Bringing Order to Special Cases of Klee’s Measure Problem

Karl Bringmann
Max Planck Institute for Informatics
Saarbrücken, Germany

Abstract
Klee’s Measure Problem (KMP) asks for the volume of the union of \( n \) axis-aligned boxes in \( \mathbb{R}^d \). Omitting logarithmic factors, the best algorithm has runtime \( O^*(n^{d/2}) \) [Overmars,Yap’91]. There are faster algorithms known for several special cases: CUBE-KMP (where all boxes are cubes), UNITCUBE-KMP (where all boxes are cubes of equal side length), HYPERVOLUME (where all boxes share a vertex), and \( k \)-GROUNDED (where the projection onto the first \( k \) dimensions is a HYPERVOLUME instance).

In this paper we bring some order to these special cases by providing reductions among them. In addition to the trivial inclusions, we establish HYPERVOLUME as the easiest of these special cases, and show that the runtimes of UNITCUBE-KMP and CUBE-KMP are polynomially related. More importantly, we show that any algorithm for one of the special cases with runtime \( T(n, d) \) implies an algorithm for the general case with runtime \( T(n, 2d) \), yielding the first non-trivial relation between KMP and its special cases. This allows to transfer \( \text{W}[1] \)-hardness of KMP to all special cases, proving that no \( n^{o(d)} \) algorithm exists for any of the special cases under reasonable complexity theoretic assumptions. Furthermore, assuming that there is no improved algorithm for the general case of KMP (no algorithm with runtime \( O(n^{d/2-\epsilon}) \)) this reduction shows that there is no algorithm with runtime \( O(n^{\lfloor d/2 \rfloor/2-\epsilon}) \) for any of the special cases. Under the same assumption we show a tight lower bound for a recent algorithm for 2-GROUNDED [Yıldız,Suri’12].
1 Introduction

Klee’s measure problem (KMP) asks for the volume of the union of $n$ axis-aligned boxes in $\mathbb{R}^d$, where $d$ is considered to be a constant. This is a classic problem with a long history [2, 8, 9, 13, 15, 16]. The fastest algorithm has runtime $O(n^{d/2} \log n)$ for $d \geq 2$, given by Overmars and Yap [15], which was slightly improved to $n^{d/2} 2^{O(\log^* n)}$ by Chan [8]. Thus, for over twenty years there has been no improvement over the runtime bound $n^{d/2}$. As already expressed in [8], one might conjecture that no improved algorithm for KMP exists, i.e., no algorithm with runtime $O(n^{d/2-\varepsilon})$ for some $\varepsilon > 0$.

However, no matching lower bound is known, not even under reasonable complexity theoretic assumptions. The best unconditional lower bound is $\Omega(n \log n)$ for any dimension $d$ [9]. Moreover, Chan [8] proved that KMP is W[1]-hard by giving a reduction to the $k$-Clique problem. Since his reduction has $k = \Theta(d)$, we can transfer runtime lower bounds from $k$-Clique to KMP, implying that there is no $n^{o(d)}$ algorithm for KMP under reasonable complexity theoretic assumptions (assuming the Exponential Time Hypothesis, see [14]). However, this does not determine the correct constant in the exponent.

Since no further progress was made for KMP for a long time, research turned to the study of special cases. Over the years, the following special cases have been investigated. As we have no space for a detailed listing of the literature on all special cases, we only mention the asymptotically fastest results.

- **Cube-KMP**: Here the given boxes are cubes, not necessarily all with the same side length. This case can be solved in time $O(n^{(d+2)/3})$ for $d \geq 2$ [6]. In dimension $d = 3$ this has been improved to $O(n \log^2 n)$ by Agarwal [1]. In dimensions $d \leq 2$ even the general case can be solved in time $O(n \log n)$, the same bound clearly applies to this special case. As described in [6], there are simple reductions showing that the case of cubes is roughly the same as the case of “$\alpha$-fat boxes”, where all side lengths of a box differ by at most a constant factor $\alpha$.

- **Unitcube-KMP**: Here the given boxes are cubes, all of the same side length. Since the combinatorial complexity of a union of unit cubes is $O(n^{d/2})$ [5], there is an immediate algorithm with runtime $O(n^{d/2} \log n)$ [12], improving upon the best general case algorithm in odd dimensions. Furthermore, there is an elaborate algorithm running in time $O(n^{\lfloor d/2 \rfloor + 1 + \frac{1}{\lceil d/2 \rceil}} \log n)$ [7], improving in even dimensions. Again, there is a generalization to “$\alpha$-fat boxes of roughly equal size”, and any algorithm for Unitcube-KMP can be adapted to an algorithm for this generalization [6].

- **Hypervolume**: Here all boxes have a common vertex. Without loss of generality, we can assume that they share the vertex $(0, \ldots, 0) \in \mathbb{R}^d$ and lie in the positive orthant $\mathbb{R}_{\geq 0}$. This special case is of particular interest for practice, as it is used as an indicator of the quality of a set of points in the field of Evolutionary Multi-Objective Optimization [3, 11, 18, 19]. The only asymptotic improvement known (compared to the general case of KMP) is an algorithm with runtime $O(n \log n)$ for $d = 3$ [4]. The same paper also shows an unconditional lower bound of $\Omega(n \log n)$ for $d > 1$.

- **k-Grounded**: Here the projection of the input boxes to the first $k$ dimensions is a Hypervolume instance, where $0 \leq k \leq d$, the other coordinates are arbitrary. This rather novel special case appeared in [17], where an algorithm with runtime $O(n^{(d-1)/2} \log^2 n)$ for $d \geq 3$ was given for 2-Grounded.

Note that for none of these special cases W[1]-hardness is known, so there is no larger lower bound than $\Omega(n \log n)$, not even under reasonable complexity theoretic assumptions. Also note that there are trivial inclusions of some of these special cases: Each special case can be seen as a subset of all instances of the general case. As such subsets, the following inclusions hold.

- **Unitcube-KMP $\subseteq$ Cube-KMP $\subseteq$ KMP**.
- **$(k + 1)$-Grounded $\subseteq$ k-Grounded** for all $k$.
- **d-Grounded = Hypervolume and 0-Grounded = KMP**.

Clearly, for such an inclusion it holds that any algorithm for the superset is also an algorithm for the subset, with the same runtime. Moreover, any lower bound for the subset is also a lower bound for the superset. This allows to transfer some results listed above to other special cases, e.g., the Cube-KMP algorithm with runtime $O(n^{(d+2)/3})$ also applies to Unitcube-KMP.
1.1 Our results

We present several reductions among the above four special cases and the general case of KMP. They provide bounds on the runtimes needed for these variants and, thus, yield some order among the special cases.

Our first reduction relates HYPERVOLUME and UNITCUBE-KMP.

