Electronic Colloquium on Computational Complexity, Report No. 145 (2020)

An Improved Exponential-Time Approximation Algorithm for Fully-Alternating Games Against Nature

Andrew Drucker*

September 23, 2020

Abstract

"Games against Nature" [Pap85] are two-player games of perfect information, in which one player's moves are made randomly (here, uniformly); the final payoff to the non-random player is given by some $[0, 1]$-valued function of the move history. Estimating the value of such games under optimal play, and computing near-optimal strategies, is an important goal in the study of decision-making under uncertainty, and has seen significant research in AI and allied areas [HRTP11], with only experimental evaluation of most algorithms' performance. The problem's PSPACE-completeness does not rule out nontrivial algorithms. Improved algorithms with theoretical guarantees are known in various cases where the payoff function F has special structure, and Littman, Majercik, and Pitassi [LMP01] give a sampling-based improved algorithm for general F, for turn-orders which restrict the number of non-random player strategies.

We study the case of general F for which the players strictly alternate with binary moves $(w_1, r_1, w_2, r_2, \ldots, w_{n/2}, r_{n/2})$—for which the approach of [LMP01] does not improve over brute force. We give a randomized algorithm to approximate the value of such games under optimal play, and to execute near-optimal strategies. Our algorithm achieves exponential savings over brute-force, making $2^{(1-\delta)n}$ queries to F for some absolute constant $\delta > 0$, and certifies a lower bound $\hat{v}$ on the game value v with additive expected error bounded as $\mathbb{E}[v - \hat{v}] \leq \exp(-\Omega(n))$. (On the downside, δ is tiny and the algorithm uses exponential space.)

Our algorithm is recursive, and bootstraps a "base case" algorithm for fixed-size inputs. The method of recursive composition used, the specific base-case guarantees needed, and the steps to establish these guarantees are interesting and, we feel, likely to find uses beyond the present work.

*Computer Science Dept., University of Chicago. Email: andy.drucker@gmail.com

ISSN 1433-8092

Contents

1 Introduction

1.1 Games against Nature

Games against Nature, as described by Papadimitriou [Pap85], are a natural and general formulation of a ubiquitous problem: *decision-making under persistent uncertainty*, in scenarios where our knowledge of the world is continually *evolving* through observation. This process is modeled in [Pap85] as a game in which a maximizing, "strategic" player takes turns with a randomly-playing player, Nature; the strategic player's moves are described by strings $w^1, w^2, \dots, w^k$ and Nature's random moves are described by strings $r^1, \dots, r^k$, which we will assume are chosen uniformly. These moves alternate, and each w^i is chosen with perfect knowledge of the preceding moves $w^1, r^1, \dots, w^{i-1}, r^{i-1}$. The *payoff* to the strategic player is then determined by some function F as a numerical value, $F(w^1, r^1, \dots, w^k, r^k) \in [0, 1]$. The function F is known to the strategic player.

In this work, we will focus on the case of "fully-alternating, binary" moves where each w_i and r_i is chosen from $\{0, 1\}$. However, these games are also of interest in more general setting where each w_i, r_i are bitstrings of some predetermined lengths, possibly depending on the specification of the game F and on the value i. This framework can represent a wide variety of practical scenarios for decision-making. It is thus natural to study the following computational problems: given as input a description of F,

1. Compute the (exact or approximate) expected payoff under *optimal* play, i.e. the game's *value* to the maximizing player—given by

$$val(F) \;=\; \max_{w^1} \; \mathbb{E}_{r^1} \dots \max_{w^k} \; \mathbb{E}_{r^k} \; \left[F(w^1, r^1, \dots, w^k, r^k)\right] \; . \tag{1}$$

2. Implement an optimal or near-optimal *strategy* choosing next moves w^i in response to observed previous moves $w^1, r^1, \dots, w^{i-1}, r^{i-1}$.

These problems are usually considered in cases where the description of F is *succinct*. Assume for now that F is specified by a description of a Boolean circuit C_F computing F, a circuit of size polynomially bounded in the total bitlength of $w^1, r^1, \dots, w^k, r^k$.

1.2 Related work in AI and other areas

Many probabilistic reasoning and planning tasks are considered in AI, and in allied areas including decision and control theory, constraint programming, and operations research (see [HRTP11] for some interdisciplinary overview). One popular framework for this study is that of *Markov Decision Processes* [Put94, BDH99]—an intensively-studied model in which a strategic player's and Nature's moves affect a system state as described by a state automaton, and payoffs are delivered incrementally, determined by the current state. Various algorithmic results are known, for state spaces that are either small or in some way well-structured, see e.g., [PT87, BDG00, GKPV03]; however, these models may fail to succinctly capture tasks in some application areas. Other existing work on more general games against Nature, e.g., in [LMP01, Wal02, TF10, TF11, LWJ18], often studies a fairly expressive class of functions F such as CNF formulas or more general Boolean formulas, giving mostly experimental assessments of algorithm performance; these works have adapted concepts and techniques from classical SAT solvers. (When the move-structure and payoff function of the game F are indicated by a Boolean formula quantified by "Exists" and "For-Random" quantifiers, the problem is known as *stochastic satisfiability* [Pap85, LMP01].)

Decision-making under persistent uncertainty is also studied in a research area known as *competitive analysis of online algorithms* (see, e.g., [Alb03]), in which a decision-maker views an input string x one bit (or one "item") at a time and must make decisions at multiple stages. The game being played is usually completely *known* in advance, rather than given as input; instead, the input x plays a role analogous to Nature's moves, but is no longer assumed to come from a known distribution. Instead researchers seek play-strategies for particular games or classes of games that minimize a *regret* measure, comparing the algorithm's performance with the best performance obtainable with foreknowledge of the input x. This can be a useful benchmark in applications, especially when a realistic distribution on Nature's behavior is not available. By contrast, in the current paper's framework we assume a simple distributional model for Nature's play, but regard the game's *description* as a variable input that must be inspected and reasoned about.

1.3 Complexity classification, algorithms, and reductions

Even in their approximate versions, problems 1-2 above for Eq. (1) are $PSPACE$-complete in the fully-alternating, binary case (where each w^i, r^i are a single bit). This follows from a beautiful line of work showing the power of *interactive proofs* for decision problems [Bab85, GMR89, GS86, LFKN92, Sha92]. A public-coin interactive proof system can be considered a family of games against Nature—games designed to achieve epistemic goals through interaction. These proof systems are specified by a polynomial-time Verifier: an efficient algorithm that simultaneously generates Nature's random moves, serves as a referee for the game, and learns something by playing the game with an optimal strategic player.

This $PSPACE$-completeness result can be contrasted with the "bounded-alternation" case where the bitstrings w^i, r^i are individually large but $k = O(1)$. In this case the approximation problem is known to lie in AM [Bab85]. However, while the fully-alternating case is "hardest" when viewed through the lens of complexity classes, such alternations may also *enable* algorithmic improvements. This phenomenon appears in the related setting of

2-player, zero-sum games with an *adversarial* opponent, where simple (randomized) pruning techniques give improved exponential-time algorithms for perfect play. These techniques yield exponential savings over brute force for binary fully-alternating games [Sni85, SW86] and form part of the bedrock of game-playing AI. Algorithmic improvements for more general move-bitlengths (or quantifier structure, in the framework of quantified Boolean formulas) against adversarial opponents were given by Santhanam and Williams [SW15].

The prospect of such improvements for games against Nature was highlighted by Littman, Majercik, and Pitassi [LMP01]. The authors gave a sampling-based evaluation approach, called SampleEvalSSAT, and showed that, for games against Nature in which the number of distinct strategic-player policies is relatively small (due to the structure of the players' move-lengths), the algorithm improves over brute force [LMP01, Sec. 2.3]. They also gave some general search algorithms (building on [Lit99, ML99]) with empirical performance studies, with followup work appearing in e.g. [TF10, TF11].

