements of $S$ to $N$.

**return** $N$

**Oracle.** In order to separate the set system-specific implementation details from the net algorithm, we will assume the existence of an Oracle that can return an unhit set in our set system with respect to the current candidate net $N$. We refer the reader to Chazelle [7] for details on oracle-based bounds for sampling in geometric set systems. For geometric set systems, the existence of efficient implementation of such oracles follow from the extensive work on range searching, reporting and emptiness data-structures (see [1]). For example, for the case where $X$ is a set of points in $\mathbb{R}^2$ and the sets are induced by disks, the oracle can be implemented to run in overall time $O(n \cdot \text{polylog}(n))$ using standard techniques (e.g., see [5]).

We refer the reader to Agarwal-Pan [2] for details on data-structures for several other geometric set systems.

Our main theorem:

▶ **Theorem 1.** ***General Net-Finder Algorithm*** *computes an $\epsilon$-net of expected optimal size for $\epsilon$-nets for the following set systems:*

1. $O\left(\frac{1}{\epsilon}\log\varphi_{\mathcal{R}}\left(O\left(\frac{d}{\epsilon}\right), O(d)\right) + \frac{d}{\epsilon}\right)$*: abstract set systems as a function of their shallow-cell complexity* $\varphi_{\mathcal{R}}(\cdot,\cdot)$ *(we will assume that* $\varphi_{\mathcal{R}}(\cdot,\cdot)$ *is non-decreasing in both arguments) and VC-dimension* $d$*,*
2. $O\left(\frac{1}{\epsilon}\right)$*: half-spaces in* $\mathbb{R}^2$*, half-spaces in* $\mathbb{R}^3$*, pseudo-disks and disks in* $\mathbb{R}^2$*, dual systems of linear union complexity,*
3. $O\left(\frac{1}{\epsilon}\log\log\frac{1}{\epsilon}\right)$*: axis-parallel rectangles in* $\mathbb{R}^2$*,*
4. $O\left(\frac{\log(\epsilon\cdot\kappa_{\mathcal{R}}(1/\epsilon))}{\epsilon}\right)$*: dual set systems as a function of their union complexity* $\kappa_{\mathcal{R}}(\cdot)$*,*
5. $O\left(\frac{d}{\epsilon}\log\frac{1}{\epsilon}\right)$*: set systems of VC-dimension at most* $d$*, half-spaces in* $\mathbb{R}^d$*.*

*See [19, Table 47.4.1] for the complete list of known bounds, all of which are produced by our algorithm.*

*Furthermore, it makes expected* $O(\frac{1}{\epsilon})$ *calls to the Oracle.*

▶ Remark 1. For example, all the bounds presented in the work of Varadarajan [23], Clarkson and Varadarajan [10], Pyrga-Ray [22], Chan et al. [6], Aronov et al. [4], Mustafa et al. [18] are achieved by our **General Net-Finder Algorithm**.

▶ Remark 2. We note here that the unhit set $S$ returned by the Oracle at each step need not be random – it can be any unhit set. The expectation is over the choice of the initial random sample as well as the $O(1)$ random points picked from $S$. The specific choice of $S$ is irrelevant.

▶ Remark 3. In particular, Theorem 1 shows that computing an $\epsilon$-net of optimal size is as easy/hard – within a multiplicative factor of $\frac{1}{\epsilon}$ – as computing one set unhit by $N \subseteq X$ in a set system $(X, \mathcal{R})$.

▶ Remark 4. This work is an example of the phenomenon where the "complexity" is moved from the algorithm to its analysis. Thus while our analysis uses specific combinatorial and geometric structures, the algorithm itself becomes very simple and oblivious to these structures (e.g., see [24]).

Lastly, we observe that for the case of set systems with linear-sized $\epsilon$-nets, it is not even necessary to take an initial random sample. The algorithm simplifies even further to:

**Special Net-Finder Algorithm.**

**Input:** base elements $X$, a set system $\mathcal{R}$ on $X$, a parameter $\epsilon > 0$.

$N = \emptyset$.
**while** *there exists a set* $S \in \mathcal{R}$, $|S| \geq \epsilon|X|$, *not hit by* $N$ **do**
  update $N$ by adding $O(1)$ uniformly chosen elements of $S$ to $N$.

**return** $N$.

Our second theorem is the following.

▶ **Theorem 2.** *If the shallow-cell complexity of* $\mathcal{R}$ *satisfies* $\varphi_{\mathcal{R}}(n,k) = O(k^c)$, *then* ***Special Net-Finder Algorithm*** *computes an* $\epsilon$*-net of expected size* $O\left(\frac{1}{\epsilon}\right)$. *In particular, it computes a* $O\left(\frac{1}{\epsilon}\right)$*-sized net for the set systems induced by half-spaces in* $\mathbb{R}^2$, *half-spaces in* $\mathbb{R}^3$, *pseudo-disks and disks in* $\mathbb{R}^2$, *and dual systems of linear union complexity.*

*Furthermore it makes expected* $O(\frac{1}{\epsilon})$ *calls to the Oracle.*

**Organization.** In Section 3 we prove some key structural lemmas about the random process common to both the above algorithms. Then in Section 4 we give the proof of Theorem 1, and in Section 5 the proof of Theorem 2.