**Theorem 1.** If there is an algorithm for UNITCUBE-KMP with runtime $T_{\text{UNITCUBE-KMP}}(n, d)$, then there is an algorithm for HYPERVOLUME with runtime

$$T_{\text{HYPERVOLUME}}(n, d) \leq O(T_{\text{UNITCUBE-KMP}}(n, d)).$$

Note that if HYPERVOLUME were a subset of UNITCUBE-KMP, then the same statement would hold, with the constant hidden by the $O$-notation being 1. Hence, this reduction can nearly be seen as an inclusion. Also note that this reduction allows to transfer runtime bounds from UNITCUBE-KMP and CUBE-KMP to HYPERVOLUME, e.g., there is an algorithm for HYPERVOLUME with runtime $O(n^{(d+2)/3})$ for $d \geq 2$. Moreover, together with the trivial inclusions this reduction establishes HYPERVOLUME as the easiest of all studied special cases. In particular, lower bounds for HYPERVOLUME also holds for all other special cases.

**Corollary 2.** For all studied special cases, HYPERVOLUME, UNITCUBE-KMP, CUBE-KMP, and k-GROUNDED (for any $0 \leq k \leq d$), we have the unconditional lower bound $\Omega(n \log n)$ for any $d > 1$.

One can find contradicting statements regarding the feasibility of a reduction as in Theorem 1 in the literature. On the one hand, existence of such a reduction has been mentioned in [17]. On the other hand, a newer paper [10] contains this sentence: “Better bounds have been obtained for the KMP on unit cubes ..., but reducing the hypervolume indicator to such problems is not possible in general.” In any case, a proof of such a statement cannot be found anywhere in the literature.

Our second reduction substantiates the intuition that the special cases CUBE-KMP and UNITCUBE-KMP are very similar, by showing that their runtimes differ by at most a factor of $O(n)$. Recall that UNITCUBE-KMP $\subseteq$ CUBE-KMP was one of the trivial inclusions. Clearly, it implies that if there is an algorithm for CUBE-KMP with runtime $T(n, d)$, then there is an algorithm for UNITCUBE-KMP with the same runtime. We prove an inequality in the other direction in the following theorem.

**Theorem 3.** If there is an algorithm for UNITCUBE-KMP with runtime $T_{\text{UNITCUBE-KMP}}(n, d)$, then there is an algorithm for CUBE-KMP with runtime

$$T_{\text{CUBE-KMP}}(n, d) \leq O(n \cdot T_{\text{UNITCUBE-KMP}}(n, d)).$$

Our third and last reduction finally allows to show lower bounds for all special cases. We show an inequality between the general case of KMP and 2k-GROUNDED, in the opposite direction than the trivial inclusions. For this, we have to increase the dimension in which we consider 2k-GROUNDED.

**Theorem 4.** If there is an algorithm for 2k-GROUNDED in dimension $d+k$ with runtime $T_{2k\text{-GROUNDED}}(n, d+k)$, then there is an algorithm for KMP in dimension $d$ with runtime

$$T_{\text{KMP}}(n, d) \leq O(T_{2k\text{-GROUNDED}}(n, d+k)).$$

Note that, if we set $k = d$, the special case 2k-GROUNDED in $d+k$ dimensions becomes HYPERVOLUME in $2d$ dimensions. Since we established HYPERVOLUME as the easiest variant, the above reduction allows to transfer W[1]-hardness from the general case to all special cases. Since the dimension is increased only by a constant factor, even the tight lower bound on the runtime can be transferred to all special cases.

**Corollary 5.** There is no $n^{o(d)}$ algorithm for any of the special cases HYPERVOLUME, UNITCUBE-KMP, CUBE-KMP, and k-GROUNDED, under the Exponential Time Hypothesis.

We get more precise lower bounds if we assume that no improved algorithm exists for KMP (no algorithm with runtime $O(n^{d/2-\varepsilon})$). Since any algorithm for HYPERVOLUME in dimension $d$ yields an algorithm for KMP in dimension $\lfloor d/2 \rfloor$ by Theorem 4, and since HYPERVOLUME is the easiest special case, we get the following corollary.
Corollary 6. If there is no improved algorithm for KMP, then there is no algorithm with runtime \( O(n^{(d+2)/3}) \) for any of HYPERVOLUME, UNITCUBE-KMP, CUBE-KMP, and k-GROUNDED, for any \( \varepsilon > 0 \).

This shows a lower bound for all studied special cases. Note that there is, however, a wide gap to the best known upper bound of \( O(n^{d+1}) \) for HYPERVOLUME, UNITCUBE-KMP, and CUBE-KMP. Also note that, other than the unconditional lower bound of \( \Omega(n \log n) \) for HYPERVOLUME, there were no lower bounds for any of these special cases before.

Furthermore, setting \( k = 1 \), Theorem 4 immediately implies that the recent algorithm for 2-GROUNDED with runtime \( O(n^{(d-1)/2} \log^2 n) \) \cite{17} is optimal (apart from logarithmic factors and if there is no improved algorithm for KMP).

Corollary 7. If there is no improved algorithm for KMP, then there is no algorithm for 2-GROUNDED with runtime \( O(n^{(d-1)/2 - \varepsilon}) \) for any \( \varepsilon > 0 \).

An alternative way of stating this result would be that any algorithm faster than the one from \cite{17} for 2-GROUNDED would yield an improved algorithm for KMP. To simplify our runtime bounds, in some proofs we use the following technical lemma. Informally, it states that for any \( k \)-GROUNDED algorithm with runtime \( T(n, d) \) we have \( T(O(n), d) \leq O(T(n, d)) \). Intuitively, this should hold, since any reasonable algorithm for this problem has runtime \( T(n, d) = O(n^d) \), i.e., bounded by a polynomial in \( n \), and for polynomial \( T \) this simplification is valid. Note that in this paper we assume \( d \) to be a constant.

Lemma 8. Fix \( 0 \leq k \leq d \) and \( c > 1 \). If there is an algorithm for \( k \)-GROUNDED with runtime \( T_{k, \text{GROUNDED}}(n, d) \) then there is another algorithm for \( k \)-GROUNDED with runtime \( T_{k, \text{GROUNDED}}'(n, d) \) satisfying

\[
T_{k, \text{GROUNDED}}'(cn, d) \leq O(T_{k, \text{GROUNDED}}(n, d)).
\]

Due to space constraints, the proofs of this and other lemmas can be found in the appendix.