The expressive power of interactive proofs has also been used in other ways to explore the "fine-grained complexity" of various algorithmic problems. For example, Williams [Wil16] shows, among related results, that 3-round interactive proofs can yield exponential savings for evaluating quantified Boolean formulas (with runtime $2^{2n/3+o(n)}$, faster than the randomized algorithm of [Sni85]), as well as exponential-time MA proof systems improving on best known runtimes for various NP problems, casting an interesting perspective on the study of their exponential-time complexity.

Chen et al. [CGL+19], building on work of Abboud and Rubinstein [AR18], use Arthur-Merlin communication protocols for space-bounded computation of Aaronson and Wigderson [AW09], along with some additional reductions, to show conditional *hardness* results for several versions of the Longest Common Subsequence (LCS) problem for pairs of strings. This is a polynomial-time solvable problem whose complexity has seen intensive study (see, e.g., [RSSS19]). Both of [AR18, CGL+19], extending a project of of [AB17], also presented evidence of the possible difficulty of obtaining certain improved deterministic approximation algorithms for LCS, by showing they would imply breakthrough circuit lower bounds.

An important tool from [AR18], used also in [CGL+19], is an approximation-preserving reduction from a "Tropical Tensor Similarity" problem, involving alternating Max and $\mathbb{E}$ quantifiers, to LCS. This problem contains some specific structure tailored to a communication setting in the authors' work, but the core part of the reduction applies directly to games against Nature as in Eq. (1): If $F(r, w)$ is Boolean-valued on n total input bits then the reduction produces two output strings u, v each of length $N = 2^n$, each with symbols over an alphabet Σ of size $\Theta(N)$. The LCS of the pair is, in the fully-alternating case, of length $\sqrt{N} \cdot val(F)$.

While the overall reduction is not sublinear-time, it can be observed to have efficient local-computability properties. Still, we are not aware of any algorithm for LCS which, in combination with the reduction of [AR18], would yield a nontrivial speedup for computing values of fully-alternating games against Nature. In particular these would need to have sublinear query complexity, and current fastest approximation algorithms for large alphabet-size ([HSSS19], see also [RSSS19]) incur linear runtime and queries while returning solutions which, in cases where the LCS is of size $\leq N^{1-\Omega(1)}$, may be a multiplicative $N^{-\Omega(1)}$ factor smaller than optimal in expectation.

A version of the reduction of [AR18] can also be given for the Longest Increasing Subsequence (LIS) problem, guided by the combination of [AR18] and a remark in [RSSS19, p. 1123] on a connection between LIS and LCS. On input a Boolean-valued game against Nature F as above, not all-zero, the output is a list of $N = 2^n$ integers, each of bitlength $O(n)$, whose LIS is, in the fully-alternating case, of length $val(F) \cdot \sqrt{N}$. See Section 9 for details. Now, some nontrivial query-efficient approximation algorithms for LIS are known—see [SS17, RSSS19] and references—but their guarantees are not interesting when the LIS is as small as $\sqrt{N}$; the algorithm of [SS17] gives fast, nontrivial multiplicative approximations only for LIS size $N^{1-o(1)}$, and the algorithm of [RSSS19] on input with LIS of size $N^{1-\varepsilon}$ finds a subsequence of size $\Omega(N^{1-3\varepsilon})$. The portion of this latter algorithm designed to handle sufficiently-small LIS size ($< N^{19/20}$) employs a simple subsampling idea; translating to the setting of fully-alternating games against Nature, and working flexibly, yields an algorithm nonadaptively querying a random p fraction of queries in expectation (for chosen value $p \in (0, 1)$) and achieving an expected certified lower bound of $p \cdot val(F)$ on the game value. Such an algorithm and its analysis may also be directly verified without the LIS reduction. In this work, we will give an $N^{1-\Omega(1)}$-query algorithm for approximating Eq. (1), whose approximation quality is generally far better, achieving expected lower bound of $(1 - o(1)) \cdot val(F)$ provided $val(F) > N^{-o(1)} = 2^{-o(n)}$.

In complexity and cryptography, many interesting *transformations* of interactive proofs are known, and these can usually be applied to general games against Nature as well, changing the structure of games while holding their value nearly fixed. Some such transformations are superficially promising as algorithmic tools for the tasks 1-2 above, but there is typically a "catch" obstructing their direct use. For example, Leshkowitz [Les18] has shown that general interactive proofs using $r(n)$ bits of randomness can have their round complexity reduced to $O(r(n)/\log n)$. But considered as a transformation on games, the blowup in bitlength of individual moves by the strategic player is apparently too large to be of help.

1.4 Our result

In this work, we show that indeed, high alternation in games against Nature can be powerfully exploited. We focus on the case of *fully alternating, binary* games against Nature, in which the players strictly alternate making 0/1-valued moves $(w_1, r_1, w_2, r_2, \ldots, w_{n/2}, r_{n/2})$, and the payoff to the non-random player is given by a general, $[0, 1]$-valued function F, provided in black-box or white-box form. We show:

Theorem 1 (Main, informal)**.** *For some absolute constant $\delta > 0$, there is a randomized algorithm to approximate the value of fully-alternating binary games against Nature under optimal play, which makes $2^{(1-\delta)n}$ queries to the input function F, and produces a lower bound $\hat{v}$ on the game value v which satisfies $\mathbb{E}[v - \hat{v}] \leq \exp(-\Omega(n))$.*

The algorithm explicitly witnesses its lower bounds $\hat{v}$ by exhibiting output values of F on particular sets of inputs that imply the lower bound. From these witnesses, a full description of a player strategy can be produced, or a next move given, also in time $2^{(1-\delta)n}$.

While we assume a specific move-structure on the games we study, we believe our algorithm and analysis contain powerful, flexible ideas that will have broader impact.

The boundedness of F is essential here; if its values can be huge, then approximating $val(F)$ to any reasonable additive error is as hard as needle-in-haystack search, which for (non-quantum) black-box algorithms admits no asymptotic speedup. Similarly, we cannot expect a speedup as above with *multiplicative* approximation guarantees on all inputs (outputting an estimate $\hat{v}$ where $\hat{v} = \Theta(v)$ with high probability, say), even if F is bounded.

For the fully-alternating case we study, the SampleEvalSSAT algorithm of [LMP01] yields no significant improvement over brute force, let alone one as strong as Theorem 1. We explain this point in Sec. 10 (with observations similar to ones in [LMP01]).

A downside to our algorithm is that the value $\delta > 0$ we obtain is tiny. We make no effort here to optimize it, and do not opine whether a "respectable" value can be achieved by something close to the current approach. Also, the algorithm given uses exponential space. Reducing space usage is a natural goal for future work.

Theorem 1 can be compared to a simple (and polynomial-space) recursive algorithm of Snir [Sni85], for which best constants in the recurrence-based analysis were obtained by Saks and Wigderson [SW86]. This algorithm computes the value of fully alternating, n-move, 2-player games against an adversarial opponent (and with $\{0,1\}$-valued payoff function F) using $O(2^{\alpha n})$ queries, with $\alpha = \log_2((1+\sqrt{33})/4) \approx .753$. The constant α here was shown to be *optimal* for black-box randomized algorithms (by [SW86] for "Las Vegas" zero-error algorithms, and by Santha [San95] for bounded-error ones).

In contrast, essentially the only black-box lower bound we know for our problem is a $\Theta(2^{n/2})$ bound for distinguishing values $[v = 0]$ from $[v = 1]$; such a lower bound can be proved easily by considering the problem of distinguishing the all-zero F from one of the functions $F_z(w, r) = \mathbf{1}[w = z]$, ranging over all possible $z \in \{0,1\}^{n/2}$. This lower-bound example, in which the random variables are irrelevant, basically relies again on the difficulty of black-box needle-in-haystack search.