## 3 Key Lemmas

We first re-state our main method, **General Net-Finder Algorithm**, more precisely by filling in the exact constant values and probabilities that will be then used in the proofs.

Let $(X, \mathcal{R})$ be the given set system with VC-dim$(\mathcal{R}) \leq d$, and shallow-cell complexity $\varphi_{\mathcal{R}}(\cdot,\cdot)$.

**General Net-Finder Algorithm.**

**Input:** $(X, \mathcal{R})$ with VC-dim$(\mathcal{R}) \leq d$ and shallow-cell complexity $\varphi_{\mathcal{R}}(\cdot,\cdot)$, parameter $\epsilon > 0$.

$\beta, \gamma, c_a$ are absolute constants (explicitly fixed later).

$N_0$: pick $x \in X$ i.i.d. with prob. $c_a \cdot \left( \frac{1}{\left(\frac{3}{4} - \frac{\beta}{2}\right)\epsilon|X|} \log\left( d^3 \varphi_{\mathcal{R}}\left(\frac{4d}{\beta\epsilon}, \frac{24d}{\beta}\right)^2 \right) + \frac{d}{\left(\frac{3}{4} - \frac{\beta}{2}\right)\epsilon|X|} \log \frac{1}{\left(\frac{3}{4} - \frac{\beta}{2}\right)} \right)$

$N = N_0$.
**while** *there exists a set* $S \in \mathcal{R}$, $|S| \geq \epsilon|X|$, *not hit by* $N$ **do**
  $N_S$: pick each $x \in S$ independently with probability $c_a \cdot \left( \frac{1}{\gamma|S|} \log 2 + \frac{d}{\gamma|S|} \log \frac{1}{\gamma} \right)$.
  $N = N \cup N_S$.

**return** $N$.

For the **Special Net-Finder Algorithm**, simply omit the initial random sample, and start with $N = N_0 = \emptyset$.

The proof of our main result builds on the technique in [18]. There the packing lemma was used to construct a maximal packing. The new insight in this paper is that through the use of *two-level* packings, it is not necessary to even know or construct maximal packings (the computational bottleneck earlier). The algorithm is blind to the specific packing; however the analysis amortizes the cost of adding new points to a second-level packing constructed from the sets of the first packing. This is the key new idea, enabling us to bound the total number of points added after the initial random sample.

For the proof of our two main theorems, we will need the following results.

▶ **Theorem A** (Epsilon-net Theorem [11, 12]). *Let $(X, \mathcal{R})$ be a set system, $d \in \mathbb{N}^+$ a positive integer such that* VC-dim$(\mathcal{R}) \leq d$*, and $\epsilon \in [0,1]$ a given real parameter. Then there exists an absolute constant $c_a > 0$ such that a random sample $N$ constructed by picking each point of $X$ independently with probability*

$$c_a \cdot \left( \frac{1}{\epsilon|X|} \log \frac{1}{\gamma} \ + \ \frac{d}{\epsilon|X|} \log \frac{1}{\epsilon} \right).$$

*is an $\epsilon$-net for $\mathcal{R}$ with probability at least $1 - \gamma$.*

Given $(X, \mathcal{R})$, a $(k, \delta)$-packing of $\mathcal{R}$ is a subset $\mathcal{P} \subseteq \mathcal{R}$ such that *i)* for all $S \in \mathcal{P}$ we have $|S| \leq k$, and *ii)* for all $S, S' \in \mathcal{P}$ we have $|\Delta(S, S')| \geq \delta$. Here $\Delta(A, B) = (A \setminus B) \cup (B \setminus A)$ is the symmetric difference of $A$ and $B$.

▶ **Theorem B** (Shallow Packing Lemma [17]). *Let $(X, \mathcal{P})$ be a $(k, \delta)$-packing on $n$ elements, for integers $k, \delta > 0$. If* VC-dim$(\mathcal{P}) \leq d$ *and $\mathcal{P}$ has shallow-cell complexity $\varphi(\cdot, \cdot)$, then $|\mathcal{P}| \leq \frac{24dn}{\delta} \cdot \varphi\big(\frac{4dn}{\delta}, \frac{12dk}{\delta}\big)$.*

In the proof below, we assume that each set $S$ considered by the algorithm has size $[\epsilon n, 2\epsilon n]$. Then we will show that, in expectation, $O(\frac{1}{\epsilon})$ additional points are added after the initial random sample $N_0$. The general case – where the sets $S$ considered by the algorithm can have any size greater than $\epsilon n$ – follows directly: we group the sets considered by the algorithm by their sizes – all sets of size $[2^i \epsilon n, 2^{i+1} \epsilon n]$ go into the same group $i$. So the algorithm can be seen as constructing different nets, a $\epsilon'$-net where $\epsilon' = 2^i \epsilon$, for group $i$, simultaneously. The proof below shows that for each group $i$, the expected number of elements added is $O\left(\frac{1}{\epsilon'}\right) = O\left(\frac{1}{2^i \epsilon}\right)$. Then summing up gives a geometric series, with the overall bound of $O\left(\frac{1}{\epsilon}\right)$ points added over all groups. The initial random sample $N_0$ can be thought of as a different sample for each group, with the total size over all groups again forming a geometric series which sums up to the stated bound.

