Top-k Querying of Incomplete Data
under Order Constraints

Antoine Amarilli, Yael Amsterdamer, Tova Milo, and Pierre Senellart

1 Institut Mines-Télécom; Télécom ParisTech; CNRS LTCI
   first.last@telecom-paristech.fr
2 Tel Aviv University
   {yaelamst,milo}@cs.tau.ac.il
3 National University of Singapore; CNRS IPAL

Abstract. Obtaining data values is often expensive, e.g., when they are retrieved from human workers, from remote data repositories, or via complex computations. In this paper, we consider a scenario where the values are taken from a numerical domain, and where some partial order constraints are given over missing and known values. We study the evaluation of top-k selection queries over incomplete data in this setting. Our work is the first to provide a way to estimate the value of the top-k items (rather than just determining the items), doing so through a novel interpolation scheme, and to formally study the computational complexity of the problem. We first formulate the top-k selection problem under possible world semantics. We then present a general solution, whose complexity is polynomial in the number of possible orderings of the data tuples. While this number is unfeasibly large in the worst case, we prove that the problem is hard and thus the dependency is probably unavoidable. We then consider the case of tree-shaped partial orders, and show a polynomial-time, constructive solution.

1 Introduction

Many data analysis tasks make it necessary to evaluate queries over ordered data, such as maximum and top-k queries. In current data analysis processes, such queries must often be evaluated on incomplete data. One important motivation for working with incomplete data is real-life scenarios where retrieving or computing exact data values is expensive, but querying the partial data may still be useful to obtain approximate results, or to decide which data values should be retrieved next. In such situations, one straightforward option is to query only the known, complete part of the data, returning an answer only valid for that part. But, more interestingly, one may be able to query also the unknown part of the data, when certain constraints are known to restrict the missing values. Specifically, we assume the existence of order constraints, i.e., a partial order that relates known and unknown values.

As a motivating application, suppose products need to be classified in a product catalog taxonomy by a crowd of workers [3, 16, 19]. This is required, for instance, when a store redesigns its product catalog, or when we wrap product information from various Web sites into a common catalog. In such an application, for a given product, a number of crowd workers would be asked to assign it to possible categories in the taxonomy, e.g., by answering questions (“Does this product belong to the following categories?”) [3]. It is common to ask multiple workers about the same product and category, and aggregate their answers into an overall compatibility score of the product with the category. However, it
Fig. 1. Sample catalog taxonomy with compatibility scores

may be extremely costly to have workers provide information about all categories for all
products. Consequently, for a given product, one obtains a taxonomy where the scores
of multiple categories are unknown. There has been much research on optimizing the choice
of questions asked to the crowd in such settings [1, 16]. Our goal is complementary – we
wish to optimize the use of the partial information once it has been collected.

Consider the example taxonomy in Fig. 1 (ignore, for now, the numbers on some
categories): it features end categories, marked by a solid border, to which the store
decides that products can be assigned; as opposed to categories with a dashed border,
which the store sees as “virtual” categories containing only other categories. Now,
observe there is a natural ordering between the scores assigned to the categories: if a
product has low compatibility score with the category Cell Phones, it has even lower
compatibility score with the more restrictive category Smartphones. More formally, we
can assume that if a category \( x \) is a sub-category of \( y \), then the product’s compatibility
score for \( x \) is no greater than its score for \( y \). \(^1\)

Suppose our intention is to design the store’s catalog so that each product appears
only under the top-\( k \) categories that it matches best among the end categories. Thus, the
non-end categories will not appear in the query result, though they can give us indirect
clues about the compatibility scores of other categories. One option is to choose the top-\( k \)
categories among those currently scored by enough workers; or one may use metadata,
such as matching likelihood estimations, to provide estimated classifications of products
to categories that the workers have not scored [19]. But these options do not exploit the
ordering of compatibility scores, that can be very helpful in determining the top-\( k \).

Let the numbers in Fig. 1 denote the known compatibility scores, and assume
that we want the top-2 end categories associated to a smartwatch. A system that only
considers categories with known scores would identify Wearable Devices and Diving
Watches as the top-2. A less naïve scheme would notice that the scores of Watches and
Diving Gear, while unknown, cannot be lower than that of Diving Watches, because of
the order constraints; so either of the two could replace Diving Watches in the top-2
answer. To choose which one to pick, we observe that the score of Diving Gear must be
exactly 0.5 (which bounds it from above and below). However, as the score of Wearable
Devices is 0.9, Clothing has score at least 0.9, so the score of Watches can be anywhere
between 0.5 and at least 0.9, making it a better choice. So a better top-2 answer is
Wearable Devices and Watches, replacing Diving Watches with Watches, which is

\(^1\) We assume that all category scores are backed by sufficiently many crowd answers to avoid any
contradictions in the ordering that would be caused by noise.
likely to have a higher score. Furthermore, as we will show, one can even estimate in a principled way the unknown compatibility scores of the top-\(k\) items (here, Watches).

In the present paper, we develop a principled, general approach to estimate the value of the top-\(k\) selected elements over incomplete data with order constraints. Our main contributions are as follows, with proofs and details given in appendix for lack of space.

- **Formal model.** We geometrically characterize the possible valuations of unknown data items as a convex polytope, allowing us to define a uniform probability distribution over the polytope. We then adapt the top-\(k\) query semantics to the uncertain setting, defining it as the computation of the top-\(k\) expected values of unknown data items under the distribution thus defined. (See Section 2.)

- **General solution.** We provide a solution for the general problem, which relies on an exact formulation of the expected value and marginal distribution of the unknown values. However, the computation of the resulting expression is polynomial in the number of possible orderings of the data, which may be unfeasibly large. (See Section 3.)

- **Hardness results.** We then show that this intractability is probably unavoidable, via hardness results. We show that not only computing the expected value of unknown items in our setting is a \#P-complete problem but, moreover, so is computing the top-\(k\) items even without their expected values. (See Section 4.)

- **Tractable cases.** We then consider the case when the order constraints have the form of a tree. We show that this case is tractable by providing a polynomial algorithm to compute the expected values of unknown items. (See Section 5.)

We survey related work in Section 6 and conclude in Section 7.

## 2 Preliminaries

**Incomplete Data under Constraints.** Our input is a set \(\mathcal{X} = \{x_1, \ldots, x_n\}\) of variables with unknown numerical values \(v(x_1), \ldots, v(x_n)\), which we assume to be in the range \([0, 1]\). We consider two kinds of constraints over these variables:

- **order constraints,** written \(x_i \leq x_j\) for \(x_i, x_j \in \mathcal{X}\), encoding that \(v(x_i) \leq v(x_j)\);
- **exact-value constraints,** written \(x_i = \alpha\) for \(0 \leq \alpha \leq 1\) (written as a rational number) and for \(x_i \in \mathcal{X}\), encoding that \(v(x_i) = \alpha\), to represent variables with known values.

In what follows, a **constraint set** with constraints of both types is typically denoted by \(\mathcal{C}\). Note that the explicit constraints within \(\mathcal{C}\) may imply additional constraints, e.g., if \(x = \alpha, y = \beta\) are given, and \(\alpha \leq \beta\), then \(x \leq y\) is implied. We assume that constraints are closed under implication, which can be done in \(\text{PTIME}\) in \(|\mathcal{C}|\) as a transitive closure computation [11] that also considers exact-value constraints.

**Example 1.** Recall the introductory example about classifying a product in a catalog with crowdsourcing. In this setting, the variable \(x_i \in \mathcal{X}\) represents the aggregated compatibility score of the product to the \(i\)-th category. If the score is known, we encode it as a constraint \(x_i = \alpha\); otherwise, it means that we do not know it. In addition, \(\mathcal{C}\) contains the order constraint \(x_j \leq x_i\) whenever category \(j\) is a sub-category of \(i\), since by our definition the score of a sub-category cannot be higher than that of a parent category.

**Possible world semantics.** The incomplete data captured by \(\mathcal{X}\) and \(\mathcal{C}\) makes many valuations of \(\mathcal{X}\) possible, in addition to the true one, \(v(\mathcal{X})\), that we do not know. We model these options via possible world semantics: formally, a possible world \(w\) for a constraint set \(\mathcal{C}\) over \(\mathcal{X} = \{x_1, \ldots, x_n\}\) is a set of values \(w = (v_1, \ldots, v_n) \in [0, 1]^n\).
We write \( d \)-volume we use \( k \) PTIME the dimension of the admissible polytope. 

We can characterize \( \text{pw}(\mathcal{C}) \) geometrically as a polytope, i.e., a finite region of \( n \)-dimensional space enclosed by a finite set of hyperplanes. For instance, in 2-dimensional space, a polytope is a polygon enclosed by line segments. A polytope is called convex if for every two points in the polytope, every point on the line segment connecting them is also within the polytope.

**Observation 1.** Given a set of order and exact-value constraints \( \mathcal{C} \), \( \text{pw}(\mathcal{C}) \) forms a convex polytope within \([0,1]^n\), called the admissible polytope.

Note that when the constraints in \( \mathcal{C} \) are contradictory, the admissible polytope is empty. We can easily show:

**Lemma 1.** Given a set of order and exact-value constraints \( \mathcal{C} \), we can determine in PTIME whether \( \text{pw}(\mathcal{C}) \) is empty.

When exact values are (explicitly or implicitly) enforced for some of the variables or when, e.g., the constraints \( x_i \leq x_j \) and \( x_j \leq x_i \) hold for two variables \( x_i, x_j \in \mathcal{X} \), the dimension of the resulting admissible polytope is less than \(|\mathcal{X}|\). We conventionally define the dimension as \(-1\) if \( \text{pw}_\mathcal{X}(\mathcal{C}) \) is empty; otherwise the dimension can be easily computed from \( \mathcal{C} \), so that we can show:

**Lemma 2.** Given a set of order and exact-value constraints \( \mathcal{C} \), we can compute in PTIME the dimension of the admissible polytope.

**Example 2.** Let \( \mathcal{X} = \{x, y, z\} \). If \( \mathcal{C} = \{x \leq y\} \), the admissible polytope has dimension 3 and is bounded by the planes defined by \( x = y, x = 0, y = 1, z = 0 \) and \( z = 1 \). If we add to \( \mathcal{C} \) the constraint \( y = 0.3 \), the admissible polytope is a 2-dimensional rectangle bounded by \( 0 \leq x \leq 0.3 \) and \( 0 \leq z \leq 1 \) on the \( y = 0.3 \) plane. If we also add \( x = 0.5 \), the admissible polytope is empty.

**Probability distribution.** Next, we quantify the likelihood of the different possible worlds, to be able to evaluate queries over the incomplete data. We assume a uniform probability distribution over \( \text{pw}(\mathcal{C}) \), which captures the case when there is no prior information about the likelihood of the possible worlds.