Similarly, in the white-box setting, and allowing reasonably expressive classes of functions F, we cannot expect an algorithm of runtime $2^{(.5-\varepsilon)n}$ unless the Strong Exponential Time Hypothesis [IP01] fails; but this also leaves us quite uncertain of the true exponential complexity of the problem. We note that our algorithm, like that of [Sni85], has little direct relation to known improved exponential-time algorithms for, say, Satisfiability of CNF formulas, notably those of [MS85, PPZ99, Sch99], which cleverly exploit specific white-box structure in the input. (In the cited works, bounded clause-width is exploited. Unlike our problem, NP Satisfiability problems have no useful "black-box structure" per se for classical algorithms to exploit, since the corresponding unstructured query problem—computing the OR of N bits—requires $\Omega(N)$ queries for classical randomized algorithms.)

Thus, we still do not know whether fully-alternating binary games against Nature are *easier, or harder,* to approximately evaluate than their adversarial counterparts![1] In view of Theorem 1, however, we at least know that a speedup qualitatively similar to that of [Sni85,

[1] There is at least one sense in which playing against a random opponent is "easier"—namely, any strategy deployed against a random opponent has value at least as large as when played against an optimal adversarial opponent. But we are concerned with *taking full advantage* of the opponent's random strategy, in games whose payoff against an optimal opponent might be very poor. Also, the improved algorithm of [Sni85] can be applied to $\{0,1\}$-payoff games against Nature to determine whether their value is 1 or less than 1, but it does not approximate games against Natures' values or yield near-optimal play in cases when the game's value is less than 1.

SW86] is possible for the games we study. We believe this is a significant finding for the study of decision-making under certainty.

1.5 Our techniques

For binary, fully alternating games against Nature lasting n moves, the task of computing the value of a game specified by its payoff function F can be viewed equivalently as computing the value of a particular *real arithmetic formula*, alternating between layers of "Max" and "Average" gates. (Each gate is of fanin two, with the formula a full binary tree of depth $n = 2k$.) The output Max gate corresponds to the initial move by the strategic, maximizing player; its two input Average gates correspond to the next, random move by Nature, and so on. This alternating "Max/Average" formula, which we denote MA_k, is given input x equal to the *payoff function* F (with appropriate indexing), and outputs the value of the game associated with F.[2] In the body of our work, and in what follows, we will adopt this formula-evaluation perspective. MA_k is a *monotone* real formula, with inputs assumed to come from $[0, 1]$, so any subset of revealed input values naturally certifies a lower bound on the output value, by direct evaluation of the formula (replacing unseen values with 0). We use this basic idea throughout, and in particular, will use it to determine the final estimate output by our algorithm, which will therefore be a lower bound on the true value.

1.5.1 All-queries algorithms and recursive composition

To describe our algorithm for estimating values $\mathsf{MA}_k(x)$ with queries to x, we begin at a high level. While our goal is an algorithm making *few* queries, almost all of our work focuses on the design and study of *"all-queries"* algorithms, which probe every input-coordinate in some order, and whose partial views of x certify progressively better lower bounds on the "true" value $\mathsf{MA}_k(x)$. Our final goal is to design such algorithms which reach "good" lower bounds so quickly that we can simply halt the algorithm early. However, to bootstrap effectively toward this goal, we will actually need approximation guarantees that are more informative about the entire course of the algorithm's execution. Thus, we consider all-queries algorithms with associated *epoch markers* dividing up their execution, and we wish to establish approximation guarantees at the end of each epoch. For now, one may think of these guarantees as bounds, for all possible inputs x, on the expected gap $\mathbb{E}[v - \hat{v}^j]$ between the true value $v = \mathsf{MA}_k(x)$, and the lower bound $\hat{v}^j \le v$ certified by the partial view after the j^{th} epoch. (This idea will need to modified later.) We note too that the epoch markers will not be evenly spaced, but will be determined by j and k.

For a given, fixed k, an all-queries algorithm A_k for MA_k can naturally serve as a *subroutine* in the evaluation of MA_K-formulas on larger inputs $x \in [0, 1]^{4^K}$, for $K > k$, by an algorithm A'_K which applies A_k to each of the *subformulas* of height $2k$ and aggregates the results in some way. (To get strong results from the recursive use of this idea, we will want $K = k + O(1)$.)

[2] This follows from Eq. (1); we have $val(F) = \mathrm{Max}\{\mathrm{Avg}(val(F_{0,0}, F_{0,1})\ ,\ val(F_{1,0}, F_{1,1})\}$, where $F_{b,b'}$ is the restricted game in which $(w_1, r_1) = (b, b')$, and this expands further to give the full arithmetic formula.

Now let $x \in [0,1]^{4^K}$ denote the input to A'_K and let the height-$2k$ subtrees be indexed by $i \in [N] = [4^\ell]$, where $\ell = K - k$. Let $x^i \in [0,1]^{4^\ell}$ be the part of x input to the i^{th} such subtree.

The all-queries property of $A_k(x^i)$ helps $A'_K(x)$ to "squeeze all the juice" from each x^i, and to avoid accumulating losses at higher levels of a recursive evaluation scheme. But queries are expensive and making all queries to one x^i before making any to another is risky, since a given x^i may not contribute much (or at all) to the value of $\mathsf{MA}_K(x)$. Thus, our $A'_K(x)$ will hedge its bets and instead maintain *parallel simulations* of each $A_k(x^i)$. Its goal will be to intelligently *allocate* successive queries between different subroutine calls based on what it has seen so far. (These parallel simulations, multiplied recursively, will cause our final algorithm to use exponential space, which is a drawback to the approach.)

Since epochs are a central unit of analysis for our query algorithms, the outer algorithm $A'_K(x)$ will as its basic action always choose to *advance one sub-input x^i's subroutine $A_k(x^i)$, by one entire epoch.* Thus, the epochs we define also structure our recursive scheme itself. This scheme allows the outer algorithm to prioritize more "promising"-looking inputs x^i—as we will do, by a subtle and context-sensitive criterion. However, this prioritization is done conservatively and does not lead to runaway disparities in the number of queries allocated to the different x^i. In fact, the outer algorithm proceeds in "phases" where, after the end of each j^{th} phase, every $A_k(x^i)$ has advanced by j epochs—equalizing the query counts between each x^i. Moreover, in every phase, only a *single* adaptive decision is made by the outer algorithm about which subroutine to advance next! It is immediately after this adaptively-chosen query block that the outer algorithm hopes to enjoy a nontrivial approximation guarantee for its certified lower bound on $\mathsf{MA}_K(x)$. Thus it is this moment, midway through the j^{th} *phase* of $A'_K(x)$, which we select to mark the end of its j^{th} *epoch.* After a final end-marker is added, our scheme endows the execution of $A'_K(x)$ with $q+1$ epochs, if each $A_k(x^i)$ has q epochs.

The question then becomes how to ensure good approximation guarantees at the new epoch-markers for $A'_K(x)$. As the j^{th} marker for A'_K is placed within a transition between the $(j-1)^{st}$ and j^{th} epochs for the subroutines $A_k(x^i)$, it is natural to imagine that the new guarantee will be obtained as some kind of *weighted average* between the guarantees for the subroutines on those two epochs. And such a guarantee can certainly be attained, even non-adaptively. For example, if $A'_K(x)$ simply randomly selects some .75 fraction of the values i to advance the execution of $A_k(x^i)$ through its j^{th} epoch (and we end the j^{th} epoch of $A'_K(x)$ at this point), then it is not hard to show that A'_K will afterwards obey an expected approximation-error bound of $\leq .25\varepsilon_{j-1} + .75\varepsilon_j$, where ε_j is an expected-error bound assumed to hold for A_k after j epochs.