Let $\beta, \gamma$ be two positive reals whose value will be fixed later, with the property that $\gamma \leq \frac{1}{4}$ and $0 \leq \beta + \gamma \leq 1$.

Fix any *maximal* $(2\epsilon n, \beta \epsilon n)$-packing $\mathcal{P}$ of $\mathcal{R}$ consisting of sets of size at least $\epsilon n$; say the packing consists of the sets

$$\mathcal{P} = \left\{P^1, \ldots, P^m\right\}, \text{ where } m \leq \frac{24dn}{\beta\epsilon n} \varphi_{\mathcal{R}} \left( \frac{4dn}{\beta\epsilon n}, \frac{24d\epsilon n}{\beta\epsilon n} \right) = O\left( \frac{d}{\beta\epsilon} \cdot \varphi_{\mathcal{R}} \left( \frac{4d}{\beta\epsilon}, \frac{24d}{\beta} \right) \right) \text{ (by Theorem B).}$$

Say the **Net-Finder Algorithm** (both general and special) continues for $t$ steps, and adds additional points of $X$ to $N$ for each of the sets $S_1, \ldots, S_t$, namely the points $N_{S_1}, \ldots, N_{S_t}$.

As $\mathcal{P}$ is a maximal $(2\epsilon n, \beta\epsilon n)$-packing of $\mathcal{R}$ and $\epsilon n \leq |S_i| \leq 2\epsilon n$ for each $i = 1, \ldots, t$, it must be that for each $S_i$, there exists an index $j \in [m]$ with $|\Delta(S_i, P^j)| < \beta\epsilon n$ (note that it is possible that $P^j = S_i$). Assign $S_i$ to the set $P^j$ (pick an arbitrary one if there are several such possibilities).

For each $j \in [m]$, let $n_j$ be the number of sets of $\{S_1, \ldots, S_t\}$ assigned to $P^j \in \mathcal{P}$, and denote them by

$$\mathcal{S}^j = \langle S_1^j, \ldots, S_{n_j}^j \rangle, \quad \text{listed } \textit{in the order} \text{ considered by the } \textbf{Net-Finder Algorithm}.$$

Note that $\sum_{j=1}^m n_j = t$, and furthermore,

$$\text{for every } j \in [m],\ \ i \in [n_j], \quad \text{we have} \quad |\Delta(S_i^j, P^j)| < \beta\epsilon n. \tag{1}$$

▷ Claim 3. For each $j \in [m]$ and $i \in [n_j]$, we have

$$|P^j \cap S_i^j| > \frac{|P^j| + |S_i^j| - \beta\epsilon n}{2}.$$

Proof. For each $i \in [n_j]$, we have

$$|P^j| + |S_i^j| = |P^j \setminus S_i^j| + |S_i^j \setminus P^j| + 2|P^j \cap S_i^j| < \beta\epsilon n + 2|P^j \cap S_i^j|, \tag{2}$$

where the last inequality follows from (1). Re-arranging the terms above gives the required statement. ◁

For each $j \in [m]$, define

$$\mathcal{S}'^j = \left\{ S \in \mathcal{S}^j : N_S \text{ turns out to be a } \gamma\text{-net for the set system } (S, \mathcal{R}|_S) \right\}.$$

▷ Claim 4. For any $j \in [m]$,

$$|\mathcal{S}'^j| = \begin{cases} O\left( \frac{d}{\frac{3}{2} - \beta - \gamma} \cdot \varphi_{\mathcal{R}}\left( \frac{4d}{\frac{3}{2} - \beta - \gamma}, \frac{12d}{\frac{3}{2} - \beta - \gamma} \right) \right) & \text{if } \beta + \gamma \geq \frac{1}{2}, \\ O(1) & \text{otherwise.} \end{cases}$$

Proof. Let $n'_j = |\mathcal{S}'^j|$. By re-labeling the sets of $\mathcal{S}^j$, we can assume that $\mathcal{S}'^j = \langle S_1^j, \ldots, S_{n'_j}^j \rangle$, again listed here in the order that they were considered by the **Net-Finder Algorithm**.

Consider the set system $\mathcal{T}'^j = \langle T_1^j, \ldots, T_{n'_j}^j \rangle$ over the base set $P^j$, where $T_i^j = S_i^j \cap P^j$. Consider two distinct indices $k, l \in [n'_j]$ with $k < l$. The set $N_{S_k^j}$ was added to $N$ in the algorithm *before* the set $S_l^j$ was considered. In particular, as $N_{S_k^j}$ is a $\gamma$-net for $\left(S_k^j, \mathcal{R}|_{S_k^j}\right)$ (by the definition of $\mathcal{S}'^j$), it must be that the set $S_l^j$ was not hit by the $\gamma$-net for $\left(S_k^j, \mathcal{R}|_{S_k^j}\right)$ and so $|S_k^j \cap S_l^j| < \gamma \cdot |S_k^j|$. In particular, this implies that

$$|T_k^j \cap T_l^j| = |S_k^j \cap S_l^j \cap P^j| \leq |S_k^j \cap S_l^j| < \gamma \cdot |S_k^j|. \tag{3}$$