1.2 Notation and Organization

A box is a set of the form \( B = [a_1, b_1] \times \ldots \times [a_d, b_d] \subseteq \mathbb{R}^d \), \( a_i, b_i \in \mathbb{R} \), \( a_i \leq b_i \). A cube is a box with all side lengths equal, i.e., \( |b_1 - a_1| = \ldots = |b_d - a_d| \). Moreover, a KMP instance is simply a set \( M \) of \( n \) boxes. In CUBE-KMP all these boxes are cubes, and in UNITCUBE-KMP all these boxes are cubes of common side length. In HYPERVOLUME, all input boxes share the vertex \( (0, \ldots, 0) \subseteq \mathbb{R}^d \), i.e., each input box is of the form \( B = [0, b_1] \times \ldots \times [0, b_d] \). In \( k \)-GROUNDED, the projection of each input box to the first \( k \) dimensions is a HYPERVOLUME box, meaning that each input box is of the form \( B = [a_1, b_1] \times \ldots \times [a_d, b_d] \) with \( a_1 = \ldots = a_k = 0 \).

We use the usual Lebesgue measure of a set \( A \subseteq \mathbb{R}^d \) as \( \text{vol}(A) \). For sets \( R, A \subseteq \mathbb{R}^d \) we write \( \text{vol}_R(A) := \text{vol}(R \cap A) \), the volume of \( A \) restricted to \( R \). For a KMP instance \( M \) we let \( \mathcal{U}(M) := \bigcup_{B \in M} B \). To shorten notation we write \( \text{vol}(M) := \text{vol}(\mathcal{U}(M)) \) and \( \text{vol}_R(M) := \text{vol}(R \cap \mathcal{U}(M)) \).

In the next section we present the proof of Theorem 1. In Section 3 we prove Theorem 3. The proof of Theorem 4 is split into Section 4 and Section 5: We first give the reduction for 2-GROUNDED (again split into the case \( d = 1 \) and a generalization to larger dimensions) and then generalize this result to 2k-GROUNDED, \( k > 1 \). We close with an extensive list of open problems.

2 Hypervolume \( \leq \) Unitcube-KMP

In this section we prove Theorem 1 by giving a reduction from HYPERVOLUME to UNITCUBE-KMP. This shows that the runtime for HYPERVOLUME can be upper bounded by the runtime for UNITCUBE-KMP.

We first describe the intuition, for clarification see Figure 1. Given an instance of HYPERVOLUME, let \( \Delta \) be the largest coordinate of any box. We extend all boxes to cubes of side length \( \Delta \), yielding a UNITCUBE-KMP instance. In this process, we make sure that the new parts of each box will not lie in the positive orthant \( \mathbb{R}_{\geq 0}^d \), but in the other orthants, as depicted in Figure 1. This means that the restriction of each box to \( \mathbb{R}_{\geq 0}^d \) stays the same, implying that the volume of the newly constructed cubes - restricted to \( \mathbb{R}_{\geq 0}^d \) - is the same as
We can slightly simplify this bound, decreasing $n$ to $n - 1$, by the technical Lemma 8. Plugging in the constructed algorithm for Hypervolume with runtime $T_{\text{Hypervolume}}(n, d)$, this lemma guarantees an algorithm for Hypervolume with runtime

$$T'_{\text{Hypervolume}}(n, d) \leq O(T_{\text{Hypervolume}}(n/2, d))$$

$$\leq O(T_{\text{Hypervolume}}(n - 1, d))$$

(1) $$\leq O(T_{\text{Unitcube-KMP}}(n, d)).$$
Figure 2: The left hand side depicts a Cube-KMP instance with smallest cube $C$. The shaded region (with volume $v$) is the set of points contained in $C$ but no other box. The result of our transformation is shown on the right hand side. All boxes now are cubes with $C$’s side length, while inside $C$ nothing changed. Computing the volume of the transformed instance once with and once without $C$ yields $v$.

3 Unitcube-KMP $\geq$ Cube-KMP

In this section we prove Theorem 3 by giving a reduction from Cube-KMP to Unitcube-KMP. This shows that the runtimes of Cube-KMP and Unitcube-KMP are polynomially related.

To get an intuition, assume we are given any Cube-KMP instance. We will show how to compute its volume assuming that we can solve Unitcube-KMP. For this, consider the smallest cube $C$ in the input instance, i.e., the one with smallest side length. We will compute the volume $v$ of space that is contained in $C$ but no other box of the input instance, see Figure 2. Then we recursively compute the volume $w$ of the union of the $n-1$ boxes that are left after deleting $C$. The sum of both, $v + w$, is the volume of the input instance.

To compute $v$, we may modify each cube $B \neq C$ in any manner such that the restriction of $B$ to $C$ stays the same, this does not change $v$. We show that we can modify all other cubes such that their restriction to $C$ remains untouched and they become cubes of $C$’s side length. This is depicted in Figure 2. Applying this construction to all input boxes, we get a Unitcube-KMP instance that, inside $C$, looks the same as the input Cube-KMP instance. Now we compute the volume of this new instance, once with and once without the cube $C$. Subtracting both values, we get exactly $v$. This finishes the reduction. In the following we give the details of this construction.

Let $M = \{B_1, \ldots, B_n\}$ be an instance of Cube-KMP. Sort the cubes $B_1, \ldots, B_n$ descendingly by side length and consider the smallest cube $C := B_n$. We want to compute the volume $v$ of the space that is contained in $C$ but no other box in $M$. With $M_* := M \setminus \{C\}$ we have

$$v = \text{vol}(M) - \text{vol}(M_*),$$

since the extra volume in $\text{vol}(M)$ compared to $\text{vol}(M_*)$ stems from the space that is contained in box $C = B_n$ but no other box $B_i$. Once we know how to compute $v$, there is an immediate recursive algorithm to compute $\text{vol}(M)$:

1. Compute $v$.
2. Recursively compute $\text{vol}(M_*)$.
3. Return $\text{vol}(M) = v + \text{vol}(M_*)$.

In the remainder of this section we describe how to reduce the computation of $v$ to solving Unitcube-KMP instances. Write

$$C = [a_1, b_1] \times \ldots \times [a_d, b_d],$$

and denote by $\Delta = b_1 - a_1 = \ldots = b_d - a_d$ the side length of the cube $C$. Consider an input box $B_i$, $i < n$, that has non-empty intersection with $C$ (all other boxes are unnecessary for computing $v$), with coordinates

$$B_i = [x_1, y_1] \times \ldots \times [x_d, y_d].$$
As \( B_i \) has larger or equal side length compared to \( C \), we cannot have \( a_i < x_i \leq y_i < b_i \) for any \( i \). Instead, for each dimension \( i \), there are the following three possible relations between the intervals \([a_i, b_i]\) and \([x_i, y_i]\).

In each case we define an interval \( B'_i \) of length \( \Delta \) satisfying \( B'_i \cap [a_i, b_i] = [x_i, y_i] \cap [a_i, b_i] \):

- If \( a_i < x_i \leq b_i < y_i \): Set \( B'_i = [x_i, x_i + \Delta] \).
- If \( x_i \leq a_i < b_i < y_i \): Set \( B'_i = [y_i - \Delta, y_i] \).
- If \( x_i \leq a_i \leq b_i \leq y_i \): Set \( B'_i = [a_i, b_i] \).