To define this formally, remember that \( \mathcal{X} \) and \( \mathcal{C} \) define a \( d \)-dimensional convex polytope \( \text{pw}_\mathcal{X}(\mathcal{C}) \) for some integer \( d \). Since the space of possible worlds is continuous, we use \( d \)-volume (also called the Lebesgue measure [13] on \( \mathbb{R}^d \)) to measure the entire set of possible worlds and continuous subsets. This measure coincides with length, area and volume for dimensions 1, 2, and 3, respectively. Then, the uniform probability density function (pdf) \( p : \text{pw}_\mathcal{X}(\mathcal{C}) \to [0,1] \) is simply defined as \( p(w) := \frac{1}{V_d(\mathcal{C})} \), where \( V_d(\mathcal{C}) \) is the \( d \)-volume of the admissible polytope, i.e., the \( \mathbb{R}^d \)-Lebesgue measure of \( \text{pw}_\mathcal{X}(\mathcal{C}) \). We write \( V(\mathcal{C}) \) when \( d \) is taken to be the dimension of \( \text{pw}(\mathcal{C}) \) (\( \mathcal{X} \) being also implicit).

**Top-k selection queries.** The goal of this work is to study the evaluation of top-k queries over incomplete data. Let us first define the query semantics in the case where all variable values are known. First, we apply a selection operator \( \sigma \), which selects a certain subset of the variables \( \mathcal{X}_\sigma \subseteq \mathcal{X} \) (e.g., only the end categories, in the product catalog example). Then, we return the \( k \) variables \( x_{i_1}, \ldots, x_{i_k} \) with the highest values in \( \mathcal{X}_\sigma \), along with
their values, breaking ties arbitrarily. If $|\mathcal{X}_\sigma| \leq k$ we return the entire $\mathcal{X}_\sigma$. Bottom-k queries can be computed in a similar manner.

In the presence of incompleteness, we redefine the semantics of top-k queries to return the $k$ variables with the highest expected values out of $\mathcal{X}_\sigma$, with their expected value. This corresponds to interpolating the unknown values from the known ones, and then querying the result (once again, breaking ties arbitrarily). As we shall see, this simple definition of top-k allows for a foundational study of the evaluation problem, as well as efficient solutions under certain assumptions. Moreover, as mentioned in the Introduction, the interpolation obtained by computing the expected value of variables can be used for more purposes than merely evaluating top-k queries.

Of course, top-k queries on incomplete data can be defined in many alternative ways [5], each yielding different results. See Appendix E for other possible definitions.

**Problem statement.** We summarize the two problems that we study. Given a constraint set $\mathcal{C}$ over variables $\mathcal{X}$, and given a variable $x \in \mathcal{X}$, the interpolation problem for $x$ is to compute the expected value of $x$ in the uniform distribution over $\text{pw}\mathcal{C}$. Given a constraint set $\mathcal{C}$ over $\mathcal{X}$, a selection predicate $\sigma$ and an integer $k$, the top-k computation problem is to compute the ordered list of the $k$ variables of $\mathcal{X}_\sigma$ (or less if $|\mathcal{X}_\sigma| \leq k$) whose expected value is maximal, with ties broken arbitrarily.

### 3 Brute-Force Solution

Having defined the setting, we study how to solve the interpolation and top-k computation problems for order and exact-value constraints. We generalize them by considering the problem of computing the marginal distribution of $x$, namely, the probability distribution over its possible values, derived from the possible world probability distribution.

Formally, let $\mathcal{C}$ be a constraint set, and $x \in \mathcal{X}$ a variable. For a value $v \in [0,1]$, we write $\mathcal{C}_{|x=v}$ the marginalized constraint set $\mathcal{C} \cup \{x=v\}$. Assuming the uniform distribution on the polytope $\text{pw}(\mathcal{C})$, the marginal distribution of $x$ is defined as follows:

**Definition 1.** The marginal distribution of $x$ is the probability density function over $[0,1]$ defined by $p_x : v \mapsto V(\mathcal{C}_{|x=v})/V(\mathcal{C})$. Conventionally, if $\text{pw}(\mathcal{C}_{|x=v}) = 0$, we set $p_x(v) := 0$.

The solution presented next to compute the marginal distributions and expected values is brute-force in the sense that we consider all the possible total orderings of the variables, which is not always tractable because there may be $|\mathcal{X}|!$ orderings in the worst case. But first, in order to consider the possible orderings, we need to deal with the technicality of worlds with tied values.

**Eliminating ties.** We say a possible world $w = (v_1, \ldots, v_n)$ of a constraint set $\mathcal{C}$ has a tie if $v_i = v_j$ for some $i, j$. Intuitively, the only situation where ties have non-zero probability is when they are enforced by $\mathcal{C}$ and hold in every possible world. In such situations, we can rewrite $\mathcal{C}$ by merging variables that have such a persistent tie, to obtain an equivalent constraint set where ties have probability 0. Formally:

**Proposition 1.** For any constraint set $\mathcal{C}$, we can construct in PTIME a constraint set $\mathcal{C}'$ such that the probability that the possible worlds of $\mathcal{C}'$ have a tie (under the uniform distribution) is zero, and such that any interpolation or top-k computation problem on $\mathcal{C}$ can be reduced in PTIME to the same type of problem on $\mathcal{C}'$.
Hence, we assume from now on that ties have zero probability in $C$. We next show how to obtain an expression of the marginal distribution and expected value of a variable, derived from the pdf of possible worlds. First, we show how to obtain an expression for a set $C$ of constraints that enforces a total order over the variables:

**Definition 2.** A constraint set $C$ over variable set $X$ is total if for every two variables $x, y \in X$, one of the order constraints $x \leq y$ or $y \leq x$ holds in $C$. (As we assumed that ties have zero probability, only one of them holds unless $x$ and $y$ are the same variable.)

We then use the expressions for total orders to compute expressions of the expected value and marginal distribution for any constraint set.

### 3.1 Total Order

We first consider the simple case when $C$ is a total order $C_{n,1}^{\alpha, \beta}$ defined as $\alpha \leq x_1 \leq \cdots \leq x_n \leq \beta$, with values $\alpha \geq 0$ and $\beta \leq 1$ (implemented as variables with an exact-value constraint), none of the $x_i$ having an exact-value constraint. For $1 \leq i \leq n$, we want to determine the marginal distribution and expected value of $x_i$ for the uniform distribution in the admissible polytope of $C$. We notice a connection to the existing problem of computing order statistics:

**Definition 3.** The $i$-th order statistic for $n$ samples of a probability distribution $\Pr$ is the distribution $\Pr_i$ of the following process: draw $n$ independent values according to $\Pr$, and return the $i$-th smallest value of the draw.

**Proposition 2 ([8], p. 63).** The $i$-th order statistic for $n$ samples of the uniform distribution on $[0, 1]$ is the Beta distribution $B(i, n+1-i)$.

We next observe the intuitive connection between the $i$-th order statistic and the marginal distribution of the $i$-th smallest variable in a total order.

**Observation 2.** The marginal distribution of $x_i$ within the admissible polytope of $C_{n,1}^{\alpha, \beta}$ is exactly the distribution of the $i$-th order statistic for $n$ samples of the uniform distribution on $[\alpha, \beta]$.

We thus obtain the following formulations: the expected value of $x_i$ in $C_{n,1}^{\alpha, \beta}$ is the mean of the Beta distribution $B(i, n+1-i)$, scaled and shifted to consider draws in $[\alpha, \beta]$, namely: $\alpha + \frac{i(\beta-\alpha)}{n+1}$. This corresponds to a linear interpolation of the unknown variables between $\alpha$ and $\beta$. The pdf of the marginal distribution $p_{x_i} : [\alpha, \beta] \rightarrow [0,1]$ is a polynomial derived from the expression of the Beta distribution. The following proposition summarizes our findings for total order with no exact-value constraints.

**Proposition 3.** Given $C_{n,1}^{\alpha, \beta}$, i.e., a constraint set implying a total order bounded by $\alpha$ and $\beta$, the expected value and the marginal distribution of any variable $x_i$ can be computed in PTIME, and the marginal distribution is a polynomial of degree at most $n$.

### 3.2 Total Order with Exact-Value Constraints

Now, we add exact-value constraints into the total order. This case is also fairly simple since we observe that we can split the total order into sub-sequences in the form of the previous section, and then compute the expected values and marginal distributions of each sub-sequence independently.

The soundness of splitting can be proved via the possible world semantics: assume that there is an exact-value constraint on $x_i$. Then the possible valuations of $x_1, \ldots, x_{i-1}$
We can now extend the result for total orders to an expression of the expected value and the marginal distribution of a variable $x$, i.e.

$$E_x[\mathcal{C}] (\text{resp., } \text{pdf of } x | \mathcal{C})$$

are affected only by the constraints in $\mathcal{C}$ on $x_1, \ldots, x_i$, and similarly for $x_{i+1}, \ldots, x_n$. The possible worlds $\text{pw}(\mathcal{C})$ are then equivalent to the concatenation of every possible valuation of $x_1, \ldots, x_{i-1}, v$ for $x_i$, and every possible valuation of $x_{i+1}, \ldots, x_n$. Formally:

**Lemma 3.** Let $\mathcal{C}$ be a set of exact-value constraints on $\mathcal{X} = \{x_1, \ldots, x_n\}$, and $x_i = v$ an exact-value constraint of $\mathcal{C}$. We write $\mathcal{X}^{\leq_{i-1}} := \{x_1, \ldots, x_{i-1}\}$, $\mathcal{X}^{>_{i+1}} := \{x_{i+1}, \ldots, x_n\}$, and $\mathcal{C}^{\leq_{i-1}}$ (resp., $\mathcal{C}^{>_{i+1}}$) the set of constraints of $\mathcal{C}$ on $\mathcal{X}^{\leq_{i-1}}$ (resp., $\mathcal{X}^{>_{i+1}}$). Then, for any $\alpha < \beta$, $\text{pw}_{\mathcal{X}}(\mathcal{C}^{\leq_{i-1}}(\alpha, v) \cup \mathcal{C})$ can be expressed as the following:

$$\text{pw}_{\mathcal{X}}(\mathcal{C}^{\leq_{i-1}}(\alpha, v) \cup \mathcal{C}) = \text{pw}_{\mathcal{X}}^{\leq_{i-1}}(\mathcal{C}^{\leq_{i-1}}(\alpha, v) \cup \mathcal{C}) \times \text{pw}_{\mathcal{X}}^{>_{i+1}}(\mathcal{C}^{>_{i+1}}(v, \beta) \cup \mathcal{C})$$

Hence, given a constraint set $\mathcal{C}$ imposing a total order and possibly exact-value constraints, the expected value of $x_i$ can be computed as follows. If $x_i$ has an exact-value constraint, its marginal distribution and expected value are trivial. Otherwise, we consider the total order $\mathcal{C}^{\leq_{i-1}}(\alpha, x_i, v)$, where $p$ is the maximal index such that $0 \leq p < i$ and $x_p$ has an exact-value constraint (we see $x_0$ as a virtual variable with exact-value constraint to 0). Similarly, $q$ is the minimal index such that $i < q \leq n + 1$ and $x_q$ has an exact-value constraint, with $x_{q+1} = 1$ a virtual variable. The expected value and marginal distribution of $x_i$ can be computed using the expression of the mean and pdf of Beta distributions.