Thus we have

$$
\begin{aligned}
|\Delta(T_k^j, T_l^j)| &= |T_k^j| + |T_l^j| - 2|T_k^j \cap T_l^j| \qquad (4)\\
&= |S_k^j \cap P^j| + |S_l^j \cap P^j| - 2|T_k^j \cap T_l^j| \\
&> \frac{|P^j| + |S_k^j| - \beta\epsilon n}{2} + \frac{|P^j| + |S_l^j| - \beta\epsilon n}{2} - 2|T_k^j \cap T_l^j| \quad \text{(by Claim 3)} \\
&> \frac{|P^j| + |S_k^j| - \beta\epsilon n}{2} + \frac{|P^j| + |S_l^j| - \beta\epsilon n}{2} - 2 \cdot \gamma \cdot |S_k^j| \quad \text{(by Inequality (3))} \\
&= |P^j| - \beta\epsilon n + \frac{|S_l^j|}{2} + \left(\frac{1}{2} - 2\gamma\right)|S_k^j| \\
&\geq |P^j| - \beta|P^j| + \frac{|P^j|/2}{2} + \left(\frac{1}{2} - 2\gamma\right)\frac{|P^j|}{2} \quad (\text{as } \epsilon n \leq |P^j|, |S_k^j|, |S_l^j| \leq 2\epsilon n,\ \gamma \leq \frac{1}{4}) \\
&= \left(\frac{3}{2} - \beta - \gamma\right) \cdot |P^j|. \qquad (5)
\end{aligned}
$$

There are two cases to consider here:

- $\beta + \gamma < \frac{1}{2}$. In this case, Inequality (5) implies that $|\Delta(T_k^j, T_l^j)| > |P^j|$, which cannot happen as both $T_k^j$ and $T_l^j$ are subsets of $P^j$. Thus $\mathcal{S}'^j$ must consist of at most one set, and we're done.
- $\beta + \gamma \geq \frac{1}{2}$. In this case, the sets of $\mathcal{T}'^j$ form a $\left(|P^j|, \left(\frac{3}{2} - \beta - \gamma\right) \cdot |P^j|\right)$-packing over the elements of $P^j$. Thus by Theorem B, we have

$$|\mathcal{S}'^j| = |\mathcal{T}'^j| = O\left(\frac{d}{\frac{3}{2} - \beta - \gamma} \cdot \varphi_{\mathcal{R}}\left(\frac{4d}{\frac{3}{2} - \beta - \gamma}, \frac{12d}{\frac{3}{2} - \beta - \gamma}\right)\right). \qquad \triangleleft$$

▶ **Lemma 5.**

$$\mathbf{E}\left[|\mathcal{S}^j|\right] = \begin{cases} O\left(\frac{d}{\frac{3}{2}-\beta-\gamma} \cdot \varphi_{\mathcal{R}}\left(\frac{4d}{\frac{3}{2}-\beta-\gamma}, \frac{12d}{\frac{3}{2}-\beta-\gamma}\right)\right) & \text{if } \beta + \gamma \geq \frac{1}{2}, \\ O(1) & \text{otherwise.} \end{cases}$$

*Further, the above bound holds for any choice of $N_0$.*

**Proof.** We prove this bound for any choice of $N_0$, relying, in the following analysis, only on the sets $N_S$ that were added iteratively. Note that there may be complicated dependencies among the sets of $\mathcal{S}^j$. For example, a set $S_l^j$ may only exist in $\mathcal{S}^j$ because of the choice of the random sample for some earlier set $S_k^j$, $k < l$. However, for a fixed $S \in \mathcal{R}$, the probability of the random sample $N_S$ being a $\gamma$-net for the set system $(S, \mathcal{R}|_S)$ is independent of earlier iterations, and occurs with probability at least $\frac{1}{2}$ by Theorem A. Recalling that $|\mathcal{S}'^j|$ is the number of sets of $\mathcal{S}^j$ for which the random sample succeeds to be a $\gamma$-net, we have

$$\mathbf{E}\left[|\mathcal{S}'^j|\right] = \sum_{S \in \mathcal{S}^j} \Pr\left[N_S \text{ is a } \gamma\text{-net for } (S, \mathcal{R}|_S)\right] \geq \frac{|\mathcal{S}^j|}{2}.$$

On the other hand, Claim 4 upper-bounds $|\mathcal{S}'^j|$ and thus $\mathbf{E}[|\mathcal{S}'^j|]$. Putting them together implies the lemma. ◀

## 4 Proof of Theorem 1

We first give the key theorem from which we will then derive all the bounds promised in Theorem 1. We continue to use the notations and definitions defined earlier.