Then setting \( B' = B'_1 \times \ldots \times B'_d \) we get a cube of side length \( \Delta \) with \( B' \cap C = B \cap C \). This completes the construction of a modified cube \( B' \) that is of \( C \)'s side length and looks the same as \( B \) when restricted to \( C \).

Now we construct two \textsc{Unitcube-KMP} instances \( M' \) and \( M'_* \) as follows. \( M' \) contains the modified version \( B'_i \) of every input cube \( B_i, 1 \leq i \leq n \), that has non-empty intersection with \( C \). In particular, \( M' \) contains \( C \), since \( C' = C \). Moreover, we set \( M'_* = M' \setminus \{C\} \). Now consider

\[
\text{vol}(M') - \text{vol}(M'_*) .
\]

This measures the volume of space contained in \( C \) but no box in \( M'_* \). As the restriction of every box in \( M'_* \) to \( C \) is the same as the restriction of the corresponding box in \( M_* \) to \( C \), we get

\[
\text{vol}(M') - \text{vol}(M'_*) = \text{vol}(M) - \text{vol}(M_*) = v .
\]

Thus, we can indeed compute \( v \) by solving two \textsc{Unitcube-KMP} instances, which completes the reduction.

What is the runtime of the constructed algorithm? Using the above recursive procedure, we reduce the computation of \textsc{Cube-KMP} to \( O(n) \) instances of \textsc{Unitcube-KMP}. After we initially sort the boxes, constructing one of these instances takes time \( O(n) \). Thus, in total we need additional time \( O(n^2) \), yielding a runtime bound of

\[
T_{\textsc{Cube-KMP}}(n, d) \leq O(n \cdot T_{\textsc{Unitcube-KMP}}(n, d)) + O(n^2) .
\]

As any algorithm for \textsc{Unitcube-KMP} at least needs to read its whole input, we have \( T_{\textsc{Unitcube-KMP}}(n, d) \geq \Omega(n) \), which allows to further simplify the above time bound to the desired bound

\[
T_{\textsc{Cube-KMP}}(n, d) \leq O(n \cdot T_{\textsc{Unitcube-KMP}}(n, d)) .
\]

4 \hspace{1cm} 2-Grounded \( \geq \) KMP

We first show the reduction of Theorem 4 for 2-Grounded, i.e., we show \( T_{\text{KMP}}(n, d) \leq O(T_{2\text{-Grounded}}(n, d + 1)) \) by giving a reduction from KMP to 2-Grounded. This already implies Corollary 7 and lays the foundations for the complete reduction given in the next section.

We begin by showing the reduction for \( d = 1 \). As a second step we show how to generalize this to larger dimensions.

4.1 Dimension \( d = 1 \)

We want to give a reduction from KMP in 1 dimension to 2-Grounded in 2 dimensions. Note that the latter is the same as HyperVOLUME in 2 dimensions. Let \( M \) be an instance of KMP in 1 dimension, i.e., a set of \( n \) intervals in \( \mathbb{R} \). We will reduce the computation of \( \text{vol}(M) \) to two instances of 2-Grounded.

Denote by \( x_1 < \ldots < x_m \) the endpoints of all intervals in \( M \) (if all endpoints are distinct then \( m = 2n \)). We can assume that \( x_1 = 0 \) after translation. Consider the boxes

\[
A_i := [m - i - 1, m - i] \times [x_i, x_{i+1}]
\]

in \( \mathbb{R}^2 \) for \( 1 \leq i \leq m - 1 \), as depicted in Figure 3. Denote the union of these boxes by \( A \). Note that the volume of box \( A_i \) is the same as the length of the interval \([x_i, x_{i+1}]\). This means that we took the chain of
Lemma 9. In the above situation we have

\[ \text{vol}_A(C_M) = \text{vol}(A) + \text{vol}(T_0 \cup U(C_M)) - \text{vol}(T_1 \cup U(C_M)). \]
Proof. Set $U := \mathcal{U}(C_M)$. Using $T_0 \subseteq T_1$ and $A = T_1 \setminus T_0$ in a sequence of simple transformations, we get

$$\begin{align*}
\text{vol}(T_1 \cup U) - \text{vol}(T_0 \cup U) &= \text{vol}((T_1 \cup U) \setminus (T_0 \cup U)) \\
&= \text{vol}((T_1 \setminus T_0) \setminus U) \\
&= \text{vol}(A \setminus U) \\
&= \text{vol}(A) - \text{vol}(A \cap U) \\
&= \text{vol}(A) - \text{vol}_A(C_M),
\end{align*}$$

which proves the claim. \hfill \square

Note that $\text{vol}(A) = \sum_i \text{vol}(A_i) = \sum_i |x_{i+1} - x_i| = |x_m - x_1|$ is trivial. Also note that both sets $T_0$ and $T_1$ are the union of $O(n)$ 2-GROUNDED boxes, so that $\text{vol}(T_b \cup \mathcal{U}(C_M))$ can be seen as a 2-GROUNDED instance of size $O(n)$, for both $b \in \{0, 1\}$. Hence, we reduced the computation of the input instance’s volume $\text{vol}(M)$ to $\text{vol}_A(C_M)$ and further to the 2-GROUNDED instances $\text{vol}(T_0 \cup \mathcal{U}(C_M))$ and $\text{vol}(T_1 \cup \mathcal{U}(C_M))$.

As we have to sort the given intervals first, we get $T_{KMP}(n, 1) \leq O(T_{2\text{-GROUNDED}}(O(n), 2) + n \log n)$. Note that this inequality alone gives no new information, as already Klee [13] showed that $T_{KMP}(n, 1) \leq O(n \log n)$. However, we get interesting results when we generalize this reduction to higher dimensions in the next section.

### 4.2 Larger Dimensions

In this section we show that the reduction from the last section easily carries over to larger dimensions, yielding a reduction from KMP in $d$ dimensions to 2-GROUNDED in $d + 1$ dimensions. This implies $T_{KMP}(n, d) \leq O(T_{2\text{-GROUNDED}}(n, d + 1))$. However, here we do not give all details of the proof, since we prove a more general statement in the next section.

Assume we are given a KMP instance $M$ in dimension $d$. The idea is that we use the dimension doubling reduction from the last section on the first dimension and leave all other dimensions untouched. More precisely, for a box $B \in M$ let $\pi_1(B)$ be its projection onto the first dimension and let $\pi_*(B)$ be its projection onto the last $d - 1$ dimensions, so that $B = \pi_1(B) \times \pi_*(B)$. Now follow the reduction from the last section on the instance $M' := \{\pi_1(B) \mid B \in M\}$. This yields sets $A, T_0, T_1$, and a box $C_I$ for each $I \in M'$.