The strength of this new error bound for A'_K is, unfortunately, counterbalanced by the query cost of advancing a .75 fraction of the subroutines $A_k(x^i)$ by one epoch, and the recursive composition of this scheme does *not* lead to strong algorithms. But a tantalizing prospect appears: if we could achieve the same error bound as above, while advancing only a .74 fraction of the subroutines, then this approach would indeed achieve our main goal! (To show this, we relate the recurrences defining the epoch markers and the error bounds to *tail probabilities* of Bernoulli sums, and use Chernoff bounds to identify an epoch after which the error bound and the fraction of queries made are both small.) And this is essentially how we proceed—with two caveats. First, the actual averaging constants involved are different, and the useful numerical gap is far smaller than .01. The main cause of this is that our outer algorithm A'_K for MA_K will need to work with a very large value $\ell = K - k$ (but still $\ell = O(1)$) to make a single, smart adaptive choice, diluting its strength.

Second, to make the above plan succeed, we will actually need to replace error bounds $\mathbb{E}[v - \hat{v}^j] \le \varepsilon_j$ associated with epoch-markers, with *exponential-moment bounds* of the form

$$\mathbb{E}[\exp(s(v - \hat{v}^j))] \ \le \ e^{s\varepsilon_j} \ ,$$

for some $s > 0$. In fact we will need such bounds with particular s-values that are specific to j, and sufficiently large compared to the gap $\varepsilon_j - \varepsilon_{j-1}$ (the product of the two should exceed a large constant). These types of bounds imply powerful well-behavedness of the error when s is sufficiently large; and despite their form, these bounds are tractable to work with due to a basic independence property maintained between our subroutines.

Furthermore, the "averaging behavior" of the formula MA_ℓ will help the execution of $A'_K(x)$ to obey exponential-moment bounds at its epoch-markers with s-values significantly *larger* than those holding for the subroutines $A_k(x^i)$. This boost is important because the minimum values among the error gaps $\varepsilon_j - \varepsilon_{j-1}$ also become *smaller* at higher levels of our recursive scheme. In our whole approach it appears unavoidable to contend with such error gaps that are arbitrarily small compared to 4^ℓ (the number of inputs to our fixed-sized reference formula MA_ℓ)—even if our final goal was a more modest additive .1-approximation to the value of games against Nature. Thus it is somewhat surprising that our approach works at all, and in hindsight, the use of exponential-moment bounds with s-values large compared to the gaps is well-motivated as a countermeasure to this challenge.

There is an inherent tension in our use of exponential-moment bounds: while such bounds are stronger and can be more useful when s is large, it is only when s is somewhat *small* that these bounds can be effectively "aggregated" over different outcomes of a random variable to yield new and useful bounds.[3] We use two methods to cope with this. First, we use Jensen's inequality to convert our bounds from higher to lower s-values at appropriate points in our analysis.

Second, we use care in our timing of the adaptive choice of query block made by the outer algorithm $A'_K(x)$. We do so after advancing not a .75 fraction, but a $1 - \Theta(N^{-1/2})$ fraction, of the subroutines $A_k(x^i)$ by one epoch, where $N = 4^\ell$. Actually, it seems merely helpful in several respects that this fraction be large (intuitively, it means we have learned a great deal about the j^{th}-epoch increments on the executions of $A_k(x^i)$ for the various i, which helps us make a good adaptive choice). But it is apparently critical that the fraction not be *too* close to 1, because we cannot allow the minimum separation $\varepsilon_j - \varepsilon_{j-1}$ to decay too quickly with the recursion depth. This is because we are only able to grow the value s in our exponential-moment bounds by (at most) about a $\sqrt{N}$ factor per layer of recursion—which is related to the fact that MA_ℓ is only $N^{-.5}$-Lipschitz.

We believe our use of exponential-moment bounds is not just a technical detail, but actually helps to illuminate the behavior of our algorithm in generating progressively more-concentrated estimates at higher stages of recursion, and to explain how such concentration

[3] A bit more concretely: a weighted average such as $pe^{sa} + (1-p)e^{sb}$, with $a > b$, is sufficiently close to $e^{s(pa+(1-p)b)}$ for our purposes, provided $s(a - b)$ is somewhat small. Our simple tool here is Lemma 3.

helps it succeed. It may also provide useful guidance for future work in related contexts. (While exponential-moment bounds are widely used to prove concentration properties for the analysis of randomized algorithms, and of algorithms on random inputs [DP09], we are not aware of previous applications with strong similarity to our work.)

1.5.2 Steepeners

We now look more closely into the behavior of the "outer" algorithm $A'_K(x)$ above and how it makes its crucial adaptive choice in each j^{th} epoch of its operation. Recall that $A'_K(x)$ maintains parallel simulations of $A_k(x^i)$, an all-queries algorithm for MA_k, on each sub-input x^i to a height-$2k$ subtree. The queries of $A_k(x^i)$ during its first $(j-1)$ epochs together yield a certified lower bound on the value $z_i := \mathsf{MA}_k(x^i)$; call this lower bound $X_i \in [0,1]$, and let $Y_i \geq X_i$ be the corresponding lower bound after j epochs. Our outer algorithm uses fresh randomness in each simulation, so the pair (X_i, Y_i) is independent of all other such pairs. Moreover, each of $(z_i - X_i)$ and $(z_i - Y_i)$ will obey an exponential-moment bound by assumption.

We call triples (X, Y, z) of vectors of random variables obeying the above properties *ensembles*. To guide A'_K, we design a special type of all-queries algorithm for $\mathsf{MA}_\ell = \mathsf{MA}_{K-k}$ (for our large, but fixed $\ell = O(1)$) that enjoys a performance guarantee for *all* ensembles for which the exponent s is large enough (relative to the quantity $\varepsilon_j - \varepsilon_{j-1}$, as described earlier). This algorithm is given the values X at the outset as "baseline advice", and queries the N coordinates of Y one at a time, making a single adaptive query toward the end of its operation (as its t^{th} query for $t = N - \Theta(\sqrt{N})$, in fact). Its goal is to maximize the certified lower bound $\hat{v}$ for $\mathsf{MA}_\ell(Y)$ implied by the "baseline" values X, superimposed by the first t queries to Y. We call an algorithm achieving a nontrivial guarantee here (expressed as an exponential-moment bound on $(\mathsf{MA}_\ell(z) - \hat{v})$) a *steepener*—it "steepens" the rate of ascent toward the final value $\mathsf{MA}_\ell(Y)$ across its first t queries, compared to a naïve approach.

The value $\mathsf{MA}_\ell(z)$ is nicely characterized as $\mathsf{MA}_\ell(z) = \max_T \ \mathrm{Avg}_{i \in T}(z_i)$, taken over a natural family of subsets T of inputs which we call *M-trees*; these consist of $\sqrt{N}$ coordinates each, and in the games-against-Nature view, correspond to strategies.[4] In the design and analysis of our steepener algorithm, a central concern is whether our adaptive t^{th} query to Y is made inside T^*, an *optimal* M-tree for the input z (this subset is not known to the algorithm). After making $t-1$ queries to Y in a semi-random way, the t^{th} query is chosen from one of two candidates $\mathbf{i}, \mathbf{j}$. These are chosen in such a way that at most one can come from T^*. The algorithm inspects the surroundings of the $\mathbf{i}$ coordinate in the partial view of Y so far over the baseline X. It essentially looks for signs marking this coordinate as "special"—which can be very roughly interpreted as "likely to come from T^*"—in which case $\mathbf{i}$ is selected; otherwise $\mathbf{j}$ is chosen by default.[5]

Before discussing our specific decision rule, we give a brief upshot of its analysis. First, in the event where neither $\mathbf{i}$ nor $\mathbf{j}$ is in T^* (as occurs most often), we are actually "lucky" in that

[4] The above was basically observed in [LMP01, Pap85], and pointed out to me by Rahul Santhanam.