### 3.3 General Constraint Sets

We can now extend the result for total orders to an expression of the expected value and marginal distribution in the general case, by “splitting” possible worlds according to the order that they impose on variables. To do this, we define the notion of linear extensions, inspired by partial order theory:

**Definition 4.** Given a constraint set $\mathcal{C}$ over variable set $\mathcal{X}$ where ties have a probability of zero, we say that a total constraint set $\mathcal{C}'$ is a linear extension\(^2\) of $\mathcal{C}$ if it is over $\mathcal{X}$, ties have zero probability in $\mathcal{C}'$, the exact-value constraints of $\mathcal{C}'$ are exactly those of $\mathcal{C}$, and any order constraint $x \leq y$ of $\mathcal{C}$ is also in $\mathcal{C}'$.

We now note that we can partition the possible worlds of $\text{pw}(\mathcal{C})$ that have no ties, according to the linear extension of $\mathcal{C}$ which is realized by their values. Worlds with ties can be neglected as having zero probability (as we can assume thanks to Proposition 1).

Following this idea, Algorithm 1 presents a general scheme to compute the expected value of a variable $x \in \mathcal{X}$, given an arbitrary constraint set $\mathcal{C}$, under the uniform distribution.

---

\(^2\) Note that the linear extensions of $\mathcal{C}'$ exactly correspond to the linear extensions of the partial order on $\mathcal{X}$ imposed by the order constraints of $\mathcal{C}$; the assumption that ties have zero probability is necessary to ensure that this partial order is indeed antisymmetric.
distribution on $\text{pw}(\mathcal{C})$. We write $d$ for the dimension of $\text{pw}(\mathcal{C})$. For each linear extension $\mathcal{T}_i$ of $\mathcal{C}$ (we can enumerate them in constant amortized delay [17]), we compute the expected value of $x$ in $\mathcal{T}_i$ and the overall probability of $\mathcal{T}_i$. Since a linear extension is a total order, the expected value of $x$ relative to $\mathcal{T}_i$, denoted by $E_{\mathcal{T}_i}[x]$, can be computed as explained in Section 3.2. As for the probability of $\mathcal{T}_i$, we compute it as the $d$-volume of $\mathcal{T}_i$. More precisely, if $\mathcal{T}_i$ only contains exact-value constraints for the first and last variables (to $\alpha$ and $\beta$ respectively), as in Section 3.1, the volume is $\frac{(\beta - \alpha)^d}{d!}$, namely, the volume of all possible worlds in the value range divided by the number of possible orderings (line 3), as $\mathcal{T}_i$ only realizes one such ordering. Otherwise, following Lemma 3, the volume can be computed as a product of volumes of constraint sets of this form (line 4). Finally, the total $d$-volume of $\text{pw}(\mathcal{C})$ is computed as the sum of volumes of the linear extensions, and the overall expected value $E_{\mathcal{C}}[x]$ is then the sum of expected values for each linear extension, weighted by their probabilities (computed as the $d$-volume of the relevant linear extension divided by the $d$-volume of $\text{pw}(\mathcal{C})$). Hence:

**Theorem 3.** For any constraint set $\mathcal{C}$ over $\mathcal{X}$, for every $x_i \in \mathcal{X}$, the expected value of $x_i$ for the uniform distribution over $\text{pw}(\mathcal{C})$ can be computed in PSPACE and in time $O(\text{poly}(N))$, where $N$ is the number of linear extensions of $\mathcal{C}$.

This implies the following, for top-$k$, by simply computing the expected value of all variables and sorting them:

**Corollary 1.** For any constraint set $\mathcal{C}$ over $\mathcal{X}$, integer $k$ and selection predicate $\sigma$, the top-$k$ query for $\mathcal{C}$ and $\sigma$ can be evaluated in PSPACE and in time $O(\text{poly}(N))$.

Computing the marginal distribution can be done in a similar manner. However, note that this distribution must be defined piecewise: let $v_0 = 0 < v_1 < \cdots < v_m < v_{m+1} = 1$ be the different exact values that occur in $\mathcal{C}$. Each of the marginal distributions computed by the splitting technique is defined for some range $[v_i, v_{i+1}]$. Hence, for each range (“piece”) we sum the expressions of the relevant marginal distributions, again weighted by their probability. The resulting distribution is piecewise polynomial. There are at most $|\mathcal{X}|$ pieces and the degree is at most $|\mathcal{X}| - 1$, and thus each piece of the end function is a polynomial that can be written in PSPACE. The result can be summarized as follows:

**Theorem 4.** For any constraint set $\mathcal{C}$ over $\mathcal{X}$, for any $x_i \in \mathcal{X}$, the marginal distribution of $x_i$ for the uniform distribution over $\text{pw}(\mathcal{C})$ can be expressed as a piecewise polynomial with $\leq |\mathcal{X}|$ pieces and degree $\leq |\mathcal{X}| - 1$, in PSPACE and in time $O(\text{poly}(N))$.

Unfortunately, the method described here makes it necessary to enumerate all possible linear extensions of $\mathcal{C}$, of which there may in general be as many as $|\mathcal{X}|!$. We will accordingly show hardness results in Section 4 and tractable cases in Section 5.

**Center of mass.** We now observe as a sanity check that our definition of interpolation, which computes the expected value of variables under the uniform distribution, coincides with another reasonable global notion of interpolation, namely that of taking the center of mass of the admissible polytope.

**Definition 5.** The center of mass of a polytope for the uniform distribution is the point where all vectors originating at points within the polytope relative to this point sum to zero. In a convex polytope, the center of mass is located within the polytope.

**Proposition 4.** For any constraint set $\mathcal{C}$ on $\mathcal{X}$, if $\Bar{x}_i$ is the expected value of $x_i$ for all $x_i \in \mathcal{X}$, then $\Bar{x}$ is the center of mass of $\text{pw}(\mathcal{C})$. 

8
4 Hardness Results

This section shows the hardness of our top-$k$ and expected value computation problems, drawing from partial order theory. Recall that #P is the class of counting problems that can be expressed as counting the number of accepting paths of a nondeterministic Turing machine that runs in polynomial time.

It is known from [4] that the problem of computing the expected rank of an element in a partial order among its linear extensions is #P-hard. Using this, we can show that our two problems, computing the top-$k$ and computing the expected value, are FP$^{#P}$-complete, namely, they can be computed in polynomial time using a #P-oracle but are hard with respect to this complexity class.\footnote{As the output of our problems are not integers, the problems are not in #P.}

Theorem 5. Given a set $C$ of order constraints with variable set $\mathcal{X}$ and $x \in \mathcal{X}$, determining the expected value of $x$ in $\text{pw}(C)$ under the uniform distribution is FP$^{#P}$-complete.

Theorem 6. The top-$k$ computation problem of computing, given a constraint set $C$ over $\mathcal{X}$, a selection predicate $\sigma$, and an integer $k$, the ordered list of the $k$ items of $\mathcal{X}_\sigma$ that have maximal expected values, is FP$^{#P}$-complete, even if $k$ is fixed to be 1 and the top-$k$ answer does not include the expected value of the variables.

The hardness of the interpolation problem (computing the expected value) is by an immediate connection to expected rank. Showing that the top-$k$ computation problem is hard even without computing the expected values is more subtle. For this, we reduce from expected rank computation by first showing that a top-1 computation oracle can be used to compare the expected value of a variable $x$ to any other value, by assigning the value to a fresh element as an exact-value constraint. Based on this observation, we can reuse results on identifying rational numbers by comparison queries [15] to compute the expected value of $x$ exactly. See the full details, as well as the membership proofs, in Appendix C.

5 Tractable Cases

To work around the hardness results of the previous section, we now study restricted classes of constraint sets for which expected value computation and top-$k$ computation are tractable. As in Section 3, we focus on the problem of computing marginal distributions, because expected values can be computed efficiently from the marginals, and top-$k$ can be computed using the expected values.

Splitting lemma. We start with a generalization of Lemma 3 to decompose constraint sets to simpler constraint sets where expected values can be computed independently.

Definition 6. Let $\mathcal{C}$ be a constraint set on variables $\mathcal{X}$. We say $x_i$ precedes $x_j$, written $x_i \prec x_j$, if we have $x_i \leq x_j$ and there is no $x_k \notin \{x_i, x_j\}$ such that $x_i \leq x_k$ and $x_k \leq x_j$. We say that $x$ influences $x'$, written $x \rightarrow x'$, if there is a path $x_1 \prec \ldots \prec x_n$ with $x_1 = x$, $x_n = x'$, and none of the $x_i$ have exact-value constraints. The influence relation $x \leftrightarrow x'$ is the symmetric, reflexive, and transitive closure of $\rightarrow$. Note that whenever $x$ has an exact-value constraint, then we cannot have $x \leftrightarrow x'$ for any $x' \neq x$.

The uninfluenced classes of $\mathcal{X}$ under $\mathcal{C}$ are, for every class $C$ of the equivalence relation $\leftrightarrow$ consisting of variables with no exact-value constraints, the variables of $C$ plus all variables with an exact-value constraint.
In particular, when \( C \) is a total order, then the uninfluenced classes of \( \mathcal{X} \) under \( C \) contain the variables in any maximal subsequence including no exact-value constraints, as for Lemma 3, plus all variables with an exact-value constraint.

We call the uninfluence partition the set \( C_1, \ldots, C_n \) of constraint sets for the uninfluenced classes \( C_1, \ldots, C_n \) of \( C \), where each \( C_i \) is on the variables of \( C_i \) and comprises all constraints implied by \( C \) about them. We now claim the analogue of Lemma 3:

**Lemma 4.** There exists a bijective correspondence between \( \text{pw}(C_1) \times \cdots \times \text{pw}(C_n) \) and \( \text{pw}(C) \), obtained by merging the variables with exact-value constraints.

We will now study restricted classes of constraint sets, relying on the above lemma to generalize these results to more settings.

**Tree-shaped constraints.** We define the restricted class of constraints that we consider:

**Definition 7.** The Hasse diagram of a partial order \( P = (V, \leq) \) is the DAG on \( P \) with an edge \( (x, y) \) if \( x \prec y \). A constraint set \( \mathcal{C} \) on variable set \( \mathcal{X} \) is tree-shaped if the probability of ties is zero, the Hasse diagram of the partial order induced on \( \mathcal{X} \) by the order constraints is a directed tree, the root has exactly one child and has an exact-value constraint, the leaves have exact-value constraints, and no other node has exact-value constraints. We call \( \mathcal{C} \) reverse-tree-shaped if the reverse of the Hasse diagram (obtained by reversing the direction of the edges) satisfies the requirements of being tree-shaped.

In other words, a tree constraint set specifies a global minimal value, and maximal values at each leaf. We note that our tractability results for tree-shaped constraint sets extend to more general constraint sets (intuitively represented as forests) via Lemma 4.

**Example 3.** Consider the taxonomy of Figure 1 (without the indicated compatibility scores). Remove Wearable Devices and Diving Watches and add dummy variables with exact-value constraints to 0 and 1 as respectively the leaves and root, to materialize the fact that all variables are assumed to be \( \geq 0 \) and \( \leq 1 \). The resulting monotonicity constraints can be expressed as a reverse-tree-shaped constraint set. If we add an exact-value constraint, e.g., for Cell Phones, then the result is no longer reverse-tree-shaped. However, using Lemma 4, we can divide the constraint set into two reverse-tree-shaped constraint sets; one where Cell Phones is a leaf, and one where it is the root.