We set $C_B := C_{\pi_1(B)} \times \pi_*(B)$ and $C_M = \{C_B \mid B \in M\}$. A possible way of generalizing $A$ would be to set $A'' := A \times [d-1]$. Then we would be interested in $\text{vol}_{A''}(C_M)$, which can be seen to be exactly $\text{vol}(M)$. This definition of $A''$ is, however, not simple enough, as it is not a difference of 2-GROUNDED instances (unlike $A = T_1 \setminus T_0$). To give a different definition, assume (after translation) that all coordinates of the input instance are nonnegative and let $\Delta$ be the maximal coordinate in any dimension. We set $A' := A \times [0, \Delta]^{d-1}$ and still get the same volume $\text{vol}_{A'}(C_M) = \text{vol}(M)$. This allows to generalize $T_0$ and $T_1$ to $T_0' := T_0 \times [0, \Delta]^{d-1}$ and $T_1' := T_1 \times [0, \Delta]^{d-1}$, while still having

$$\text{vol}_{A'}(C_M) = \text{vol}(A') + \text{vol}(T_0' \cup \mathcal{U}(C_M)) - \text{vol}(T_1' \cup \mathcal{U}(C_M)).$$

Note that $T_0'$ and $T_1'$ are also a union of $O(n)$ 2-GROUNDED boxes, so a volume such as $\text{vol}(T_0' \cup \mathcal{U}(C_M))$ can be seen as a 2-GROUNDED instance. This completes the reduction and yields the time bound $T_{KMP}(n, d) \leq O(T_{2\text{-GROUNDED}}(O(n), d + 1) + n \log n)$. Using the lower bound $\Omega(n \log n)$ of Corollary 2 we can hide the additional $n \log n$ in the first summand.

We may simplify further by the technical Lemma 8: By using the algorithm guaranteed by Lemma 8 with runtime $T_{2\text{-GROUNDED}}(n, d)$ we get a runtime bound of

$$T_{KMP}(n, d) \leq O(T_{2\text{-GROUNDED}}'(O(n), d + 1)) \leq O(T_{2\text{-GROUNDED}}(n, d + 1)),$$

which implies Corollary 7.
5 2k-Grounded $\geq$ KMP

In this section we now prove the full version of Theorem 4, i.e., we give a reduction from KMP in dimension $d$ to 2k-Grounded in dimension $d+k$. For this we enrich the reduction from the last section with an advanced application of the inclusion-exclusion principle.

Let $M$ be a KMP instance. After translation we can assume that in every dimension the minimal coordinate among all boxes in $M$ is 0. Denoting the largest coordinate of any box by $\Delta$ we thus have $B \subseteq [0, \Delta]^d = \Omega^d$ for all $B \in M$, where $\Omega := [0, \Delta]$.

The first steps of generalizing the reduction from the last section to the case $k > 1$ are straightforward. We want to use the dimension doubling reduction from Section 4.1 on each of the first $k$ dimensions. For any box $B \in \mathbb{R}^d$ denote its projection onto the $i$-th dimension by $\pi_i(B)$, $1 \leq i \leq k$, and its projection onto dimensions $k+1, \ldots, d$ by $\pi_+(B)$. We use the reduction from Section 4.1 on each dimension $1 \leq i \leq k$, i.e., on each instance $M(i) := \{\pi_i(B) \mid B \in M\}$, yielding sets $A(i), T(i)_0, T(i)_1$, and a box $C(i)_I$ for each $I \in M(i)$. We assume that all these sets are contained in $[0, \Delta]^d = \Omega^d$, meaning that all coordinates are upper bounded by $\Delta$ (this holds after possibly increasing the $\Delta$ we chose before).

For a box $B \in M$ we now define

$$ C_B := C(i)_1 \times \ldots \times C(k)_i(B) \times \pi_+(B). $$

This is a box in $\mathbb{R}^{d+k}$, it is even a 2k-Grounded box, as its projection onto the first $2k$ coordinates has the vertex $(0, \ldots, 0)$. Set $C_M := \{C_B \mid B \in M\}$. As shown by the following lemma, we want to determine the volume $\text{vol}_A(C_M)$ in

$$ A := A(1) \times \ldots \times A(k) \times [0, \Delta]^{d-k}. $$

Lemma 10. In the above situation we have

$$ \text{vol}(M) = \text{vol}_A(C_M). $$

Unfortunately, the set $A$ is not simply a difference of two 2k-Grounded instances. Thus, the hard part is to reduce the computation of $\text{vol}_A(C_M)$ to 2k-Grounded instances, which we will do in the remainder of this section. For $1 \leq i \leq k$ and $b \in \{0, 1\}$ we set

$$ \tilde{T}_b(i) := \Omega^{2(i-1)} \times T_b(i) \times \Omega^{d+k-2i}. $$

This set in $\Omega^{d+k}$ consists of all points $x$ whose projection to dimensions $2i-1$ and $2i$ is contained in $T_b(i)$. Note that each set $\tilde{T}_b(i)$ can be written as the union of $O(n)$ 2k-Grounded boxes, since $T_b(i)$ is the union of $\ell = O(n)$ 2-Grounded boxes in $\mathbb{R}^2$, i.e., $T_b(i) = \bigcup_{j=1}^{\ell} C_j$, so that we may write $\tilde{T}_b(i) = \bigcup_{j=1}^{\ell} \Omega^{2(i-1)} \times C_j \times \Omega^{d+k-2i}$. Thus, we can use an algorithm for 2k-Grounded to compute any volume of the form $\text{vol}(\tilde{T}_b(i) \cup V)$, where $V$ is a union of $O(n)$ 2k-Grounded boxes.

Furthermore, define for $S \subseteq [k]$

$$ D_S := \left( \bigcup_{i \in S} \tilde{T}_1(i) \right) \cup \bigcup_{i \in [k] \setminus S} \tilde{T}_0(i). $$

Note that $D_S \subseteq D_{S'}$ holds for $S \subseteq S'$. We can express $A$ using the sets $D_S$ as shown by the following lemma.

Lemma 11. In the above situation we have

$$ A = \bigcap_{1 \leq i \leq k} D_{\{i\}} \setminus D_{\emptyset}. $$

Moreover, each $D_S$ can be written as the union of $O(n)$ 2k-Grounded instances, since the same was true for the sets $\tilde{T}_b(i)$. Hence, we can use an algorithm for 2k-Grounded to compute the volume

$$ H_S := \text{vol}(D_S \cup \mathcal{U}(C_M)). $$

Next we show that we can compute $\text{vol}_A(C_M)$ from the $H_S$ by an interesting usage of the inclusion-exclusion principle.
Lemma 12. In the above situation we have
\[ \text{VOL}_A(C_M) = \text{VOL}(A) + \sum_{S \subseteq [k]} (-1)^{|S|} H_S. \]