[5] There is some notional resemblance between our approach here and an improved randomized, zero-error query algorithm of Magniez et al. [MNS+16] for the (exact) Recursive Majority-of-3 problem, in that their algorithm forms "predictions" for the values of certain nodes and uses these to drive decisions. The details appear very different, however.

we have made sufficiently many queries to T^* already to get satisfactory bounds regardless of the adaptive decision's outcome. If instead $\mathbf{j} \in T^*$ (which we call the "$\mathbf{j}$-critical case"), we essentially show that most possible values for $\mathbf{i}$ will be rejected, causing $Y_{\mathbf{j}}$ to receive the adaptive query as desired. In the trickier "$\mathbf{i}$-critical case" ($\mathbf{i} \in T^*$), we show something weaker but still sufficient: in a "substantial bulk" of cases (under an appropriate exponential measure), it holds that *either* $Y_{\mathbf{i}}$ receives the adaptive query, *or* an appreciable portion of the gap $(z_{\mathbf{i}} - X_{\mathbf{i}})$ is not "felt" as a contribution to the gap $(\mathsf{MA}_\ell(z) - \hat{v})$, so that a failure to query $Y_{\mathbf{i}}$ would not be as harmful to the estimate $\hat{v}$ as it might naïvely appear. (We mention that our actual disjunctive analysis here in the $\mathbf{i}$-critical case seems to rely, for its needed quantitative strength, on the use of exponential-moment bounds. More specifically, these bounds' useful properties for our work, when s is large, are identified in Lemmas 5 and 6.)

As for the decision rule, we adaptively query $Y_{\mathbf{i}}$ if at least one of three "special" selection-conditions hold for the path $P_{\mathbf{i}}$ from the output gate to input $\mathbf{i}$, considered on the "hybrid" input u which superimposes the partial view of Y after $t-1$ queries upon the baseline X. The path $P_{\mathbf{i}}$ is called "dominant" for u at a Max gate g, if it passes through g to the larger-valued of its two input/child Avg gates. As a first selection-condition, if $P_{\mathbf{i}}$ is dominant on more than, say, a .51 fraction of its Max gates, then (as the input size N is large) this is a fairly strong sign that $\mathbf{i}$ is "special" and worthy of the adaptive t^{th} query.

To motivate a further selection-condition, we consider cases in which the previous one fails to apply. If $\mathbf{i} \in T^*$ but $P_{\mathbf{i}}$ does not have the ".51-dominant" property on u, let us consider the derived input $\tilde{u}$ in which $u_{\mathbf{i}} = X_{\mathbf{i}}$ is replaced with z_i. If $\mathbf{i}$ is not part of every optimal M-tree for input $\tilde{u}$, then a decrement of coordinate $\mathbf{i}$ from $z_{\mathbf{i}}$ down to $X_{\mathbf{i}}$ does not reduce the MA_ℓ-value, and in this case we can be content not to query $Y_{\mathbf{i}}$ for our adaptive t^{th} query.

On the other hand, if $\mathbf{i}$ *is* in each optimal M-tree for $\tilde{u}$, then $P_{\mathbf{i}}$ is dominant at each Max gate on $\tilde{u}$. To study this situation, we will define the "decisiveness" of a Max gate on a given input, as a certain "height-normalized" multiple of the gap between its two input values. (To build intuition: if a path P_i is dominant at each Max gate, then incrementing input i by θ will increase the decisiveness of each such gate by θ.) Then the aforementioned decrement of coordinate $\mathbf{i}$ can be seen to reduce the MA_ℓ-value only proportionally to ($N^{-.5}$ times) the *minimum* decisiveness of $P_{\mathbf{i}}$ on $\tilde{u}$. At the same time, the total decrement must be proportional to the *near-median* decisiveness along the same path, in order to destroy this path's dominance on a .49 fraction of these gates. If the gap between these two decisiveness values is large, then enough of the coordinate decrement is not felt as loss in MA_ℓ-value, and again we are content not to query $Y_{\mathbf{i}}$ for our adaptive t^{th} query.

Thus we are led to worry about the case where the minimum and near-median decisiveness along $P_{\mathbf{i}}$ on $\tilde{u}$ are near-equal. But overall, we need only worry if this happens for *sufficiently many* possible values of $\mathbf{i}$. Now strictly speaking, we need to clarify relative to which partial conditioning this holds, and exercise care since $\tilde{u}$ is defined relative to $\mathbf{i}$. Such concerns make the $\mathbf{i}$-critical case subtle, but we pass over details here and just suggest the main idea. We recenter our analysis on $\hat{u}$, which is $\tilde{u}$ with the $\mathbf{i}^{th}$ coordinate $z_{\mathbf{i}}$ replaced by $Y_{\mathbf{i}}$, and now consider $\mathbf{i}$ as undetermined (but we condition on $\mathbf{i} \in T^*$; in fact it could be almost any index in T^*). Focusing on our "worrisome" case sketched above, consider an outcome to the vectors (X, Y), determining $\hat{u}$, for which:

1. Most paths P_i with $i \in T^*$ are dominant for $\hat{u}$ at all Max gates; and,
2. Most such paths also have minimum decisiveness (on $\hat{u}$) nearly equal to their near-median decisiveness.

By an analysis of random walks from the root/output gate, and constrained to end at an input i with $i \in T^*$, we conclude that most such P_i also have many *"pendant"* Max gates (adjacent to P_i, as a sibling to one of its Max gates) whose decisiveness is also nearly equal to the minimum decisiveness on P_i itself.

Consider one such P_i for $\hat{u}$, and now imagine $\mathbf{i} = i$ is selected and Y_i is decremented down to X_i, yielding the input u seen by our steepener algorithm before the adaptive t^{th} query. If this decrement is significantly larger than the minimum decisiveness d_{min} for P_i on $\hat{u}$, then an appreciable portion of this decrement is not felt as loss in MA_ℓ-value, and our algorithm can be content not to adaptively query $\mathbf{i}$. But if the decrement is very nearly equal to P_i's minimum decisiveness, then one can show that the quantity $S_i^-(u)$, measuring the *sum* of decisiveness values of all Max gates on which P_i is *non*-dominant, is small compared to d_{min}, and thus also small compared to the decisiveness (for u) of many Max gates *pendant* to P_i—whose decisiveness values are, moreover, all nearly equal.

It is this type of observable property for $P_\mathbf{i}$ on u that we identify as our next "special" one, and use to define our second selection-condition (to make the adaptive t^{th} query to $Y_\mathbf{i}$). Crucially, we also show (by another, similar analysis of random walks on trees) that this selection-condition is unlikely to be met by $P_\mathbf{i}$ in the $\mathbf{j}$-critical case (where we should query $\mathbf{j} \in T^*$ instead), a case in which $\mathbf{i}$ is essentially uniform over $\approx N/2$ possible values.

A third selection-condition for querying $Y_\mathbf{i}$ concerns a relatively short initial segment of $P_\mathbf{i}$ on u starting from the root. The third condition is less central, but facilitates our random-walk-based analysis of the other two conditions. This completes our sketch description and motivation for the steepener algorithm we provide. We have glossed over some significant aspects of its analysis, but we believe the above represents the most important core ideas of our approach.

1.6 Organization of the paper

In Section 2, we give preliminaries including background and definitions for query algorithms and the MA_k function, and some useful probabilistic inequalities. We also define some of our key concepts, including ensembles, steepeners, epoch-based error bounds for algorithms, and our main method of epoch-based recursive composition.

In Section 3, we show that the existence of good steepeners, in combination with our recursive composition scheme, suffice to give the improved algorithm claimed in this work. Sections 4- 8 then show that such steepeners do exist.