We now show that for a tree-shaped constraint set, in contrast with the usual setting, the volume of the polytope \( \text{pw}(\mathcal{C}) \), its center of mass, as well as exact expressions of the marginal distributions, can be tractably computed. Note that, in this section, we assume arithmetic operations on rationals to have unit cost, which is reasonable if they are performed up to a fixed numerical precision. If the cost depends on the size of the operands, then the complexities remain polynomial but the degrees may be different.

**Theorem 7.** For any tree-shaped constraint set \( \mathcal{C} \) on variable set \( \mathcal{X} \), we can compute \( V(\mathcal{C}) \) in time \( O(|\mathcal{C}| + |\mathcal{X}|^2) \).

This result can be applied to prove the tractability of computing marginal probabilities in a tree-shaped constraint set:

**Theorem 8.** For any tree-shaped constraint set \( \mathcal{C} \) on variable set \( \mathcal{X} \) and variable \( x \in \mathcal{X} \) with no exact-value constraint, the marginal distribution for \( x \) is piecewise polynomial and can be computed in time \( O(|\mathcal{C}| + |\mathcal{X}|^3) \).
By symmetry, our results about tree-shaped constraint sets extend to reverse-tree-shaped constraint sets. Indeed, a reverse-tree-shaped constraint set $C$ can be transformed into a tree-shaped constraint set $C'$ in which $E_{C'}(x) = 1 - E_C(x)$ for every variable $x$, by reversing order constraints and replacing exact-value constraints $x = \alpha$ with $x = 1 - \alpha$.

Combining the results of this section with Lemma 4, we deduce:

**Corollary 2.** The top-$k$ and expected value computation problems can be solved in PTIME for constraint sets whose uninfluence partition contains only tree-shaped or reverse-tree-shaped constraint sets.

6 Related Work

Existing work has focused on providing semantics and evaluation methods for order queries over uncertain databases, including top-$k$ and ranking queries (e.g., [5, 10, 12, 14, 18, 20, 21]). Such works consider two main uncertainty types: *tuple-level uncertainty*, where the existence of tuples (i.e., variables) is uncertain, and hence affects the query results [5, 12, 14, 20, 21]; and *attribute-level uncertainty*, more relevant to our problem, where the data tuples are known but some of their values are unknown or uncertain [5, 10, 12, 18]. These studies are relevant to ours as they identify multiple possible semantics for order queries in presence of uncertainty, and specify desired properties for such semantics [5, 12]; our definition of top-$k$ satisfies the desiderata that are relevant to attribute-level uncertainty [12]. The relevance of partial order constraints was already considered in [18].

However, *none of these studies considers the interpolation problem*, or the estimation of unknown values; instead, they focus on ranks. Further, those papers that study probabilistic settings assume, for simplicity, that *independent* marginal distributions are known for the missing values [5, 12, 18]. Therefore they do not capture the mutual effect of dependent unknown values on their marginal distributions, as we do. We also mention our previous work [2] which considers estimation of uncertain values (their expectation and variance) in a more general setting. However, the initial results presented in [2] were for *approximate* interpolation of expected value and variance in a *total order*. These are superseded by our present results for computing *exactly* the marginal distributions in *general partial orders*, enabled by the possible world semantics considered here. In addition, top-$k$ querying and complexity issues were not studied in [2].

Another relevant research topic is *partial order search*: querying the elements of a partially ordered set to find a subset of elements with a certain property [1, 7, 9, 16]. This relates to many applications, e.g., crowd-assisted graph search [16], frequent itemset mining with the crowd [1], and knowledge discovery, where the unknown data is queried via oracle calls [9]. These studies are *complementary to ours*: if such search processes are stopped before all values are known, our method can be used to estimate the output (assuming that the target function can be phrased as a top-$k$ or interpolation problem).

While various methods of interpolation have been considered for totally ordered data, we are the first, to our knowledge, to propose a principled interpolation scheme for partially ordered data. The work of [6] has considered a non-linear interpolation method based on spline curves (highly-smooth, piecewise polynomial functions) for partial orders. However, its focus is mostly on providing a parameterized formulation for interpolation, not motivated by possible world semantics or geometric interpretations.
7 Conclusion

We have studied top-\(k\) query evaluation and interpolation for data with incomplete values and partial order constraints, and provided foundational solutions, including a brute-force computation scheme, hardness results, and analysis of tractable cases.

We note that the uniform interpolation scheme that we have described, while intuitive, fails to respect a natural stability property. Intuitively, an interpolation scheme is stable if adding exact-value constraints to fix some variables to their interpolated values does not change the interpolated values of the other variables. We can show that this property is not respected by our scheme (see Appendix F). We leave for future work the study of alternative interpolation schemes on partial orders that would be stable in this sense.

References

A Proofs for Section 2 (Preliminaries)

Observation 1. Given a set of order and exact-value constraints \( \mathcal{C} \), \( pw(\mathcal{C}) \) forms a convex polytope within \([0, 1]^n\), called the admissible polytope.

Proof. Both order and exact-value constraints can be encoded as linear constraints, i.e., a set of inequalities of linear expressions on \( \mathcal{X} \) with constants in \([0, 1]\). \( x_i \leq x_j \) can be encoded as \( x_j - x_i \geq 0 \) and \( x_i = \alpha \) can be encoded as \( x_i \geq \alpha \) and \( x_i \leq \alpha \). As usual in linear programming, every linear constraint defines a half-space of the entire space of valuations to the variables of \( \mathcal{X} \). The set of possible worlds is then the convex polytope obtained by intersecting \([0, 1]^n\) with the half-spaces of the constraints. Since each half-space is convex (and \([0, 1]^n\) is as well), and since the intersection of convex sets is also convex, \( pw(\mathcal{C}) \) is convex.

Lemma 1. Given a set of order and exact-value constraints \( \mathcal{C} \), we can determine in PTIME whether \( pw(\mathcal{C}) \) is empty.

Proof. Denote by \( G \) the directed graph on \( \mathcal{X} \) defined by the order constraints of \( \mathcal{C} \). We compute in PTIME the directed acyclic graph \( G' \) of the strongly connected components of \( G \).

If \( G \) contains two vertices \( x \) and \( y \) in the same connected component that have exact-value constraints to two values \( v_x \neq v_y \), we claim that the admissible polytope is empty. Indeed, assuming without loss of generality that \( v_x < v_y \), considering the sequence of comparability relations \( y = x_1 \leq x_2 \leq \cdots \leq x_n = x \) (either in \( \leq \mathcal{C} \) or implied by the exact-value constraints) that was used to justify the existence of a path from \( y \) to \( x \), any possible world in the admissible polytope would have to satisfy these order constraints, which is impossible as \( v_x < v_y \) and the value of \( x \) and \( y \) in the possible world must be \( v_x \) and \( v_y \), respectively.

If there are no two such vertices, we set the value of each strongly connected component in \( G' \) to be undefined if none of its nodes has an exact-value constraint, or to be the value of the exact-value constraint of its nodes, if any. (By the previous paragraph, there is at most one such value.) If there are two strongly connected components \( s, s' \) such that \( s' \) is reachable from \( s \) in \( G' \) but the value of \( s \) in \( G' \) is strictly greater than the value of \( s' \), we claim that the admissible polytope is empty. Indeed, considering a path of comparability relations from the node of \( s \) with an exact-value constraint to the node of \( s' \) with an exact-value constraint, if we assume the existence of a possible world, we reach a contradiction like before.

Otherwise, we claim that the admissible polytope is non-empty. This is easily checked by completing the values assigned to nodes in \( G' \) so that all nodes have a value and no order constraint is violated, and extending this to a possible world of \( \mathcal{C} \). \( \square \)

Lemma 2. Given a set of order and exact-value constraints \( \mathcal{C} \), we can compute in PTIME the dimension of the admissible polytope.

Proof. We can compute this dimension of \( pw_{\mathcal{X}}(\mathcal{C}) \) by viewing the set of constraints as a graph. First, consider \( \mathcal{C} \), the version of \( \mathcal{C} \) extended with \( 0 \leq x \leq 1 \) constraints for all \( x \in \mathcal{X} \). Second, build a directed graph \( G_{\mathcal{C},\mathcal{X}} \) of all variables in \( \mathcal{X} \) and all constant
values within $\widetilde{G}$, with an edge between $u$ and $v$ if $u \preceq v$ holds in $\mathcal{G}$, for any variables or constant values $u$ and $v$. The dimension of $\text{pw}(\mathcal{G})$ is conventionally -1 if $G_{\mathcal{G}, \mathcal{X}}$ has a cycle containing two different constant values; otherwise, it is the number of strongly connected components of $G_{\mathcal{G}, \mathcal{X}}$ that do not contain any constant $\alpha$.

Example 4. Refer back to Example 2. For the original $\mathcal{G}$, $G_{\mathcal{G}, \mathcal{X}}$ is a graph with 5 strongly connected components, 3 of which have no constants. Once we add $y = 0.3$, $G_{\mathcal{G}, \mathcal{X}}$ still has 5 strongly connected components, but the one of $y$ also contains the constant 0.3. When we add $x = 0.5$, $G_{\mathcal{G}, \mathcal{X}}$ contains a cycle including 0.3 and 0.5.

B Proofs for Section 3 (Brute-Force Solution)

B.1 Details for Eliminating Ties

We first define formally the notion of tied values.

Definition 8. Given a constraint set $\mathcal{C}$, we say that $x$ and $y$ have a persistent tie if $\mathcal{C}$ contains $x \preceq y$ and $y \preceq x$. (Remember that $\mathcal{C}$ is closed under implication; so in particular $x$ and $y$ have a persistent tie if they have an exact-value constraint to the same value.)

We say that $x \sim y$ if $x$ and $y$ have a persistent tie. It is immediate that $\sim$ is an equivalence relation.

We say that $\mathcal{C}$ has a persistent tie if there are $x \neq y$ such that $x \sim y$.

Lemma 5. For any constraint set $\mathcal{C}$ which has no persistent tie, the mass under the uniform distribution of possible worlds with ties is 0.

Proof. Let $W$ be the subset of $\text{pw}(\mathcal{C})$ of the possible worlds that have ties. We write $W$ as $\bigcup_{i \neq j} W_{i,j}$, where $W_{i,j}$ is the subset of $\text{pw}(\mathcal{C})$ of the possible worlds where there is a tie between $x_i$ and $x_j$. We clearly have:

$$V(\mathcal{C}) \leq |\mathcal{X}|^2 \max_{i \neq j} V(W_{i,j})$$

Hence, it suffices to show that for any $x_i, x_j \in \mathcal{X}$, $i \neq j$, $V(W_{i,j}) = 0$.