As \( \text{VOL}(A) = \Delta^{d-k} \cdot \prod_{1 \leq i \leq k} \text{VOL}(A^{(i)}) \) is trivial, we have reduced the computation of \( \text{VOL}_A(C_M) \) to \( 2^k = \mathcal{O}(1) \) instances of \( 2k \)-\textsc{Grounded}, each consisting of \( \mathcal{O}(n) \) boxes. During the construction of these instances we need to sort the coordinates, so that we need additional time \( \mathcal{O}(n \log n) \). This yields
\[ T_{\text{KMP}}(n, d) \leq \mathcal{O}(T_{2k-\text{GROUNDED}}(\mathcal{O}(n), d + k) + n \log n). \]

Because of the lower bound from Corollary 2, we have \( T_{2k-\text{GROUNDED}}(\mathcal{O}(n), d + k) = \Omega(n \log n) \), so we can hide the second summand in the first,
\[ T_{\text{KMP}}(n, d) \leq \mathcal{O}(T_{2k-\text{GROUNDED}}(\mathcal{O}(n), d + k)). \]

We may use the technical Lemma 8 to get rid of the inner \( \mathcal{O} \): This lemma guarantees an algorithm with runtime \( T'_{2k-\text{GROUNDED}}(n, d) \) such that we get
\[ T_{\text{KMP}}(n, d) \leq \mathcal{O}(T'_{2k-\text{GROUNDED}}(\mathcal{O}(n), d + k)) \leq \mathcal{O}(T_{2k-\text{GROUNDED}}(n, d + k)). \]

This finishes the proof.

6 Conclusion

We presented reductions between the special cases \textsc{Cube-KMP}, \textsc{Unitcube-KMP}, \textsc{Hypervolume}, and \( k \)-\textsc{Grounded} of Klee’s measure problem. These reductions imply statements about the runtime needed for these problem variants. We established \textsc{Hypervolume} as the easiest among all studied special cases, and showed that the variants \textsc{Cube-KMP} and \textsc{Unitcube-KMP} have polynomially related runtimes. Moreover, we presented a reduction from the general case of \textsc{KMP} to \( 2k \)-\textsc{Grounded}. This allows to transfer \( \mathsf{W}[1] \)-hardness from \textsc{KMP} to all special cases, proving that no \( n^{o(d)} \) algorithm exists for any of the special cases under reasonable complexity theoretic assumptions. Moreover, assuming that no improved algorithm exists for \textsc{KMP}, we get a tight lower bound for a recent algorithm for \( 2 \)-\textsc{Grounded}, and a lower bound of roughly \( n^{(d-1)/4} \) for all other special cases. Thus, we established some order among the special cases of Klee’s measure problem.

Our results lead to a number of open problems, both asking for new upper and lower bounds:

• Is there a polynomial relation between \textsc{Hypervolume} and \textsc{Unitcube-KMP}, similar to the relation between \textsc{Cube-KMP} and \textsc{Unitcube-KMP}, or do both problems have significantly different runtimes?

• Show that no improved algorithm exists for \textsc{KMP}, e.g., assuming the Strong Exponential Time Hypothesis, as has been done for the Dominating Set problem, see [14]. Or give an improved algorithm.

• Assuming that no improved algorithm for \textsc{KMP} exists, we know that the optimal runtimes of \textsc{Hypervolume} and \textsc{Cube-KMP/Unitcube-KMP} are of the form \( n^{c-d+o(d)} \), with \( c \in [1/4, 1/3] \). Determine the correct value of \( c \).

• Generalize the \( \mathcal{O}(n^{(d-1)/2} \log^2 n) \) algorithm for \( 2 \)-\textsc{Grounded} [17] to an \( \mathcal{O}(n^{(d-k)/2+o(1)}) \) algorithm for \( 2k \)-\textsc{Grounded}. This would again be optimal by Theorem 4.

• We showed the relation \( T_{\text{KMP}}(n, d) \leq \mathcal{O}(T_{2k-\text{GROUNDED}}(n, d + k)) \). Show an inequality in the opposite direction, i.e., a statement of the form \( T_{k-\text{GROUNDED}}(n, d) \leq \mathcal{O}(T_{\text{KMP}}(n, d')) \) with \( d' < d \).
References

[1] P. K. Agarwal. An improved algorithm for computing the volume of the union of cubes. In Proc. 2010 Annual Symposium on Computational Geometry (SoCG’10), pp. 230–239, New York, NY, USA, 2010. ACM.

[2] J. L. Bentley. Algorithms for Klee’s rectangle problems, 1977. Department of Computer Science, Carnegie Mellon University, Unpublished notes.

[3] N. Beume, B. Naujoks, and M. T. M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research, 181:1653–1669, 2007.

[4] N. Beume, C. M. Fonseca, M. López-Ibáñez, L. Paquete, and J. Vahrenhold. On the complexity of computing the hypervolume indicator. IEEE Trans. Evolutionary Computation, 13:1075–1082, 2009.

[5] J.-D. Boissonnat, M. Sharir, B. Tagansky, and M. Yvinec. Voronoi diagrams in higher dimensions under certain polyhedral distance functions. Discrete & Computational Geometry, 19:485–519, 1998.

[6] K. Bringmann. An improved algorithm for Klee’s measure problem on fat boxes. Computational Geometry: Theory and Applications, 45:225–233, 2012.

[7] T. M. Chan. Semi-online maintenance of geometric optima and measures. SIAM J. Comput., 32:700–716, 2003.

[8] T. M. Chan. A (slightly) faster algorithm for Klee’s measure problem. Computational Geometry: Theory and Applications, 43:243–250, 2010.

[9] M. L. Fredman and B. W. Weide. On the complexity of computing the measure of $\bigcup [a_i, b_i]$. Commun. ACM, 21:540–544, 1978.

[10] A. P. Guerreiro, C. M. Fonseca, and M. T. M. Emmerich. A fast dimension-sweep algorithm for the hypervolume indicator in four dimensions. In Proc. 24th Canadian Conference on Computational Geometry (CCCG’12), pp. 77–82, 2012.

[11] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. Evolutionary Computation, 15:1–28, 2007.

[12] H. Kaplan, N. Rubin, M. Sharir, and E. Verbin. Counting colors in boxes. In Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA ’07), pp. 785–794, 2007.

[13] V. Klee. Can the measure of $\bigcup [a_i, b_i]$ be computed in less than $O(n \log n)$ steps? American Mathematical Monthly, 84:284–285, 1977.

[14] D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis, The Complexity Column by V. Arvind. Bulletin of the EATCS, 105:41–72, 2011.

[15] M. H. Overmars and C.-K. Yap. New upper bounds in Klee’s measure problem. SIAM J. Comput., 20:1034–1045, 1991.

[16] J. van Leeuwen and D. Wood. The measure problem for rectangular ranges in $d$-space. J. Algorithms, 2:282–300, 1981.

[17] H. Yildiz and S. Suri. On Klee’s measure problem for grounded boxes. In Proc. 2012 Annual Symposium on Computational Geometry (SoCG’12), pp. 111–120, New York, NY, USA, 2012. ACM.