▶ **Theorem 6.** *Let $(X, \mathcal{R})$ be a set system with shallow-cell complexity $\varphi_{\mathcal{R}}(\cdot,\cdot)$ and VC-dimension at most $d$. Then the* ***General Net-Finder Algorithm*** *returns an $\epsilon$-net of expected size $O\left(\frac{1}{\epsilon}\log\varphi_{\mathcal{R}}\left(O\left(\frac{d}{\epsilon}\right), O\left(d\right)\right) + \frac{d}{\epsilon}\right)$. Furthermore, it makes expected $O\left(\frac{1}{\epsilon}\right)$ calls to the Oracle.*

**Proof.** Clearly the algorithm only stops when $N$ is an $\epsilon$-net. Thus it remains to bound the expected size and the expected running time.

Consider an index $j \in [m]$. By Claim 3, we have that for any $i \in [n_j]$,

$$|P^j \cap S_i^j| > \frac{|P^j| + |S_i^j| - \beta\epsilon n}{2} \geq \frac{|P^j| + |P^j|/2 - \beta|P^j|}{2} = \left(\frac{3}{4} - \frac{\beta}{2}\right) \cdot |P^j|,$$

recalling that $\beta \leq 1$. Thus if $N_0$ is a $\left(\frac{3}{4} - \frac{\beta}{2}\right)$-net for $\left(P^j, \mathcal{R}|_{P^j}\right)$, then any $S \in \mathcal{S}^j$ would be hit by $N_0$ and so it must be that $\mathcal{S}^j = \emptyset$. By Theorem A, for a fixed index $j$, $N_0$ fails to be a $\left(\frac{3}{4} - \frac{\beta}{2}\right)$-net for $\left(P^j, \mathcal{R}|_{P^j}\right)$ with probability $O\left(\frac{1}{d^3\varphi_{\mathcal{R}}\left(\frac{4d}{\beta\epsilon}, \frac{24d}{\beta}\right)^2}\right)$.

At each iteration, for a set $S \in \mathcal{R}$ not hit by $N$, we add

$$\mathbf{E}\left[|N_S|\right] = |S| \cdot c_a \cdot \left(\frac{1}{\gamma|S|}\log 2 \; + \; \frac{d}{\gamma|S|}\log\frac{1}{\gamma}\right) = O\left(\frac{d}{\gamma}\log\frac{1}{\gamma}\right)$$

new points to $N$. Thus the points added to $N$ over all iterations are

$$\mathbf{E}\left[\sum_{i=1}^{t}|N_{S_i}|\right] = O\left(\frac{d}{\gamma}\log\frac{1}{\gamma}\right) \cdot \mathbf{E}[t] = O\left(\frac{d}{\gamma}\log\frac{1}{\gamma}\right) \cdot \mathbf{E}\left[\sum_{j=1}^{m}|\mathcal{S}^j|\right]$$

$$= O\left(\frac{d}{\gamma}\log\frac{1}{\gamma}\right) \cdot \sum_{j=1}^{m}\Pr\left[N_0 \text{ is not a } \left(\frac{3}{4} - \frac{\beta}{2}\right)\text{-net for } (P^j, \mathcal{R}|_{P^j})\right] \cdot$$

$$\mathbf{E}\left[|\mathcal{S}^j| \middle| N_0 \text{ is not a } \left(\frac{3}{4} - \frac{\beta}{2}\right)\text{-net for } (P^j, \mathcal{R}|_{P^j})\right]$$

$$= O\left(\frac{d}{\gamma}\log\frac{1}{\gamma}\right) \cdot m \cdot O\left(\frac{1}{d^3\varphi_{\mathcal{R}}\left(\frac{4d}{\beta\epsilon}, \frac{12d}{\beta/2}\right)^2}\right) \cdot O\left(\frac{d}{\frac{3}{2} - \beta - \gamma} \cdot \varphi_{\mathcal{R}}\left(\frac{4d}{\frac{3}{2} - \beta - \gamma}, \frac{12d}{\frac{3}{2} - \beta - \gamma}\right)\right).$$

As $\frac{3}{2} - \beta - \gamma \geq \frac{1}{2} \geq \max\left\{\beta\epsilon, \frac{\beta}{2}\right\}$ for $\epsilon \leq 0.5$ and by the assumption that $\varphi_{\mathcal{R}}(\cdot,\cdot)$ is non-decreasing in the first and second arguments,

$$\leq O\left(\frac{1}{\gamma\left(\frac{3}{2} - \beta - \gamma\right)}\log\frac{1}{\gamma}\right) \cdot m \cdot O\left(\frac{1}{d\varphi_{\mathcal{R}}\left(\frac{4d}{\beta\epsilon}, \frac{24d}{\beta}\right)}\right)$$

$$= O\left(\frac{1}{\gamma\left(\frac{3}{2} - \beta - \gamma\right)}\log\frac{1}{\gamma}\right) \cdot O\left(\frac{d}{\beta\epsilon} \cdot \varphi_{\mathcal{R}}\left(\frac{4d}{\beta\epsilon}, \frac{24d}{\beta}\right)\right) \cdot O\left(\frac{1}{d\varphi_{\mathcal{R}}\left(\frac{4d}{\beta\epsilon}, \frac{24d}{\beta}\right)}\right)$$

$$= O\left(\frac{1}{\gamma\left(\frac{3}{2} - \beta - \gamma\right)}\log\frac{1}{\gamma}\right) \cdot O\left(\frac{1}{\beta\epsilon}\right).$$