Hence, fix $x_i, x_j \in \mathcal{X}$, $i \neq j$. Let $d$ be the dimension of $\text{pw}(\mathcal{C})$ and $n := |\mathcal{X}|$. Remember that we have assumed that we only consider constraint sets $\mathcal{C}$ such that $\text{pw}(\mathcal{C})$ is non-empty. Because $\mathcal{C}$ is closed under implication, it cannot be the case that $t_i = t_j$ in every possible world $t \in \text{pw}(\mathcal{C})$; otherwise this implies that the constraints $x_i \preceq x_j$ and $x_j \preceq x_i$ are implied by $\mathcal{C}$, so they hold in $\mathcal{C}$, and $\mathcal{C}$ has a persistent tie, contradicting our assumption. Thus $\text{pw}(\mathcal{C})$ is a convex polytope of dimension $d$ in $[0,1]^n$ which is not included in the hyperplane defined by the equation $x_i = x_j$. We deduce that $W_{i,j}$, the projection of $\text{pw}(\mathcal{C})$ to that hyperplane, has dimension $d - 1$, so that its volume $V(W_{i,j}) = \int_{W_{i,j}} d\mu_\mathcal{C}$ is 0. This concludes the proof.

In the situation with persistent ties, we observe that, informally, tied variables can be collapsed to single variables, so as to remove the persistent ties without changing substantially the possible worlds and their probability. Formally:
**Definition 9.** We define $\mathcal{X}/\sim$ as the set of equivalence classes of $\mathcal{X}$ for $\sim$, and we define $\mathcal{C}/\sim$ to be the constraint set where every occurrence of a variable $x_i$ of $\mathcal{X}$ is replaced by the variable of $\mathcal{X}/\sim$ corresponding to its equivalence class.

The following is immediate:

**Lemma 6.** For any constraint set $\mathcal{C}$ on variable set $\mathcal{X}$ with uniform distribution $p_w$, the constraint set $\mathcal{C}/\sim$ is without persistent ties, can be computed in PTIME, and there is a bijection $f$ from $\text{pw}(\mathcal{C})$ to $\text{pw}(\mathcal{C}/\sim)$.

**Proof.** The computation of $\mathcal{C}/\sim$ can be performed by computing in PTIME the strongly connected components of the graph of the $\sim$ relation, which we can clearly compute in PTIME from $\mathcal{C}$. The absence of persistent ties is by observing that any persistent tie in the quotient between $X_i$ and $X_j$ in $\mathcal{X}/\sim$ would imply the existence of $x_i, x'_i \in X_i$ and $x_j, x'_j \in X_j$ such that $\mathcal{C}$ contains $x_i \leq x_j$ and $x'_i \leq x'_j$, from which we deduce the existence of a persistent tie between $x_i$ and $x_j$ as $\mathcal{C}$ contains $x_j \leq x'_i, x'_j \leq x_i$ by definition of $x_i \sim x'_i, x_j \sim x'_j$, so that we should have identified $x_i$ and $x_j$ when constructing $\mathcal{X}/\sim$.

The bijection between the possible worlds is obtained by mapping any possible world $t \in \text{pw}(\mathcal{C})$ to $t'$ obtained by choosing, for every $X_i \in \mathcal{X}/\sim$, the value $t'_j$ as $t_j$ such that $x_j \in X_i$. (The choice of representative does not matter as the definition of $\sim$ implies that $t_j = t'_j$ for every $t \in \text{pw}(\mathcal{C})$ whenever $x_j \sim x_j$.)

We conclude by showing how problems on $\mathcal{C}$ can be reduced to problems on $\mathcal{C}/\sim$, concluding the proof of Proposition 1.

The interpolation problem for a variable $x_i$ on $\mathcal{C}$ clearly reduces to interpolation for variable $X_i$ on $\mathcal{C}/\sim$.

The top-$k$ computation problem on $\mathcal{C}$ for a selection predicate $\sigma$ reduces to the top-$k$ computation problem on $\mathcal{C}/\sim$ for the selection predicate $\sigma'$ that selects variables $X_i$ such that some $x_j \in X_i$ is selected by $\sigma$. Given the top-$k$ result $(X_{1},\ldots,X_{k})$ on $\mathcal{C}/\sim$, we rewrite it to the top-$k$ result on $\mathcal{C}$ by replacing each $X_{i}$ by all the variables $x_j \in X_{i}$ selected by $\sigma$ (there is at least one of them), and truncating to length $k$.

**B.2 Total Order (Section 3.1)**

**Observation 2.** The marginal distribution of $x_i$ within the admissible polytope of $\mathcal{C}^n_{\alpha \beta}(\alpha, \beta)$ is exactly the distribution of the $i$-th order statistic for $n$ samples of the uniform distribution on $[\alpha, \beta]$.

**Proof.** The distribution on $n$ uniform independent samples in $[\alpha, \beta]$ can be described as first choosing a total order for the samples, uniformly among all permutations of $\{1, \ldots, n\}$ (with ties having a probability of 0 and thus being neglected). Then, the distribution for each total order is exactly that of $x_i$ when variables are relabeled.

**B.3 General Constraint Sets (Section 3.3)**

**Proposition 4.** For any constraint set $\mathcal{C}$ on $\mathcal{X}$, if $\bar{x}_i$ is the expected value of $x_i$ for all $x_i \in \mathcal{X}$, then $\bar{x}$ is the center of mass of $\text{pw}(\mathcal{C})$. 

15
We conclude by illustrating the entire construction of Section 3 with a simple, complete example. 

By computing the marginal distribution of \( y \) we have \( pw_x \infty \mu \) as well. These order constraints are closed under implication, so they also include \( x \leq z \). The figure below shows the Hasse diagram of the partial order \( \leq \) defined by \( C \) on \( X \) (edges are drawn from bottom to top). Note that ties have a probability of zero in \( pw(C) \).

![Hasse diagram](image)

The two linear extensions of \( C \) are \( T_1 : x \leq y \leq y' \leq z \) and \( T_2 : x \leq y' \leq y \leq z \). By Lemma 3 we have \( pw(T_1) = pw_{\{x,y\}}(C') \times \{ y \} \times \{ y, 1 \} \) where \( C' \) is defined on variables \( \{ x, y \} \) by \( 0 \leq x \leq y \leq 1 \). We can compute the volume of \( pw(T_1) \) as \( \alpha_1 = \frac{1}{2} \times (1 - \gamma) \) and similarly the volume of \( pw(T_2) \) is \( \alpha_2 = \gamma \times \frac{1-\gamma^2}{2} \).

Let us compute the expected value and marginal distribution of \( y \) for \( C \). We do so by computing the marginal distribution of \( y \) for \( T_1 \) and \( T_2 \). We first study \( T_1 \), where by Lemma 3 it suffices to determine it for \( C' \), where we compute by Proposition 3 that the marginal distribution of \( y \) has pdf \( f_1 : t \mapsto \frac{\gamma}{3} \) for \( t \in [0, \gamma] \). Verify that indeed \( f_1 \) integrates to 1 over \([0, \gamma]\), that its mean is, as we expect, \( \mu_1 := \frac{2}{3} \gamma \), and that, intuitively, larger values of \( t \) are more likely than smaller values. For \( T_2 \) we determine in the same way that the pdf of \( y \) is \( f_2 : t \mapsto \frac{2}{3} \frac{1-t}{\gamma (1-\gamma)} \) for \( t \in [\gamma, 1] \) which also integrates to 1 and has mean \( \mu_2 := \frac{1+2\gamma}{3} \). 

---

**Proof.** Fix \( C \) and a variable \( x_i \). We show that the expected value \( \bar{x}_i \) of \( x_i \) is the coordinate of the center of mass \( \bar{x} \) of \( pw(C) \) along vector \( u_i \), where \( u_i \) is the unit vector for the \( i \)-th coordinate.

The vector of the center of mass of \( pw(C) \) is given by: 

\[ \bar{x} := \frac{1}{pw(C)} \int_{pw(C)} \mathbf{r} \, dw \]

where \( \mathbf{r} \) is the function that gives the coordinates of each point in the basis of the \( u_i \)'s. Hence, 

\[ \bar{x}_i = \frac{1}{pw(C)} \int_{pw(C)} \mathbf{r} \cdot u_i \, dw. \]

Since the coordinate of the center of mass along \( u_i \) must be in \([0, 1]\), because \( pw(C) \subseteq [0, 1]^n \), we can split the integral to an integration along \( u_i \) (for \( t \) on the finite interval \([0, 1]\)), and then on the section of \( pw(C) \) projecting to coordinate \( t \) on \( u_i \); in other words, the intersection of \( pw(C) \) with the hyperplane with coordinate \( t \) on axis \( u_i \). Formally, 

\[ \bar{x}_i = \frac{1}{pw(C)} \int_{[0, 1]} V_t \, dt, \]

where \( V_t \) is the volume of the section of \( pw(C) \) along the hyperplane with coordinate \( t \) on axis \( u_i \), i.e., 

\[ \bar{x}_i = \frac{1}{pw(C)} \int_{[0, 1]} V_t(C_{x=t}) \, dt. \]

But this is exactly the definition of the expected value of the marginal distribution of Definition 1, which concludes the proof. \( \square \)
We deduce that the marginal distribution of \( y \) for \( \mathcal{C} \) has pdf \( \alpha_1 f_1 \) on \([0, \gamma]\) and \( \alpha_2 f_2 \) on \([\gamma, 1]\). The average of \( y \) is \( \mu = \frac{\alpha_1 \mu_1 + \alpha_2 \mu_2}{\alpha_1 + \alpha_2} \).

For \( \gamma = 1/2 \) we obtain \( \mu = 1/2 \), as we would expect by symmetry. For \( \gamma = 0.42 \) we obtain \( \mu \approx 0.473 \). Note that \( \mu \) goes to \( 1/3 \) and \( 2/3 \) as \( \gamma \) goes to \( 0 \) or \( 1 \) respectively, which is what we expect (because \( \mathcal{C} \) intuitively goes closer to the constraint sets \( 0 \leq y \leq z \leq 1 \) or \( 0 \leq x \leq y \leq 1 \) respectively, for which the mean of \( y \) is clearly \( 1/3 \) and \( 2/3 \) following linear interpolation). Hence, for \( \sigma \) selecting \( y \) and \( y' \), and \( k = 1 \), the top-\( k \) query for \( \mathcal{C} \) and \( \sigma \) returns \((y', \gamma)\) if \( 1/2 \geq \gamma \geq 1 \), and \((y, \mu)\) if \( 0 \geq \gamma \geq 1/2 \).

C Proofs for Section 4 (Hardness Results)

The following result is known about partial orders:

**Theorem 9** [BW91], Theorem 1. It is \#P-complete to compute, given a partial order \((\mathcal{X}, \leq_P)\), the number of linear extensions of \((\mathcal{X}, \leq_P)\).

Our hardness results will rely on the following consequence of that result:

**Corollary 3** [BW91], Theorems 6 and 7. For any input partial order \((\mathcal{X}, \leq_P)\) and two elements \( x \) and \( y \) of \( \mathcal{X} \), it is \#P-complete to compute the expected rank of \( x \) among linear extensions of \((\mathcal{X}, \leq_P)\), or the probability that \( x < y \) holds in linear extensions of \((\mathcal{X}, \leq_P)\).

We now prove our results:

**Theorem 5.** Given a set \( \mathcal{C} \) of order constraints with variable set \( \mathcal{X} \) and \( x \in \mathcal{X} \), determining the expected value of \( x \) in \( \text{pw}(\mathcal{C}) \) under the uniform distribution is \( \text{FP}^{\#P} \)-complete.