In Section 4, we make a basic but careful study of MA_k formulas, and in particular, of the effect of decrementing a single input coordinate value. Section 4 also includes some lemmas on random walks in binary trees, that will be used to analyze the behavior of MA_k formulas.

In Section 5, we give our steepener candidate construction. In Section 6, we provide the high-level framework for its analysis.

Section 7 studies the "**j**-critical case" mentioned in the Introduction, and Section 8 is devoted to the **i**-critical case, completing the proof of the main result. Finally, Section 9 describes the aforementioned reduction from MA_k to the Longest Increasing Subsequence problem, and Section 10 discusses the limitations of the SampleEvalSSAT algorithm of [LMP01] for the fully-alternating case we study.

1.7 Acknowledgements

This work owes a great debt to Rahul Santhanam. Rahul provided generous support with a postdoc at U. Edinburgh (2014-15); made sustained joint efforts with me on the problem studied in the present work, and discussed extensions and variant problems [San]; has provided many, many helpful conversations; and has offered material and moral support throughout this research. I acknowledge support from his ERC Consolidator Grant (AL Unif) and from the University of Edinburgh, and travel support from Oxford University.

I also thank Laci Babai for helpful discussions about the exposition of games against Nature and interactive proofs.

2 Preliminaries and core concepts

We will often use $\exp(x)$ to denote e^x, and use either $\mathbf{1}_S$ or $\mathbf{1}[S]$ as the indicator random variable for an event S, e.g. $\mathbf{1}_{[X>.2]}$. If X is a finitely-supported random variable, we use $\mathrm{supp}(X) = \{x : \Pr[X = x] > 0\}$ to indicate its support.

2.1 The Max-Average function; M-trees

For any integer $k \geq 0$, we define the *Max-Average function* on $N := 4^k$ variables $x_i \in \mathbb{R}$, for $i \in [N]$. For $k = 0$ we have $N = 1$ and we simply take $\mathsf{MA}_0(x_1) := x_1$. Inductively, for $k > 0$ we break the input vector into four equal-sized parts, $x = (x^a, x^b, x^c, x^d)$, and define

$$\mathsf{MA}_k(x) \ := \ \mathrm{Max}\{\ \mathrm{Avg}(\mathsf{MA}_{k-1}(x^a), \mathsf{MA}_{k-1}(x^b))\ ,\ \mathrm{Avg}(\mathsf{MA}_{k-1}(x^c), \mathsf{MA}_{k-1}(x^d))\ \}\ ,$$

using the notation

$$\mathrm{Avg}(u, v) \ := \ .5(u + v)\ .$$

(In terms of indices, we use $x^a = (x_1, \ldots, x_{N/4}), \ldots, x^b = (x_{N/4+1}, \ldots, x_{N/2})$, and so on.) Thus, MA_k computes a real arithmetic formula with the shape of a full binary tree, composed of alternating layers of Max and Avg gates (k layers of each), with Avg gates closest to the inputs and a Max gate at the root.

This function is interesting for the special case of Boolean inputs, for which the real-valued output lies in $[0, 1]$. In this setting, the Max-Average function is a relative of the commonly-studied *balanced AND/OR function* on 4^k Boolean variables, which can defined similarly to the above but with min (or equivalently $\wedge$) in place of Avg. Our general study in this work will be of MA_k applied to inputs from $[0, 1]^N$.

We state some simple, useful properties of the Max/Avg function.

Proposition 1. *For $N = 4^k$ and random variables $X_1, \ldots, X_N$ (not necessarily independent), we have*

$$\mathbb{E}[\mathsf{MA}_k(X_1, \ldots, X_N)] \;\geq\; \mathsf{MA}_k(\mathbb{E}[X_1], \ldots, \mathbb{E}[X_N]) \;.$$

The proposition follows readily from the facts that $\mathbb{E}[\mathrm{Avg}(X, Y)] = \mathrm{Avg}(\mathbb{E}[X], \mathbb{E}[Y])$ and $\mathbb{E}[\mathrm{Max}(X, Y)] \geq \mathrm{Max}(\mathbb{E}[X], \mathbb{E}[Y])$.

Proposition 2. *For $N = 4^k, c \in \mathbb{R}$, and $x \in \mathbb{R}^N$, we have $\mathsf{MA}_k(x_1 + c, \ldots, x_N + c) = \mathsf{MA}_k(x) + c$.*

Let $N = 4^k$ with $k > 0$. Let $\mathbf{T}$ denote the full binary formula for MA_k composed of Max-gates, Avg-gates, and variable-gates $x_1, \ldots, x_N$; we also freely regard $\mathbf{T}$ as a directed graph with edges going from a gate g to each of its input gates (opposite to the flow of values in a computation; the input gates to g are regarded as its children in $\mathbf{T}$). We let

$$P_j \;=\; (p^{j,0}, p^{j,1}, \ldots, p^{j,2k}) \tag{2}$$

denote the path from the root $r = p^{j,0}$ (an output Max gate) to input gate $x_j = p^{j,2k}$. Each of the vertices $p^{j,0}, p^{j,2}, \ldots, p^{j,2k-2}$ along this path of even depth are Max gates (excepting the input gate $p^{j,2k} = x_j$), the odd-depth ones being Avg gates.

The next definition is essentially a special case of the "policies" studied in [LMP01], and used for a characterization similar to Prop. 4 below. (See also what are referred to as "admissible subtrees" in [Pap85]. I thank Rahul Santhanam for noting Def. 1 and Props. 3-4 below before learning of [LMP01].)

Definition 1. *An* M-tree *is a subset T of $[N]$, regarded as a subset of variable-gate indices, specifiable by a* ***selection function*** sel *giving, for each* Max *gate $g = \mathrm{Max}(h, h')$ in $\mathbf{T}$, a "selected child" $\mathrm{sel}(g) \in \{h, h'\}$. The associated $T = T_{\mathrm{sel}}$ is defined as the set of $i \in [N]$ for which every step along P_i from some* Max *gate g goes to $\mathrm{sel}(g)$.*

We also define, for T as above, the ***canonical partial selection function*** sel' *which agrees with* sel *on all* Max *gates g lying on a path P_i with $i \in T$, and is undefined elsewhere.*

Intuitively one might regard T as consisting of the union of paths P_i, ranging over $i \in T$; however, it is convenient to formally take T as just the variable-gate indices of these paths' endpoints. The selection function sel is not unique for a given T, since its values off of the union of P_i, $i \in T$, are irrelevant, but the canonical partial selection functions are in 1-to-1 correspondence with M-trees T.

The next Propositions are easily verified by induction on k.

Proposition 3. *Every M-tree is of size $\sqrt{N}$.*

Proposition 4. *For every $y \in \mathbb{R}^N$, we have*

$$\mathsf{MA}_k(y) \;=\; \max_T \;\; \mathrm{Avg}_{i \in T} \; (y_i) \;,$$

where T ranges over all M-trees of $[N]$ and where $\mathrm{Avg}_{i \in T}(y_i) = \frac{1}{|T|} \sum_{i \in T} y_i$.

If $\mathsf{MA}_k(y) = \mathrm{Avg}_{i \in T^*}(y_i)$, we say that the M-tree T^* is *optimal for y*. (There may be more than one such M-tree.)

2.2 Query algorithms and certified lower bounds

We presume familiarity with (deterministic and randomized) query algorithms applied to Boolean inputs. We will further consider query algorithms applied to *real-valued* inputs. Our basic model makes no restriction on the decision rules used to determine the next query (which in principle could be arbitrarily complex); however, the actual algorithms we describe will have fairly efficiently-computable rules.

The key subroutines we define, "steepeners", will be such query algorithms which also take advice, called "baseline advice", as defined below.