[18] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In Proc. 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), Vol. 3242 of LNCS, pp. 832–842. Springer, 2004.

[19] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Trans. Evolutionary Computation, 3:257–271, 1999.
A Proof of Lemma 8

Lemma 8. Fix $0 \leq k \leq d$ and $c > 1$. If there is an algorithm for $k$-GROUNDED with runtime $T_{k, \text{GROUNDED}}(n, d)$ then there is another algorithm for $k$-GROUNDED with runtime $T'_{k, \text{GROUNDED}}(n, d)$ satisfying

$$T'_{k, \text{GROUNDED}}(cn, d) \leq O(T_{k, \text{GROUNDED}}(n, d)).$$

Proof. Let $M$ be an instance of $k$-GROUNDED of size $|M| = n$. Denote by $z_1 \leq \ldots \leq z_{2n}$ the coordinates in the first dimension of all boxes in $M$. Let $a_1 := z_{(1-\alpha)n}$ and $b_1 := z_{(1+\alpha)n}$, where $\alpha := 1/(3d)$. Denote by $\{x_1 \leq a_1\}$ the set of all points $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ with $x_1 \leq a_1$, similarly for $\{a_1 \leq x_1 \leq b_1\}$ and $\{x_1 \geq b_1\}$. We consider the three KMP instances

$$M_a := \{B \cap \{x_1 \leq a_1\} \mid B \in M\},$$
$$M_b := \{B \cap \{x_1 \geq b_1\} \mid B \in M\},$$
$$M' := \{B \cap \{a_1 \leq x_1 \leq b_1\} \mid B \in M\}.$$

Note that all three instances can be seen as $k$-GROUNDED instances: Projected onto the first $k$ dimensions, all boxes in $M_a$ share the vertex $(0, \ldots, 0)$, all boxes in $M_b$ share the vertex $(b_1, 0, \ldots, 0)$, and all boxes in $M'$ share the vertex $(a_1, 0, \ldots, 0)$, so after translation they all share the vertex $(0, \ldots, 0)$. Moreover, $\{x_1 < a_1\}$ and $\{x_1 > b_1\}$ contain only $(1-\alpha)n$ coordinates of boxes in $M$ in the first dimension. Hence, there are at most $(1-\alpha)n$ boxes intersecting $\{x_1 < a_1\}$, so after deleting boxes with volume 0 we get $|M_a| \leq (1-\alpha)n$, similarly for $M_b$. This reasoning does not work for $M'$, it might even be that all $n$ boxes are present in $M'$: If a box has left coordinate smaller than $a_1$ and right coordinate larger than $b_1$, then none of its coordinates is seen in $\{a_1 \leq x_1 \leq b_1\}$, although it has non-empty intersection with $\{a_1 \leq x_1 \leq b_1\}$. However, such a box in $M'$ is trivial in the first dimension: its coordinates in the first dimension are simply $[a_1, b_1]$. If all boxes in $M'$ were trivial in the first dimension, then $M'$ would clearly be simpler than the input instance. Although this is not the case, we can bound the number of boxes in $M'$ that are non-trivial in the first dimension: Since there are at most $2\alpha n$ coordinates $z_i$ in $[a_1, b_1]$, all but at most $2\alpha n$ boxes in $M'$ are trivial in the first dimension. Thus, also $M'$ is easier than the input instance $M$, in a certain sense.

Note that $\text{vol}(M) = \text{vol}(M_a) + \text{vol}(M_b) + \text{vol}(M')$ and $M_a, M_b$ are strictly easier than $M$, as they contain at most $(1-\alpha)n$ boxes. We have to simplify $M'$ further. For this, we use the same construction as above (on $M$ and dimension 1) on $M'$ and dimension 2, i.e., we split by coordinates in dimension 2 at $a_2$ and $b_2$. This yields three $k$-GROUNDED instances. Two of them contain at most $(1-\alpha)n$ boxes. The third one, $M''$, may contain up to $n$ boxes. However, all but at most $4\alpha n$ of these boxes are trivial in the first and second dimension, meaning that their projection onto the first 2 dimensions is $[a_1, b_1] \times [a_2, b_2]$.

Iterating this reduction $d$ times yields $2^d$ instances of $k$-GROUNDED containing at most $(1-\alpha)n$ points and one instance $M^*$ that may contain up to $n$ boxes. However, all but at most $2d\alpha n$ of these boxes are trivial in all $d$ dimensions, meaning that they are equal to $[a_1, b_1] \times \ldots \times [a_d, b_d]$. Since all boxes in $M^*$ are contained in $[a_1, b_1] \times \ldots \times [a_d, b_d]$, if any such trivial box exists, the volume of $M^*$ is trivial. Otherwise $M^*$ only contains at most $2d\alpha n = \frac{2}{3}n \leq (1-1/(3d))n = (1-\alpha)n$ boxes. Thus, we have reduced the computation of $\text{vol}(M)$ to $2d + 1$ instances of $k$-GROUNDED with at most $(1-\alpha)n$ boxes each. The reduction itself can be made to run in $O(n)$ time. Hence, if we solve the reduced problems by an algorithm with runtime $T_{k, \text{GROUNDED}}(n, d)$, then we get an algorithm with runtime $T'_{k, \text{GROUNDED}}(n, d)$ satisfying

$$T'_{k, \text{GROUNDED}}(n, d) \leq (2d + 1)T_{k, \text{GROUNDED}}((1-\alpha)n, d) + O(n).$$

As every algorithm for $k$-GROUNDED at least has to read its whole input, we can hide the $O(n)$ by the first term,

$$T'_{k, \text{GROUNDED}}(n, d) \leq O(T_{k, \text{GROUNDED}}((1-\alpha)n, d)),$$

or,

$$T'_{k, \text{GROUNDED}}(n/(1-\alpha), d) \leq O(T_{k, \text{GROUNDED}}(n, d)).$$
Repeating this construction an appropriate number of times we can increase the constant $1/(1 - \alpha)$ to any constant $c > 1$, while the factor on the right hand side is still bounded by a constant. This finally yields an algorithm with runtime satisfying

$$T_{k \text{-GROUNDED}}(cn, d) \leq O(T_{k \text{-GROUNDED}}(n, d)).$$

\[ \square \]

### B  Proof of Lemma 10

**Lemma 10.** In the situation of Section 5 we have

$$\text{VOL}(M) = \text{VOL}_A(C_M).$$

**Proof.** Denote by $x_1^{(i)} < \ldots < x_m^{(i)}$ the coordinates of all boxes in $M$ in the $i$-th dimension. We can express $\text{VOL}(M)$ in terms of the boxes

$$E_{j_1,\ldots,j_d} := [x_{j_1}^{(i)}, x_{j_1+1}^{(i)}] \times \ldots \times [x_{j_d}^{(i)}, x_{j_d+1}^{(i)}],$$

for $1 \leq j_i < m_i$. Since each such box is either completely included in some box in $M$ or does not contribute to $\text{VOL}(M)$, we have

$$\text{VOL}(M) = \sum_{j_1,\ldots,j_d} [E_{j_1,\ldots,j_d} \subseteq \mathcal{U}(M)] \cdot \text{VOL}(E_{j_1,\ldots,j_d}),$$

where $[X]$ is 1 if $X$ is true, and 0 otherwise.