**Proof.** We first show \#P-hardness. We reduce from the problem of computing the expected rank of an element in a partial order, which is \#P-hard by Corollary 3. Let \( P = (\mathcal{X}, \leq_P) \) be a partial order, and let \( \mathcal{C} = \{ x \leq y \mid x, y \in \mathcal{X}, x \leq_P y \} \) be the set of constraints over \( \mathcal{X} \) corresponding to \( \leq_P \). Note that \( \mathcal{C} \) contains no exact-value constraints, and that ties have a probability of 0, since by antisymmetry there can be no constraints of the form \( x_i \leq x_j \) and \( x_j \leq x_i \) for \( i \neq j \) in a partial order. We claim that, writing \( n := |\mathcal{X}| \), if the expected value of \( x \) is \( v \), then the expected rank of \( x \) in \( P \) is \( n + 1 \) for \( 1/2 \geq \gamma \geq 1 \), and \((y, \mu)\) if \( 0 \geq \gamma \geq 1/2 \).

Let \( \mathcal{T}_1, \ldots, \mathcal{T}_N \) be the linear extensions of \( P \). Consider the partition of the set of possible worlds \( \text{pw}(\mathcal{C}) \) into subsets \( W_1, \ldots, W_N \), where \( W_i := \text{pw}(\mathcal{T}_i) \) for each \( i \). Their union is indeed exactly \( \text{pw}(\mathcal{C}) \), as every possible world of \( \mathcal{C} \) realizes some linear extension of \( P \) (possibly many, if there are ties), and as we assumed that the probability of a tie is 0, this is indeed a partition of \( \text{pw}(\mathcal{C}) \) up to worlds that have a total probability of 0. Now, it is easily seen by symmetry that all \( \mathcal{T}_i \) have the same volume. Hence, denoting \( v \) the (unknown) expected value of \( x \) in \( \mathcal{C} \), we have \( v = \frac{1}{n} \sum_{i=1}^{N} v_i \), where \( v_i \) is the expected value of \( x \) in \( \mathcal{T}_i \). As there are no exact-value constraints, by Section 3.1, we know that, following linear interpolation between 0 and 1, for any \( i \), we have \( v_i = \frac{r_i}{n+1} \), where \( r_i \) is the rank of \( x \) in \( \mathcal{T}_i \) (that is, its index in the total order \( \mathcal{T}_i \), between 1 and \( n \)).
Hence, we have \( v = \frac{1}{N(N+1)} \sum_{i=1}^{N} r_i \). Now, the expected rank of \( x \) in \( P \) is defined as \( r = \frac{1}{N} \sum_{i=1}^{N} r_i \), so it is clear that \( r = (n+1)v \), showing the correctness of the reduction.

We now show membership in \( \text{FP}^{\#P} \). Let \( \mathcal{C} \) be an arbitrary constraint set with order constraints and exact-value constraints on a variable set \( X \), and let \( n := |X| \). Use Proposition 1 to ensure that there are no ties. To simplify the reasoning, we will make all values occurring in exact-value constraints be integers that are multiples of \((n+1)!\); let \( \Delta \) be \((n+1)!\) times the product of the denominators of all exact-value constraints occurring in \( \mathcal{C} \), which can be computed in PTIME, and consider \( \mathcal{C}' \) the constraint set defined on \([0, \Delta]^n\) by keeping the same variables and order constraints, and replacing any exact-value constraint \( x_i = v \) by \( x_i = v\Delta \); \( \mathcal{C}' \) is computable in #P from \( \mathcal{C} \), and the polytope \( pw(\mathcal{C}') \) is obtained by scaling \( pw(\mathcal{C}) \) by a factor of \( \Delta \) along all dimensions; hence, if we can compute the expected value of \( x_i \) in \( \mathcal{C}' \) (which is the coordinate of the center of mass of \( pw(\mathcal{C}') \) on the component corresponding to \( x_i \)), we can compute the expected value of \( x_i \) in \( \mathcal{C} \) by dividing by \( \Delta \). So we can thus assume that \( pw(\mathcal{C}') \) is a polytope of \([0, \Delta]^n\) where all exact-value constraints are integers which are multiples of \((n+1)!\).

We use Lemma 5.2 of \cite{ACK+11} to argue that the volume of \( pw(\mathcal{C}) \) can be computed in \( \#P \). The PTIME generating Turing machine \( T \), given the constraint set \( \mathcal{C} \), chooses nondeterministically a linear extension of \((X, \leq_{\mathcal{C}})\), which can clearly be represented in polynomial space and checked in PTIME. The PTIME-computable function \( g \) computes, as a rational, the volume of the polytope for that linear extension, and does so according to the scheme of Section 3: the volume is the product of the volumes of each \( \mathcal{C}_i^{\#P}(\alpha, \beta) \), whose volume is \( \frac{B-\alpha}{n!} \). This is clearly PTIME-computable, and as \( \alpha \) and \( \beta \) are values occurring in exact-value constraints, they are integers and multiples of \( n! \), so the result is an integer, so the overall result is a product of integers, so it is an integer. Hence, by Lemma 5.2 of \cite{ACK+11}, \( V(\mathcal{C}') \), which is the sum of \( V(\mathcal{T}) \) across all linear extensions \( \mathcal{T} \) of \( \mathcal{C}' \) (because there are no ties), is PTIME-computable.

We now apply the same reasoning to show that the sum, across all linear extensions \( \mathcal{T} \), of \( V(\mathcal{T}) \) times the expected value of \( x_i \) in \( \mathcal{T} \), is computable in \( \#P \). Again, we use Lemma 5.2 of \cite{ACK+11}, with \( T \) enumerating linear extensions, and with a function \( g \) that computes the volume of the linear extension as above, and multiplies it by the expected value of \( x_i \), by linear interpolation in the right \( \mathcal{C}_i^{\#P} \) as in the previous section (it is an integer, as all values of exact-value constraints are multiples of \( n+1 \)). So this concludes, as the expected value \( v \) of \( x_i \) is \( \frac{1}{V(\mathcal{T})} \sum_{\mathcal{T}} V(\mathcal{T}) v_{\mathcal{T}} \), where \( v_{\mathcal{T}} \) is the expected value of \( x_j \) for \( \mathcal{T} \), and we can compute both the sum and the denominator in \#P. Hence, the result of the division, and reducing back to the answer for the original \( \mathcal{C} \), can be done in \( \text{FP}^{\#P} \).

\[\square\]

**Theorem 6.** The top-k computation problem of computing, given a constraint set \( \mathcal{C} \) over \( X \), a selection predicate \( \sigma \), and an integer \( k \), the ordered list of the \( k \) items of \( X_\sigma \) that have maximal expected values, is \( \text{FP}^{\#P} \)-complete, even if \( k \) is fixed to be 1 and the top-k answer does not include the expected value of the variables.

**Proof.** We first show \#P-hardness. We will perform a reduction from expected value computation: given a constraint set \( \mathcal{C} \) and a variable \( x_i \) in the variable set \( X \) of \( \mathcal{C} \),
determine the expected value of $x_i$ in $\mathcal{C}$, which is $\#P$-hard by Theorem 5. From the proof of that theorem, we can further assume that $\mathcal{C}$ contains no exact-value constraints, and hardness still holds. Write $n := |\mathcal{X}|$. Assume using Proposition 1 that the probability of ties in $\text{pw}(\mathcal{C})$ is zero.

We first observe, as in the proof of Theorem 5, that the expected value $v$ of $x_i$ can be written as $\frac{1}{N(n+1)} \sum r_\mathcal{X}$, where the sum is over all the linear extensions $\mathcal{I}$ of $\mathcal{C}$, $r_\mathcal{X} \in \{1, \ldots, n\}$ is the rank of $x_i$ in the linear extension $\mathcal{I}$, and $N \leq n!$ is the number of linear extensions. This implies that $v$ can be written as a rational $p/q$ with $0 \leq p \leq q$ and $0 \leq q \leq M$, where we write $M := (n+1)!$.

We determine this fraction $p/q$ using the algorithm of [Pap79], that proceeds by making queries of the form “is $p/q \leq p'/q'$” with $0 \leq p', q' \leq M$, and runs in time logarithmic in the value $M$, so polynomial in the input $\mathcal{C}$. To do so, we must describe how to decide in PTIME whether $v \leq p'/q'$ for $0 \leq p', q' \leq M$, using an oracle for the top-1 computation problem that does not return the expected values.

Fix $v' = p'/q'$ the query value and let $v = p/q$ be the unknown target value, the expected value of $x_i$. We illustrate how to decide whether $v \leq v'$. The general idea is to add a variable with exact-value constraint to $v'$ and compute the top-1 between $x_i$ and the new variable, but we need a slightly more complicated scheme because the top-1 answer variable can be arbitrary in the case where $v = v'$ (i.e., we have a tie in computing the top-1). Let $\epsilon := 1/(2(M^2 + 1))$, which is computable in PTIME in the value of $n$ (so in PTIME in the size of the input $\mathcal{C}$). Construct $\mathcal{C}'$ (resp., $\mathcal{C}', \mathcal{C}_{\epsilon}$) by adding an exact-value constraint $x' = v'$ (resp., $x' = v' - \epsilon, x' = v' + \epsilon$) for a fresh variable $x'$ (resp., $x', x_{\epsilon}'$). Now use the oracle for $\mathcal{C}'$ (resp., $\mathcal{C}', \mathcal{C}_{\epsilon}$) and the selection predicate that selects $x_i$ and $x'$ (resp., $x, x_{\epsilon}'$), taking $k = 1$ in all cases. The additional variables do not affect the expected value of $x_i$ in $\mathcal{C}', \mathcal{C}_{\epsilon}$, and $\mathcal{C}_{\epsilon}$, so it is also $v$ in them. Further, we know that $v = p/q, v' = p'/q'$, with $0 \leq p, q, p', q' \leq M$, hence either $v = v'$, or $|v - v'| \geq \frac{1}{2M^2}$. Hence, letting $S = \{v, v' - \epsilon, v' + \epsilon\}$, there are three possibilities: $v = v'$, $v < v'$ for all $v'' \in S$, or $v > v''$ for all $v'' \in S$. Thus, if the top-1 variable in all oracle calls is always $x_i$ (resp., never $x_i$), then we are sure that $v' > v$ (resp., $v < v'$). If some oracle calls return $x_i$ but not all of them, we are sure that $v = v'$. Hence, we can find out in PTIME using the oracle whether $v \leq v'$. This concludes the proof, as we then have an overall PTIME reduction from the $\text{FP}^\text{P}$-hard problem of expected value computation to the top-1 computation problem, showing that the latter is also $\text{FP}^\text{P}$-hard.

We now show $\text{FP}^\text{P}$-membership. Let $\mathcal{C}$ be a constraint set. By Theorem 5, computing the expected value of each variable is in $\text{FP}^\text{P}$. Now clearly we can sort the resulting values in PTIME to deduce the top-$k$ for any input $k$ and $\sigma$, proving the result. \qed

D Proofs for Section 5 (Tractable Cases)

Lemma 4. There exists a bijective correspondence between $\text{pw}(\mathcal{C}_1) \times \cdots \times \text{pw}(\mathcal{C}_n)$ and $\text{pw}(\mathcal{C})$, obtained by merging the variables with exact-value constraints.