Definition 2 (Query algorithms with baseline advice)**.** *Fix $N = 4^\ell$. A* ***query algorithm with baseline advice*** *is a possibly-randomized algorithm $A = A^x(y)$ that is given unlimited access to a "baseline advice" vector $x \in [0,1]^N$ and query access to an unknown $y \in [0,1]^N$, with the promise $x \le y$.*

A is called an ***all-queries algorithm*** *(with baseline advice) if it always queries every coordinate of y, for all such pairs $x \le y$.*

(We will generally use an "ℓ" in the input size, $N = 4^\ell$, in contexts when the input is expected to be given to a "steepener"—a particular type of query algorithm with baseline advice which will act on a fixed-size input $4^\ell = O(1)$ for a suitable large constant ℓ.)

The progress of query algorithms computing a monotone function $f(y)$ with baseline advice x will be measured using the lower bounds *certified* by their partial view relative to the known baseline advice, as in the following definition.

Definition 3. *For $w \in \{[0,1] \cup \{*\}\}^N$, let*

$$[w \searrow 0^N] \;\in\; [0,1]^N$$

be w with all $*$ entries replaced by 0s. Now, suppose $f : [0,1]^N \to \mathbb{R}^{\ge 0}$ is a <u>*monotone*</u>*, nonnegative real-valued function (in our applications this will be MA_ℓ). Observe that $f([w \searrow 0^N])$ is a* lower bound *on $f(y)$ for all $y \in [0,1]^N$ agreeing with w on the non-$*$ entries of w. Moreover, it is the* largest possible *such lower bound. We call this the* ***lower bound certified for*** *f* ***by*** *w, and let*

$$LB_f(w) \;:=\; f([w \searrow 0^N])$$

denote this value.

Similarly, if w is as above and $x \in [0,1]^N$ satisfies $x_i \le w_i$ whenever $w_i \ne *$, then we let $[w \searrow x]$ be w with all entries $w_i = *$ replaced with the corresponding x_i. We let

$$LB_f(w; x) \;:=\; f([w \searrow x]) \;,$$

and note that $LB_f(w;x)$ is the largest possible lower bound on $f(y)$ valid for all $y \in [0,1]^N$ satisfying $y \ge x$ and agreeing with w on the non-$*$ entries of w. We call this the ***lower bound certified for*** *f* ***by*** *w* ***relative to baseline*** *x.*

2.3 Exponential-moment bounds

We use evaluations of the *moment-generating function* $M_U(s) := \mathbb{E}[\exp(sU)]$ of a random variable U; such functions are an important tool in probability theory. We will refer to this function's value on a particular s as the **s-exponential moment** of U, although this is non-standard. We will be interested in variables whose s-exponential moments are bounded:

Definition 4. *Let U be a nonnegative random variable, and $s > 0$. For $\beta \in \mathbb{R}$ we say that U is (s, β)-*small *if*

$$\mathbb{E}[\exp(sU)] \;\leq\; e^{s\beta} \;.$$

We say that U is $(+\infty, \beta)$-small if it is (s, β)-small for all finite $s > 0$. Finally, we say that U is $(0, \beta)$-small if $\mathbb{E}[U] \leq \beta$.

Proposition 5. *U is $(+\infty, \beta)$-small if and only if $\Pr[U \leq \beta] = 1$.*

Proof. If $\Pr[U \leq \beta] = 1$ then for any finite $s > 0$ we have $\exp(sU) \leq e^{s\beta}$, so U is (s, β)-small.

On the other hand, if $\Pr[U \leq \beta] < 1$ then there is some $\varepsilon \in (0, 1)$ such that $\Pr[U > \beta + \varepsilon] > \varepsilon$. Letting $s := 1/\varepsilon^2$, we have

$$\mathbb{E}[\exp(sU)] \;\geq\; \varepsilon \cdot \exp(s(\beta + \varepsilon)) \;=\; (\varepsilon \exp(1/\varepsilon)) \cdot e^{s\beta} \;>\; e^{s\beta} \;,$$

so U is not (s, β)-small. □

An "s-normalized" statement of the smallness condition (for finite $s > 0$) would equivalently say $\frac{1}{s} \ln\left(\mathbb{E}[\exp(sU)]\right) \leq \beta$. We have the following standard fact:

Proposition 6. *If $s \in (0, +\infty]$ and U is (s, β)-small, and if $0 < s' < s$, then U is (s', β)-small.*

Proof. The $s = +\infty$ case holds by definition. Now suppose s is finite. We have

$$e^{s\beta} \;\geq\; \mathbb{E}[\exp(sU)] \;=\; \mathbb{E}\left[(\exp(s'U))^{s/s'}\right] \;\geq\; \mathbb{E}\left[\exp(s'U)\right]^{s/s'} \;,$$

using Jensen's inequality. Raising the (positive) left- and right-hand sides above to the s'/s power yields the desired result. □

This also justifies our definition of $(+\infty, \beta)$-smallness. Exponent values s under discussion will henceforth be assumed finite unless explicitly noted. Also, we defined the $s = 0$ case of smallness only for comparison's sake, to indicate a relation between ordinary expectations and exponential moments. One can show by a similar use of Jensen's inequality that (s, β)-smallness for $s > 0$ implies $(0, \beta)$-smallness. On the other hand, for *bounded* $U \in [-1, 1]$ and for $s \in (0, 1)$, in the requirement $\frac{1}{s} \ln \mathbb{E}[\exp(sU)] \leq \beta$ the left-hand side equals

$$s^{-1} \ln\left(\mathbb{E}[1 + sU + O(s^2)]\right) \;=\; s^{-1} \ln(1 + s\mathbb{E}[U] + O(s^2)) \;=\; \mathbb{E}[U] + O(s) \;,$$

which approaches $\mathbb{E}[U]$ as $s \to 0$. Thus the $s = 0$ case of the definition "fits in" fairly naturally, at least in the bounded case which is our focus.

2.4 Some probabilistic inequalities

Here we collect some facts about random variables for later use, including the following forms of the Chernoff bound.

Lemma 1. *If $B_1, \ldots, B_n$ are Bernoulli random variables with expectation $p \in [0,1]$, then for $\delta > 0$,*

1. $\Pr[\ B_1 + \ldots + B_n \geq (1+\delta)pn\ ] \ \leq \ [e^{\delta}/(1+\delta)^{1+\delta}]^{pn}$,
2. $\Pr[\ B_1 + \ldots + B_n \leq (1-\delta)pn\ ] \ \leq \ [e^{-\delta}/(1-\delta)^{1-\delta}]^{pn}$.

Our next lemma gives conditions to bound the expected product of two random variables.

Lemma 2. *Suppose A, B are nonnegative-valued random variables, that W is an event, and let $\mathbf{t}$ be a random variable over a finite set $\mathfrak{T}$, such that $\mathbf{t}$ is independent of W and $\mathbb{E}[AB|W \wedge \mathbf{t} = \tau] = \mathbb{E}[A|W \wedge \tau] \cdot \mathbb{E}[B|W \wedge \tau]$ for any outcome $\tau \in \mathrm{supp}(\mathbf{t})$. Assume $\mathbb{E}[A|W] \leq a$ and that $\mathbb{E}[B|W \wedge \tau] \leq b$ for any $\tau \in \mathrm{supp}(\mathbf{t})$. Let $b' \in [0, b)$ and $p \in [0,1)$ be given.*

Let $S \subset \mathrm{supp}(\mathbf{t})$ be a subset of values such that $\mathbb{E}[B|W \wedge \mathbf{t} = \tau] \leq b'$ for any $\tau \notin S$. Also suppose that $\mathbb{E}[\mathbf{1}_{[\tau \in S]} \cdot A|W] \leq pa$. Then, $\mathbb{E}[AB|W] \leq a[pb + (1-p)b']$.