Recall from the reduction in Section 4.1 that $A^{(i)} = A_1^{(i)} \cup \ldots \cup A_{m_i-1}^{(i)}$, and there is a one-to-one correspondence between intervals $[x_j^{(i)}, x_{j+1}^{(i)}]$ and 2-dimensional boxes $A_j^{(i)}$, in particular both have the same volume. This carries over to a one-to-one correspondence between $d$-dimensional boxes $E_{j_1,\ldots,j_d}$ and $(d+k)$-dimensional boxes

$$E'_{j_1,\ldots,j_d} := A_{j_1}^{(1)} \times \ldots \times A_{j_k}^{(k)} \times [x_{j_k+1}^{(k+1)}, x_{j_k+1+1}^{(k+1)}] \times \ldots \times [x_{j_d}^{(d)}, x_{j_d+1}^{(d)}],$$

in particular both have the same volume.

Additionally, recall that an interval $I \in M^{(i)}$ includes $[x_j^{(i)}, x_{j+1}^{(i)}]$ if and only if the 2-dimensional box $C_j^{(i)}$ contains $A_j^{(i)}$. If $I$ does not include $[x_j^{(i)}, x_{j+1}^{(i)}]$, then $\text{VOL}(I \cap [x_j^{(i)}, x_{j+1}^{(i)}]) = 0$, and we also have $\text{VOL}(C_j^{(i)} \cap A_j^{(i)}) = 0$. Hence, we have for any $B \in M$ that $E_{j_1,\ldots,j_d} \subseteq B$ if and only if $E'_{j_1,\ldots,j_d} \subseteq C_B$, which implies that $E_{j_1,\ldots,j_d} \subseteq \mathcal{U}(M)$ if and only if $E'_{j_1,\ldots,j_d} \subseteq \mathcal{U}(C_M)$. Furthermore, $E'_{j_1,\ldots,j_d}$ is either included in $\mathcal{U}(C_M)$ or does not contribute to $\text{VOL}(C_M)$. In total, we get

$$\text{VOL}(M) = \sum_{j_1,\ldots,j_d} [E_{j_1,\ldots,j_d} \subseteq \mathcal{U}(M)] \cdot \text{VOL}(E_{j_1,\ldots,j_d})$$

$$= \sum_{j_1,\ldots,j_d} [E'_{j_1,\ldots,j_d} \subseteq \mathcal{U}(C_M)] \cdot \text{VOL}(E'_{j_1,\ldots,j_d}) = \text{VOL}_A(C_M),$$

which completes the proof. \[ \square \]

### C  Proof of Lemma 11

**Lemma 11.** In the situation of Section 5 we have

$$A = \bigcap_{1 \leq i \leq k} D_i \setminus D_\emptyset.$$
Proof. We have $D_∅ = \bigcup_{i \in [k]} \tilde{T}_0^{(i)}$ and $\tilde{T}_0^{(i)} \subseteq \tilde{T}_1^{(i)}$ for all $i$, implying

$$D_{(i)} = \tilde{T}_1^{(i)} \cup D_∅.$$ 

This yields

$$\bigcap_{1 \leq i \leq k} D_{(i)} \setminus D_∅ = \left( \bigcap_{1 \leq i \leq k} \tilde{T}_1^{(i)} \right) \setminus \bigcup_{i \in [k]} \tilde{T}_0^{(i)}.$$

A point $x = (x_1, \ldots, x_{d+k}) \in \mathbb{R}^{d+k}$ is in the set on the right hand side if and only if it has the following three properties:

- $x_i \in [0, \Delta]$ for all $1 \leq i \leq d + k$,
- $(x_{2i-1}, x_{2i})$ lies in $T_1^{(i)}$ for all $1 \leq i \leq k$,
- $(x_{2i-1}, x_{2i})$ does not lie in $T_0^{(i)}$ for all $1 \leq i \leq k$.

Since $A_{(i)} = T_1^{(i)} \setminus T_0^{(i)}$, this description captures exactly $A^{(1)} \times \ldots \times A^{(k)} \times [0, \Delta]^{d-k} = A$, finishing the proof.

### D Proof of Lemma 12

**Lemma 12.** In the situation of Section 5 we have

$$\text{vol}_{A}(C_M) = \text{vol}(A) + \sum_{S \subseteq [k]} (-1)^{|S|} H_S.$$

**Proof.** In this proof we write for short $U := U(C_M)$. We clearly have

$$\text{vol}(A) - \text{vol}_A(U) = \text{vol}(A \setminus U). \quad (2)$$

We first show

$$\text{vol}(A \setminus U) = \sum_{\emptyset \neq S \subseteq [k]} (-1)^{|S|+1} (H_S - H_∅), \quad (3)$$

and simplify the right hand side later. Using $A = \bigcap_{1 \leq i \leq k} D_{(i)} \setminus D_∅$ (Lemma 11) and the inclusion-exclusion principle we arrive at

$$\text{vol}(A \setminus U) = \text{vol} \left( \bigcap_{1 \leq i \leq k} D_{(i)} \setminus (D_∅ \cup U) \right) = \sum_{\emptyset \neq S \subseteq [k]} (-1)^{|S|+1} \text{vol} \left( \bigcup_{i \in S} D_{(i)} \setminus (D_∅ \cup U) \right).$$

Note that $D_S = \bigcup_{i \in S} D_{(i)}$, so that the above equation simplifies to

$$\text{vol}(A \setminus U) = \sum_{\emptyset \neq S \subseteq [k]} (-1)^{|S|+1} \text{vol} \left( D_S \setminus (D_∅ \cup U) \right), \quad (4)$$

Using the definition of $H_S$ and $D_∅ \subseteq D_S$ for any $S \subseteq [k]$, we get

$$H_S - H_∅ = \text{vol}(D_S \cup U) - \text{vol}(D_∅ \cup U) = \text{vol}((D_S \cup U) \setminus (D_∅ \cup U)) = \text{vol}(D_S \setminus (D_∅ \cup U)).$$
Plugging this into (4) yields (3).

Observe that

$$\sum_{\emptyset \neq S \subseteq \{k\}} (-1)^{|S|+1}(-H_{\emptyset}) = -H_{\emptyset}.$$ 

This allows to further simplify (3) to

$$\text{VOL}(A \setminus U) = \sum_{S \subseteq [k]} (-1)^{|S|+1}H_{S}.$$ 

Plugging this into (2) yields the desired equation. \qed