Proof. It is immediate that any $t \in \text{pw}(\mathcal{C})$ thus yields a tuple of possible worlds of $\mathcal{C}_1, \ldots, \mathcal{C}_n$. Conversely, consider any tuple of possible worlds of $\mathcal{C}_1, \ldots, \mathcal{C}_n$. Note first that, indeed, those tuples match on the variables that occur in multiple sets (which are
Theorem 7. For any tree-shaped constraint set $\mathcal{C}$ on variable set $\mathcal{X}$, we can compute $V(\mathcal{C})$ in time $O(|\mathcal{C}| + |\mathcal{X}|^2)$.

Proof. Let $T$ be the tree with vertex set $\mathcal{X}$ which is the Hasse diagram of the order constraints imposed by $\mathcal{C}$. For any variable $x \in \mathcal{X}$ that has no exact-value constraint (so it is not the root of $T$ or a leaf of $T$), let $\mathcal{C}_v$ be the constraint set obtained as a subset of $\mathcal{C}$ by keeping only constraints between $x$ and its descendants in $T$, as well as between $x$ and its parent. For $v \in [0, 1]$, we call $V_v(x)$ the $d$-volume of $\text{pw}(\mathcal{C}_v \cup \{x = v\})$ where $x'$ is the parent of $x$ and $d$ is the dimension of $\text{pw}(\mathcal{C}_v)$. In other words, $V_v(x)$ is the $d$-volume of the admissible polytope for the subtree $T_v$ of $T$ rooted at $x$, as a function of the minimum value on $x$ imposed by the exact-value constraint on the parent of $x$. It is clear that, letting $x'_v$ be the one child of the root $x_v$ of $T$, we have $V(\mathcal{C}) = V_{x'_v}(v)$, where $v$ is the exact value imposed on $x_v$.

We show by induction on $T$ that, for any node $x$ of $T$, letting $m_x$ be the minimum, among all leaves that are descendants of $x$, of the values to which those leaves have an exact-value constraint, the function $V_x$ is zero in the interval $[m_x, 1]$ and can be expressed in $[0, m_x]$ as a polynomial whose degree is at most the number of nodes in $T_x$, written $|T_x|$. Since the probabilities of ties is 0, we have $m_x > 0$ for all $x$.

The base case is for a node $x$ of $T$ which has only leaves as children; in this case it is clear that $V_x(v)$ is $m_x - v$ for $v \in [0, m_x]$, and is zero otherwise. For the inductive case, let $x$ be a variable. It is clear that $V_x(v)$ is 0 for $v \in [m_x, 1]$. Otherwise, let $v' \in [0, m_x]$ be the value of the parent $x'$ of $x$. For every value $v' \leq v \leq m_x$ of $x$, consider the constraint set $\mathcal{C}_{x,v'} = \mathcal{C}_v \cup \{x' = v', x = v\}$. By Lemma 4, we have $V_{x,v'} = \prod_i V_{x_i}(v)$ where $x_1, \ldots, x_l$ are the children of $x$. Hence, by definition of the volume, we know that $V_{x,v'} = \int_{v'}^{m_x} \prod_i V_{x_i}(v)dv$. Now, we use the induction hypothesis to deduce that $V_{x_i}(v)$, for all $i$, in the interval $[0, m_x]$, is a polynomial whose degree is at most $|T_{x_i}|$. Hence, as the product of polynomials is a polynomial whose degree is the sum of the input polynomials, and integrating a polynomial yields a polynomial whose degree is one plus that of the input polynomial, $V_x$ in the interval $[0, m_x]$ is a polynomial whose degree is at most $|T_x|$. Hence, we have proved the claim by induction, and we use it to determine $V(\mathcal{C})$ as explained in the first paragraph.

We now prove that the computation is quadratic. We first assume that the tree $T$ is binary. We show by induction that there exists a constant $\alpha \geq 0$ such that the computation of the polynomial $V_{x_0}$ in expanded form has cost less than $\alpha m_0^2$, where $n_i$ is $|T_{x_i}|$. The
claim is clearly true for nodes where all children are leaves, because the cost is linear in the number of child nodes as long as $\alpha$ is at least the number the number of operations per node $x_0$. For the induction step, if $x_i$ is an internal node, let $x_p$ and $x_q$ be the two children. By induction hypothesis, computing $V_{x_p}$ and $V_{x_q}$ in expanded form has cost $\leq \alpha(n_p^2 + n_q^2)$. Remembering that arithmetic operations on rationals are assumed to take unit time, computing the product of $V_{x_p}$ and $V_{x_q}$ in expanded form has cost linear in the product of the degrees of $V_{x_p}$ and $V_{x_q}$ which are less than $n_p$ and $n_q$, so the cost of computing the product is $\leq \alpha_1 n_p n_q$ for some constant $\alpha_1$. Integrating has cost linear in the degree of the resulting polynomial, that is, $n_p + n_q$. So the total cost of computing $V_{x_i}$ is $\leq \alpha(n_p^2 + n_q^2) + \alpha_1 n_p n_q + \alpha_3(n_p + n_q) + \alpha_3$ for some constants $\alpha_2$, $\alpha_3$. Now, as $n_q = n_i - n_p - 1$, computing $V_{x_i}$ costs less than:

$$\begin{align*}
\alpha n_i^2 + 2\alpha n_p^2 + \alpha - 2\alpha n_i n_p - 2\alpha n_i + 2\alpha n_p \\
+ \alpha_1 n_p n_q + \alpha_2 n_i - \alpha_2 + \alpha_3
= \alpha n_i^2 + (\alpha_1 - 2\alpha)n_p n_q + (\alpha_2 - 2\alpha)n_i + \alpha - \alpha_2 + \alpha_3
\end{align*}$$

As long as $\alpha$ is set to be at least $\frac{\alpha_1}{2}$ and at least $\frac{\alpha_2}{\alpha_3}$, the second and third terms are negative, which means (since $n_p n_q$ and $n_i$ are both $\geq 1$) that $V_{x_i}$ costs less than:

$$\begin{align*}
\alpha n_i^2 + \alpha_1 - 2\alpha + \alpha_2 - 2\alpha + \alpha - \alpha_2 + \alpha_3 \\
= \alpha n_i^2 - 3\alpha + \alpha_1 + \alpha_3 \geq \alpha n_i^2
\end{align*}$$

if $\alpha \geq \frac{\alpha_1 + \alpha_2}{3}$. This concludes the induction case, for $\alpha$ set to an arbitrary value $\geq \max(\alpha_0, \frac{\alpha_1}{2}, \frac{\alpha_2}{\alpha_3}, \frac{\alpha_1 + \alpha_2}{3})$. Hence the claim is proven if $T$ is binary.

If $T$ is not binary, we use the associativity of product to make $T$ binary, by adding virtual nodes that represent the computation of the product. In so doing, the size of $T$ increases only by a constant multiplicative factor (recall that the number of internal nodes in a full binary tree is one less than the number of leaves, meaning that the total number of nodes in a binary expansion of a $n$-ary product is less than twice the number of operands of the product). So the claim also holds for arbitrary $T$.

**Theorem 8.** For any tree-shaped constraint set $C$ on variable set $X$ and variable $x \in X$ with no exact-value constraint, the marginal distribution for $x$ is piecewise polynomial and can be computed in time $O(|C| + |X|^3)$.

**Proof.** Recall that $C'_{[x=v]}$ is $C$ plus the exact-value constraint $x = v$. For any variable $x'$, we let $n_{x'}$ be the minimum, among all leaves reachable from $x'$, of the values to which those leaves have an exact-value constraint. By definition, the marginal distribution for $x$ is $v \mapsto \frac{1}{V(C')} V(C'_{[x=v]})$. We have seen in Theorem 7 that $\frac{1}{V(C')} V(C'_{[x=v]})$ can be computed in quadratic time; we now focus on the function $V(C'_{[x=v]})$.

By Lemma 4, letting $x_1, \ldots, x_n$ be the children of $x$, $D_1, \ldots, D_n$ be their descendants (the $x_i$ included), and $D$ be all variables except $x$ and its descendants, that is, $D := X \setminus \{x\} \cup \bigcup_i D_i$, we can express $V(C'_{[x=v]})$ as $V'_1(v) \times \prod_i V_{x_i}(v)$, where $V_{x_i}$ is as in the proof of Theorem 7, and $V'_1(v)$ is the volume of the constraint set $C'_{[x=v]}$ over $D$ obtained by keeping all constraints in $C$ about variables of $D$, plus the exact-value constraint $x = v$.  

21
Indeed, the uninfluenced classes of $\mathcal{G}_{i=x}$ are clearly $D, D_1, \ldots, D_n$, except that the root and leaves are in singleton classes because they have exact-value constraints.

We know by the proof of Theorem 7 that $V_{x_i}$, in the interval $[0, m_i]$, is a polynomial whose degree is at most $|T_{|x_i}|$, and that it can be computed in $O(|T_{|x_i}|^2)$. Hence, the product of the $V_{x_i}(v)$ can be computed in cubic time overall and has linear degree. We thus focus on $\mathcal{G}'_{x_i}$, for which it suffices to show that $V(\mathcal{G}'_{x_i})$ is a piecewise polynomial function with linearly many pieces, each piece having a linear degree and being computable in quadratic time. Indeed, this suffices to justify that computing the product of $V(\mathcal{G}'_{x_i})$ with $\prod_i V_{x_i}(v)$, and integrating to obtain the marginal distribution, can be done in cubic time, and that the result is indeed piecewise polynomial.

For any node $x_i$ of $D$ with no exact-value constraint, we let $V'_{x_i}(v,v')$ be the volume of the constraint set obtained by restricting $\mathcal{G}'_{x_i}$ to the descendants of the parent $x_i'$ of $x_i$ and adding the exact-value constraint $x'_i = v$. We let $(v_1, \ldots, v_q)$ be the values occurring in exact-value constraints in $\mathcal{G}'$, in increasing order. We show by induction on $D$ the following claim: for any $1 \leq i < q$, for any variable $x_i$ in $D$ with no exact-value constraint, in the intervals $v \in [0, m_i]$ and $v' \in [v_i, v_{i+1}]$, $V'_{x_i}(v,v')$ can be expressed as $P(v) + v'P'(v)$, where $P$ and $P'$ are polynomials of degree at most $|T_{|x_i}|$ and can be computed in quadratic time.

The proof is the same as in Theorem 7: for the base case where all children of $x_i$ have exact-value constraints, $V_{x_i}(v,v')$ is either $m_i - v$ if $x$ is not reachable from $x_i$ or $v_i \geq m_i$, or $v' - v$ otherwise. For the inductive case, we do the same argument as before, noting that, clearly, taking the product of the $V'_{x_i}(v,v')$ among the children of $x_i$, the variable $v'$ occurs in at most one of them, namely the one from which $x$ is reachable. We conclude that $V'_{x_i}(v,v')$ is indeed a piecewise polynomial function with linearly many pieces that have a linear degree, by evaluating $V'_{x''_i}(v'',v')$, where $x''_i$ is the one child of the root of $T$ and $v''$ is the value to which it has an exact-value constraint, and the overall computation time is cubic.