For the initial set $N_0$, we have

$$\mathbf{E}\left[|N_0|\right] = O\left(\frac{1}{\left(\frac{3}{4} - \frac{\beta}{2}\right)\epsilon}\log\left(d\varphi_{\mathcal{R}}\left(\frac{4d}{\beta\epsilon}, \frac{24d}{\beta}\right)\right) + \frac{d}{\left(\frac{3}{4} - \frac{\beta}{2}\right)\epsilon}\log\frac{1}{\left(\frac{3}{4} - \frac{\beta}{2}\right)}\right)$$

Putting everything together, we get

$$\mathbf{E}\left[|N|\right] = \mathbf{E}\left[|N_0|\right] + \mathbf{E}\left[\sum_{i=1}^{t} |N_{S_i}|\right]$$

$$= O\left(\frac{1}{\left(\frac{3}{4}-\frac{\beta}{2}\right)\epsilon}\log\left(d\varphi_{\mathcal{R}}\left(\frac{4d}{\beta\epsilon},\frac{24d}{\beta}\right)\right) + \frac{d}{\left(\frac{3}{4}-\frac{\beta}{2}\right)\epsilon}\log\frac{1}{\left(\frac{3}{4}-\frac{\beta}{2}\right)}\right)$$

$$+ O\left(\frac{1}{\gamma\left(\frac{3}{2}-\beta-\gamma\right)}\log\frac{1}{\gamma}\right)\cdot O\left(\frac{1}{\beta\epsilon}\right).$$

We can set $\gamma$ to be any small-enough constant, say $\gamma = \frac{1}{100}$, and set $\beta = \frac{3}{4}$. Thus we get

$$\mathbf{E}\left[|N|\right] = O\left(\frac{1}{\epsilon}\log\left(d\varphi_{\mathcal{R}}\left(\frac{4d}{\beta\epsilon},\frac{24d}{\beta}\right)\right) + \frac{d}{\epsilon}\right) = O\left(\frac{1}{\epsilon}\log\varphi_{\mathcal{R}}\left(O\left(\frac{d}{\epsilon}\right), O\left(d\right)\right) + \frac{d}{\epsilon}\right). ◀$$

▶ **Theorem 1.** ***General Net-Finder Algorithm*** *computes an $\epsilon$-net of expected optimal size for $\epsilon$-nets for the following set systems:*

1. $O\left(\frac{1}{\epsilon}\log\varphi_{\mathcal{R}}\left(O\left(\frac{d}{\epsilon}\right), O(d)\right) + \frac{d}{\epsilon}\right)$*: abstract set systems as a function of their shallow-cell complexity $\varphi_{\mathcal{R}}(\cdot,\cdot)$ (we will assume that $\varphi_{\mathcal{R}}(\cdot,\cdot)$ is non-decreasing in both arguments) and VC-dimension $d$,*
2. $O\left(\frac{1}{\epsilon}\right)$*: half-spaces in $\mathbb{R}^2$, half-spaces in $\mathbb{R}^3$, pseudo-disks and disks in $\mathbb{R}^2$, dual systems of linear union complexity,*
3. $O\left(\frac{1}{\epsilon}\log\log\frac{1}{\epsilon}\right)$*: axis-parallel rectangles in $\mathbb{R}^2$,*
4. $O\left(\frac{\log(\epsilon\cdot\kappa_{\mathcal{R}}(1/\epsilon))}{\epsilon}\right)$*: dual set systems as a function of their union complexity $\kappa_{\mathcal{R}}(\cdot)$,*
5. $O\left(\frac{d}{\epsilon}\log\frac{1}{\epsilon}\right)$*: set systems of VC-dimension at most $d$, half-spaces in $\mathbb{R}^d$.*

*See [19, Table 47.4.1] for the complete list of known bounds, all of which are produced by our algorithm.*

*Furthermore, it makes expected $O(\frac{1}{\epsilon})$ calls to the Oracle.*

**Proof.** All except one required bound follows directly from Theorem 6; we refer the reader to the survey [19]. Briefly, for the case of halfspaces in $\mathbb{R}^2$, $\mathbb{R}^3$, disks and pseudo-disks in $\mathbb{R}^2$, we have $\varphi(n,k) = O\left(k^c\right)$, for a constant $c$ (by the bound on ($\leq k$)-sets [9]) and so the algorithm returns $\epsilon$-nets of size $O(\frac{1}{\epsilon})$. For the case of half-spaces in $\mathbb{R}^d$ and balls in $\mathbb{R}^{d-1}$, $d \geq 4$, we have $\varphi(n,k) = n^{\lfloor d/2 \rfloor - 1}k^{\lceil d/2 \rceil}$ and so the algorithm returns $\epsilon$-nets of size $O\left(\frac{d}{\epsilon}\log\frac{1}{\epsilon}\right)$. Similarly the bounds follow for the dual set systems as a function of their union complexity.