\end{proof}

### E Alternative Top-K definitions

This appendix studies alternative ways to define the top-$k$ computation problem.

Recall that given a constraint set $\mathcal{G}$ on variables $\mathcal{X}$, given an integer $k$ and selection predicate $\sigma$ selecting a subset $\mathcal{X}_\sigma$ of $\mathcal{X}$, letting $k' := \min(k, \vert \mathcal{X}_\sigma \vert )$, we defined the result of the top-$k$ computation as the $k'$ variables with the highest expected values out of $\mathcal{X}_\sigma$, under the uniform distribution on $\text{pw}(\mathcal{G})$, along with their expected values, sorted by decreasing value. We rename this to local-top-$k$.

Rather than taking the top-$k$ of individual variables in this sense, we could alternatively define the top-$k$ as a choice between variable sequences, namely, computing the ordered sequence of $k'$ variables that has the highest probability of being the $k'$ largest variables (in decreasing order) among those of $\mathcal{X}_\sigma$, for the uniform distribution on $\text{pw}(\mathcal{G})$. We call this alternative definition U-Top-$k$ by analogy with $\text{SICCC07, CLY09}$. Note that U-Top-$k$ cannot return a score per variable, but can instead return a global probability that the $k'$ variables returned are indeed the top-$k$. We can show that the U-top-$k$ and local-top-$k$ definitions do not match on some constraint sets:
Lemma 7. There is a constraint set $\mathcal{C}$ and selection predicate $\sigma$ such that local-top-$k$ and U-top-$k$ do not match, even for $k = 1$ and without returning expected values or probabilities.

Proof. Let $\mu = \frac{2}{3}$, $m = \frac{1}{\sqrt{2}}$, and any $v$ such that $\mu < v < m$. Consider variables $x$, $x'$ and $y$, with the constraint set that imposes $x' \leq x$ and $y = v$. Fix $k = 1$ and consider the predicate $\sigma$ that selects all variables. It is immediate by linear interpolation that the expected value of $x$ is $\mu$. Further, it is easily computed that the marginal distribution $p_x$ of $x$ has the pdf $p_x: t \mapsto \frac{2}{t}$ on $[0, 1]$, for which we can check that $\int_{m}^{1} p_x(t) dt = \int_{0}^{m} p_x(t) dt$. Hence, as $v < m$, the probability that $x$ is larger than $v$ in $\text{pw}(\mathcal{C})$ is $> \frac{1}{2}$. This implies that the U-top-1 answer is $x$. By contrast, as $\mu < m$, the local-top-1 answer is $y$. \qed

While we leave open the precise complexity of U-top-$k$ computation in our setting, we can easily provide a simple algorithm that shows it to be computable in PSPACE and polynomial time in the number of linear extensions of $\mathcal{C}$. Intuitively, we can compute the probability of each linear extension as in Algorithm 1, and then sum on linear extensions depending on the top-$k$ sequence that they realize on the variables retained by $\sigma$, to determine the probability of each top-$k$ sequence. We then return the sequence with the highest probability. Hence, we have shown:

Proposition 5. For any constraint set $\mathcal{C}$ over $\mathcal{X}$, integer $k$ and selection predicate $\sigma$, the U-top-$k$ query for $\mathcal{C}$ and $\sigma$ can be computed in PSPACE and in time $O(\text{poly}(N))$, where $N$ is the number of linear extensions of $\mathcal{C}$.

Unlike Theorem 6, however, this result does not imply F#P-membership, because, when selecting the most probable sequence, we compare a number of candidate sequences which may not be polynomial (as $k$ is not fixed). We leave to future work an investigation of the precise complexity of U-top-$k$, although we suspect it to be F#P-complete, like local-top-$k$.

We now show that in our setting (namely, a uniform distribution over possible worlds defined by order constraints) U-top-$k$ does not satisfy a natural containment property [CLY09], a negative result that also holds in other settings [CLY09]. We define the containment property as follows, taking variable order into account:

Definition 10. We say a top-$k$ definition satisfies the containment property if for any constraint set $\mathcal{C}$ on variables $\mathcal{X}$, selection predicate $\sigma$ retaining variables $\mathcal{X}_\sigma$, and $k < |\mathcal{X}_\sigma|$, letting $S_k$ and $S_{k+1}$ be the top-$k$ and top-$(k+1)$ variables (i.e., without scores), we have that $S_k$ is a strict prefix of $S_{k+1}$.

It is immediate by definition that local-top-$k$ satisfies the containment property, except in the case of ties. However:

Lemma 8. There is a constraint set $\mathcal{C}$ such that the U-top-$k$ definition does not satisfy the containment property for the uniform distribution on $\text{pw}(\mathcal{C})$, even though there are no ties.
Proof. Consider variables \( x_6, x_7, x_1 \) and the constraint set \( \mathcal{C} \) that imposes \( x_1 \leq x_6 \), \( x_1 = .45 \) and \( x_7 = .75 \). Let \( k = 2 \) and consider \( \sigma \) that selects all variables. The volume of \( \text{pw}(\mathcal{C}) \) is computed to be \( V := (1 - .45) \times 1 \).

The variable sequence \((x_6, x_7)\) is the top-2 if \( x_6 \geq x_7 \geq x_1 \), the probability of which is easily computed as \( \frac{1}{V}(1 - .75) \times .75 \), and checking other possibilities we deduce that \((x_6, x_7)\) is in fact the most likely such sequence, so it is the U-top-2 answer.

Now, set \( k := 3 \). To compute U-top-3, considering all 8 possible answers, the one with highest probability is \( x_7 \geq x_6 \geq x_1 \), with probability \( \frac{1}{V}(.75 - .45) \times .45 \). Hence, the U-top-3 result is \((x_7, x_6, x_1)\), of which \((x_6, x_7)\) is not a prefix. \( \square \)

We see this as a drawback of U-top-\( k \) compared to our local-top-\( k \) definition. In fact, there are other possible definitions of top-\( k \) that do not satisfy the containment property in the absence of ties, even for definitions that return individual variables sorted by a score, rather than full sequences. This is for instance the case of the global-top-\( k \) definition [ZC09], which we now consider; once again, we show that similarly to other settings [CLY09], the containment property does not hold for global-top-\( k \) in our setting.

**Definition 11.** The global-top-\( k \) query, for a constraint set \( \mathcal{C} \) on variables \( \mathcal{X} \), selection predicate \( \sigma \) selecting variables \( \mathcal{X}_\sigma \), and integer \( k \), letting \( k' := \min(\lvert \mathcal{X}_\sigma \rvert, k) \), returns the \( k' \) variables that have the highest probability in the uniform distribution on \( \text{pw}(\mathcal{C}) \) to be among the variables with the \( k' \) highest values, sorted by decreasing probability.

**Lemma 9.** There is a constraint set \( \mathcal{C} \) such that the global-top-\( k \) definition does not satisfy the containment property for the uniform distribution on \( \text{pw}(\mathcal{C}) \), even though there are no ties.

**Proof.** Consider the example used in the proof of Lemma 8, but replacing .75 by .73, yielding \( \mathcal{C}' \). We compute the volume of \( \text{pw}(\mathcal{C}') \) as \( V \) again.

Set \( k := 1 \). Variable \( x_6 \) has the highest value with probability \( \frac{1}{V}(.73 \times (1 - .73) + (1 - .73)^2)/2 \). Variable \( x_7 \) has the highest value with probability \( \frac{1}{V}(.73 - .45) \times .73 \), which is less. The probability for \( x_1 \) is also less. So the global-top-1 is \((x_6)\).

Now, set \( k := 2 \). Variable \( x_6 \) has one of the two highest values in all cases except for \( x_7 \geq x_6 \geq x_1 \) and \( x_6 \geq x_7 \geq x_1 \), so it has one of the two highest values with probability \( 1 - \frac{1}{V}( (.73 - .45) \times (1 - .73) + (.73 - .45)^2)/2 \). However, variable \( x_7 \) has one of the two highest values in all cases except for \( x_6 \geq x_7 \geq x_1 \) and \( x_6 \geq x_7 \geq x_1 \), so it has one of the two highest values with probability \( 1 - \frac{1}{V}(1 - .73)^2 \), which is more. Hence the first variable of the global-top-2 is \( x_7 \) and not \( x_6 \). \( \square \)

There are yet other possible definitions of top-\( k \), see [CLY09, ZC09]. However, in the context of [CLY09], these definitions do not satisfy the containment property either, except for two of them. The first, U-kRanks [SICCC07], does not satisfy the natural property that top-\( k \) answers always contain \( k \) different variables. The second, expected ranks [CLY09], matches our local-top-\( k \) definition of taking the variables with highest expected values, but uses ranks instead of values, so the definition is value-independent; the value of variables does not modify their ordering in the top-\( k \) as long as the order between values remains the same. While this requirement makes sense for top-\( k \) queries designed to return tuples, as in [CLY09], we argue that this is less sensible in our setting.
where we are interested in the numerical value of variables. This, in addition to the connection with interpolation, justifies our choice of local-top-$k$ as the main definition of top-$k$ studied in our work.

## F Other Interpolation Schemes

This appendix re-examines our choice of definition for the interpolation (and hence top-$k$) computation problem. We first define abstractly what an interpolation scheme is:

**Definition 12.** An interpolation scheme is a function that maps any constraint set $C$ on variables $X$ to a mapping from $X$ to $[0, 1]$, mapping each variable to its interpolated value following $C$.

The expected value interpolation that we studied, that maps each variable to its expected value under the uniform distribution on $\text{pw}(C)$, is an interpolation scheme. We call it the uniform scheme.

We then define the following desirable property for stable interpolation schemes:

**Definition 13.** We say that an interpolation scheme is stable if, for any constraint set $C$, letting $f$ be the function mapping variables in $X$ to their interpolated value for $C$, for any subset of variables $X' \subseteq X$, the constraint set $C' = C \cup \{x = f(x) \mid x \in X'\}$, letting $f'$ be the function mapping variables in $X$ to their interpolated value for $C'$, we have $f = f'$.

In other words, an interpolation scheme is stable if assigning variables to their interpolated value does not change the result of interpolation elsewhere. This property is clearly respected by vanilla linear interpolation on total orders. However, this section shows that the stability property is not respected by the uniform scheme:

**Lemma 10.** The uniform scheme is not stable, even on tree-shaped constraint sets.

**Proof.** Consider the set of variables $\{x_r, x_a, x_b, x_c, x_d, x_e\}$ and the constraint set $C$ formed of the order constraints $x_r \leq x_a \leq x_b \leq x_c \leq x_d \leq x_e$, and the exact-value constraints $x_r = 0, x_c = .5$ and $x_e = 1$. We can compute that the interpolated values for $x_a$ and for $x_b$ are $3/20$ and $13/40$ respectively. However, adding the exact-value constraint $x_b = 13/40$, the interpolated value for $x_a$ becomes $611/4020$, which is different from $3/20$.

As pointed out in the Conclusion, we leave open the question of designing a principled alternate interpolation scheme for incomplete data that would satisfy the stability property.

## REFERENCES FOR THE APPENDIX