The exception is the bound of $O\left(\frac{1}{\epsilon}\log\log\frac{1}{\epsilon}\right)$ for the primal system induced by axis-parallel rectangles in the plane. Here we use a result of Aronov et al. [4] which shows that given a set $X$ of $n$ points in the plane, and $\mathcal{R}$ the set system induced on $X$ by all axis-parallel rectangles, there exists another set system $\mathcal{R}'$ on $X$ with the following property:

**1)** For each set $R \in \mathcal{R}$ of size at least $\epsilon n$ induced by an axis-parallel rectangle in the plane, there exists a set $f(R) \in \mathcal{R}'$ also induced by an axis-parallel rectangle in the plane, with $f(R) \subseteq R$ and further $|f(R)| \geq \frac{|R|}{2}$.

**2)** The shallow-cell complexity of $\mathcal{R}'$ is small – $\varphi_{\mathcal{R}'}(n,k) = O\left(\log n \cdot k^3\right)$.

Now **General Net-Finder Algorithm** takes a $O(1)$-sized uniform random sample from a currently unhit set of $\mathcal{R}$, say $R$. But then by property 1) above, it takes a $O(1)/2$-sized uniform random sample from $f(R)$. From property 2) above, we have $\varphi_{\mathcal{R}'}(n,k) = O\left(\log n \cdot k^3\right)$ and so Theorem 6 implies a bound of $O\left(\frac{1}{\epsilon}\log\log\frac{1}{\epsilon}\right)$. ◀

## 5 Proof of Theorem 2

All the known bounds of $O\left(\frac{1}{\epsilon}\right)$ for set systems follow because for such a system $\mathcal{R}$, we have $\varphi_{\mathcal{R}}(n,k) = k^c$ for some constant $c$. Thus Theorem 2 can be deduced from the following theorem.

▶ **Theorem 7.** *Let $(X, \mathcal{R})$ be a set system with shallow-cell complexity $\varphi(n,k) = O(k^c)$ for some absolute constant $c$. Then the* ***Special Net-Finder Algorithm*** *returns an $\epsilon$-net of expected size $O\left(\frac{1}{\epsilon}\right)$.*

**Proof.** Clearly the algorithm only stops when $N$ is an $\epsilon$-net. Thus it remains to bound the expected size and the expected running time.

At each iteration, for a set $S \in \mathcal{R}$ not hit by $N$, we add

$$\mathbf{E}\left[|N_S|\right] = |S| \cdot c_a \cdot \left(\frac{1}{\gamma|S|}\log 2 \;+\; \frac{d}{\gamma|S|}\log\frac{1}{\gamma}\right) = O\left(\frac{d}{\gamma}\log\frac{1}{\gamma}\right)$$

new points to $N$. Thus the points added to $N$ over all iterations are

$$\mathbf{E}\left[\sum_{i=1}^{t}|N_{S_i}|\right] = O\left(\frac{d}{\gamma}\log\frac{1}{\gamma}\right)\cdot\mathbf{E}[t] = O\left(\frac{d}{\gamma}\log\frac{1}{\gamma}\right)\cdot\mathbf{E}\left[\sum_{j=1}^{m}|\mathcal{S}^j|\right]$$
$$= O\left(\frac{d}{\gamma}\log\frac{1}{\gamma}\right)\cdot m\cdot O\left(\frac{d}{\frac{3}{2}-\beta-\gamma}\cdot\varphi_{\mathcal{R}}\left(\frac{4d}{\frac{3}{2}-\beta-\gamma},\frac{12d}{\frac{3}{2}-\beta-\gamma}\right)\right)$$
$$= O\left(\frac{d}{\gamma}\log\frac{1}{\gamma}\right)\cdot O\left(\frac{d}{\beta\epsilon}\cdot\varphi_{\mathcal{R}}\left(\frac{4d}{\beta\epsilon},\frac{24d}{\beta}\right)\right)\cdot O\left(\frac{d}{\frac{3}{2}-\beta-\gamma}\cdot\varphi_{\mathcal{R}}\left(\frac{4d}{\frac{3}{2}-\beta-\gamma},\frac{12d}{\frac{3}{2}-\beta-\gamma}\right)\right)$$

Using $\gamma = \frac{1}{100}$, $\beta = \frac{3}{4}$, $c$ is an absolute constant, and $d = O(c)$ since $\varphi_{\mathcal{R}}(n,k) = O(k^c)$,

$$= O(d)\cdot O\left(\frac{d}{\epsilon}\cdot d^c\right)\cdot O\left(d\cdot d^c\right) = O\left(\frac{1}{\epsilon}\right).$$ ◀

▶ **Theorem 2.** *If the shallow-cell complexity of $\mathcal{R}$ satisfies $\varphi_{\mathcal{R}}(n,k) = O(k^c)$, then* ***Special Net-Finder Algorithm*** *computes an $\epsilon$-net of expected size $O\left(\frac{1}{\epsilon}\right)$. In particular, it computes a $O\left(\frac{1}{\epsilon}\right)$-sized net for the set systems induced by half-spaces in $\mathbb{R}^2$, half-spaces in $\mathbb{R}^3$, pseudo-disks and disks in $\mathbb{R}^2$, and dual systems of linear union complexity.*

*Furthermore it makes expected $O(\frac{1}{\epsilon})$ calls to the Oracle.*

**Proof.** For each of these specific cases, we have $\varphi_{\mathcal{R}}(n,k) = O(k^c)$ for some constant $c$, and thus Theorem 7 implies the bounds. ◀

## 6 Conclusion

Some final remarks:

- The algorithms, as they are stated, need the bound on the shallow-cell complexity (in fact, in the case of rectangles, even a finer decomposition bound) to set the initial sample size. However, by a standard exponential search trick, one can start with an initial guess of $O(1)$ for $N_0$, run the algorithm for $O\left(\frac{1}{\epsilon}\right)$ iterations and if the resulting set is not an $\epsilon$-net, re-run the algorithm with a doubled initial sample size. This incurs an additional $O\left(\log\frac{1}{\epsilon}\right)$ penalty in the running time.

- Our algorithm, unlike earlier work, is adaptive: after the initial random sample, further additional points are added incrementally, and each additional set of $O(1)$ points that are added takes into account the previously added points.
- There has been recent work towards new algorithmic approaches for sketches and samples that work well in practice [13]. We leave for future work the experimental evaluation of our algorithm and comparison with earlier approaches, both in efficiency and whether the adaptive nature of our algorithm leads to improved size bounds in practice.

## References


1. P. K. Agarwal. Range Searching. In J. E. Goodman, J. O'Rourke, and C. D. Tóth, editors, *Handbook of Discrete and Computational Geometry*. CRC Press LLC, 2017.
2. P. K. Agarwal and J. Pan. Near-Linear Algorithms for Geometric Hitting Sets and Set Covers. In *Symposium on Computational Geometry*, page 271, 2014.
3. N. Alon and J. Spencer. *The Probabilistic Method*. Wiley-Interscience, third edition, 2008.
4. B. Aronov, E. Ezra, and M. Sharir. Small-size $\epsilon$-nets for axis-parallel rectangles and boxes. *SIAM Journal on Computing*, 39(7):3248–3282, 2010.
5. N. Bus, S. Garg, N. H. Mustafa, and S. Ray. Tighter estimates for $\epsilon$-nets for disks. *Computational Geometry: Theory and Applications*, 53:27–35, 2016.
6. T. M. Chan, E. Grant, J. Könemann, and M. Sharpe. Weighted Capacitated, Priority, and Geometric Set Cover via Improved Quasi-uniform Sampling. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1576–1585, 2012.
7. B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, 2000.
8. K. L. Clarkson. A Randomized Algorithm for Closest-Point Queries. *SIAM J. Comput.*, 17(4):830–847, 1988.
9. K. L. Clarkson and P. W. Shor. Application of random sampling in computational geometry, II. *Discrete & Computational Geometry*, 4:387–421, 1989.
10. K. L. Clarkson and K. R. Varadarajan. Improved approximation algorithms for geometric set cover. In *Proceedings of the 21st Annual ACM Symposium on Computational Geometry (SoCG)*, pages 135–141, 2005.
11. D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete & Computational Geometry*, 2:127–151, 1987.
12. J. Komlós, J. Pach, and G. Woeginger. Almost tight bounds for $\varepsilon$-nets. *Discrete & Computational Geometry*, 7:163–173, 1992.
13. M. Matheny and J. M. Phillips. Practical Low-Dimensional Halfspace Range Space Sampling. In *26th Annual European Symposium on Algorithms (ESA)*, pages 62:1–62:14, 2018.
14. J. Matoušek. *Lectures on Discrete Geometry*. Springer, 2002.
15. J. Matoušek. Reporting Points in Halfspaces. *Computational Geometry: Theory and Applications*, 2(3):169–186, 1992.
16. J. Matoušek. *Geometric Discrepancy: An Illustrated Guide*. Algorithms and Combinatorics. Springer, 1999.
17. N. H. Mustafa. A Simple Proof of the Shallow Packing Lemma. *Discrete & Computational Geometry*, 55(3):739–743, 2016.
18. N. H. Mustafa, K. Dutta, and A. Ghosh. A Simple Proof of Optimal Epsilon-nets. *Combinatorica*, 38(5):1269–1277, 2018.
19. N. H. Mustafa and K. Varadarajan. Epsilon-approximations and Epsilon-nets. In J. E. Goodman, J. O'Rourke, and C. D. Tóth, editors, *Handbook of Discrete and Computational Geometry*. CRC Press LLC, 2017.
20. J. Pach and P. K. Agarwal. *Combinatorial geometry*. John Wiley & Sons, 1995.
21. J. Phillips. Coresets and Sketches. In J. E. Goodman, J. O'Rourke, and C. D. Tóth, editors, *Handbook of Discrete and Computational Geometry*. CRC Press LLC, 2017.

**22** E. Pyrga and S. Ray. New existence proofs for $\varepsilon$-nets. In *Proceedings of the 24th Annual ACM Symposium on Computational Geometry (SoCG)*, pages 199–207, 2008.

**23** K. R. Varadarajan. Weighted Geometric Set Cover via Quasi-uniform Sampling. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 641–648, 2010.

**24** H. Yu, P. K. Agarwal, R. Poreddy, and K. R. Varadarajan. Practical Methods for Shape Fitting and Kinetic Data Structures using Coresets. *Algorithmica*, 52(3):378–402, 2008.