Proof. Implicitly restricting our sums to within $\mathrm{supp}(\mathbf{T})$, and letting expectations be over $\tau \sim \mathbf{t}$ (a distribution unaffected by conditioning on W), we have

$$
\begin{aligned}
\mathbb{E}[AB|W] &= \sum_{\tau} \Pr[\mathbf{t} = \tau] \cdot \mathbb{E}[AB|W \wedge \tau] \\
&= \left(\sum_{\tau \in S} \Pr[\mathbf{t} = \tau] \cdot \mathbb{E}[AB|W \wedge \tau]\right) + \left(\sum_{\tau \notin S} \Pr[\mathbf{t} = \tau] \cdot \mathbb{E}[AB|W \wedge \tau]\right) \\
&= \left(\sum_{\tau \in S} \Pr[\mathbf{t} = \tau] \cdot \mathbb{E}[A|W \wedge \tau] \cdot \mathbb{E}[B|W \wedge \tau]\right) + \left(\sum_{\tau \notin S} \Pr[\mathbf{t} = \tau] \cdot \mathbb{E}[A|W \wedge \tau] \cdot \mathbb{E}[B|W \wedge \tau]\right) \\
&\leq \left(\sum_{\tau \in S} \Pr[\mathbf{t} = \tau] \cdot \mathbb{E}[A|W \wedge \tau] \cdot b\right) + \left(\sum_{\tau \notin S} \Pr[\mathbf{t} = \tau] \cdot \mathbb{E}[A|W \wedge \tau] \cdot b'\right) \\
&= b \cdot \mathbb{E}[\mathbf{1}_{[\tau \in S]} \cdot A|W] + b' \cdot \mathbb{E}[\mathbf{1}_{[\tau \notin S]} \cdot A|W] \\
&= b \cdot \mathbb{E}[\mathbf{1}_{[\tau \in S]} \cdot A|W] + b' \cdot (\mathbb{E}[A|W] - \mathbb{E}[\mathbf{1}_{[\tau \in S]} \cdot A|W]) \\
&\leq b(pa) + b'(1-p)a \\
&= a[pb + (1-p)b']\ ,
\end{aligned}
$$

the last inequality using $b > b'$ and $\mathbb{E}[A|W] \leq a$. □

The remaining facts in this section relate to exponential-moment bounds. We do not attempt to optimize constants involved beyond our direct needs. The first lemma will help us to bound moments of a *mixture* of random variables for which individual moment bounds are known.

Lemma 3. *Suppose $x < y$ and that $y - x \leq .1$. Then, for $q \in [0,1]$, we have*

$$qe^y + (1-q)e^x \;\leq\; \exp\left[1.1qy + (1-1.1q)x\right] \;.$$

This result's strength depends on the bounds e^x, e^y being relatively close to each other, which in the application will mean that the exponent s in our smallness conditions must be sufficiently *small* (relative to other quantities) before the Lemma can be used.

Proof of Lemma 3. For $t \in [-.1, .1]$ we have $e^t \leq 1 + t + t^2$. Also, $e^t \geq 1 + t$ for all t. Using these relations, and letting $\varepsilon := y - x \leq .1$, we have

$$\begin{aligned} qe^y + (1-q)e^x &= e^x[(1-q) + qe^{y-x}] \\ &\leq e^x[(1-q) + q(1+\varepsilon+\varepsilon^2)] \\ &= e^x[1 + q\varepsilon(1+\varepsilon)] \\ &\leq e^x[1 + (1.1q)\varepsilon] \\ &\leq e^x e^{1.1q(y-x)} \\ &= \exp[1.1qx + (1-1.1q)x + 1.1q(y-x)] \\ &= \exp\left[1.1qy + (1-1.1q)x\right] \;. \end{aligned}$$

□

The property of (s,β)-smallness of a random variable U imposes stronger control on U when s is larger. The next lemma extends the message of Prop. 6, by saying that exponential-moment bounds with respect to a higher exponent implies some measure-concentration behavior for a lower exponent.

Lemma 4. *If $U \geq 0$ is (s,β)-small for some finite $s > 0$ and $C > 1$, and if $\rho > 0$, then*

$$\mathbb{E}\left[\mathbf{1}_{[U>\beta+\rho]} \cdot \exp((s/C)U)\right] \;\leq\; e^{(s/C)(\beta-(C-1)\rho)} \;.$$

Proof. Using our smallness assumption, we have

$$\begin{aligned} e^{s\beta} &\geq \mathbb{E}[\exp(sU)] \\ &\geq \mathbb{E}\left[\mathbf{1}_{[U>\beta+\rho]} \cdot \exp((s/C)U) \cdot \exp((1-1/C)s(\beta+\rho))\right] \\ &= \exp((1-1/C)s(\beta+\rho)) \cdot \mathbb{E}\left[\mathbf{1}_{[U>\beta+\rho]} \cdot \exp((s/C)U)\right] \;, \end{aligned}$$

and then dividing both sides by $\exp((1-1/C)s(\beta+\rho))$ gives the result. □

Next, we use Lemma 4 to get a related statement about a *family* of small random variables. The use of Lemma 5 below is one of two tools where we seem to rely on exponential-moment bounds (with s sufficiently large), as opposed to plain expected values.

Lemma 5. *Let $m > 10^6$. Suppose $U_1, \ldots, U_m \geq 0$ are independent random variables, where each U_i is (s,β)-small, for some $s \geq 1000$. More strongly, we assume $s\Delta \geq 1000$ for some $\Delta \in (0,1]$. Define H_i as the indicator variable for the event $[U_i > \beta + .06\Delta]$, and $H := \sum_i H_i$. Then,*

$$\mathbb{E}\left[\mathbf{1}_{[H\geq .02m]} \cdot \exp\left(10^{-4}s\left(\sum_i U_i\right)\right)\right] \;\leq\; \exp((10^{-4}s)(\beta - 5.88\Delta)m) \;.$$

Proof. Take $\rho := .06\Delta \leq .06$ and $\nu := .02$. We have

$$\mathbf{1}_{[H \geq \nu m]} \;\leq\; \sum_{S \subseteq [m], |S| = \binom{m}{\lfloor \nu m \rfloor}} \; \prod_{i \in S} H_i \; . \tag{3}$$

Now fix any S as above. Using independence of the U_i and regrouping, we have

$$\begin{aligned}
\mathbb{E}\left[\left(\prod_{i \in S} H_i\right) \cdot \exp\left(10^{-4} s \left(\sum_i U_i\right)\right)\right] \;&=\; \left(\prod_{i \in S} \mathbb{E}[\exp(10^{-4} s U_i) \cdot H_i]\right) \cdot \left(\prod_{j \notin S} \mathbb{E}[\exp(10^{-4} s U_i)]\right) \\
&\leq\; \exp(10^{-4} s(\beta - 999\rho))^{|S|} \cdot \exp(10^{-4} s\beta)^{m - |S|} \\
&=\; \exp[m(10^{-4}s)\beta - 999\rho|S|] \\
&\leq\; \exp[m(10^{-4}s)\beta - 999\rho(\nu m - 1)] \\
&\leq\; \exp[(10^{-4}s)(\beta - 998\nu\rho)m] \; ,
\end{aligned}$$

the first inequality by Lemma 4 (with $C = 10^4$) applied to the variables U_i with $i \in S$, and the last inequality using $\nu m > 10^4$. Then, using Eq. (3) and linearity of expectation, we have

$$\mathbb{E}\left[\mathbf{1}_{[H \geq \nu m]} \cdot \exp\left(10^{-4} s \left(\sum_i U_i\right)\right)\right] \;\leq\; \sum_{S \subseteq [m], |S| = \binom{m}{\lfloor \nu m \rfloor}} \exp((10^{-4}s)(\beta - 998\nu\rho)m) \; .$$

Applying a standard estimate,

$$\binom{m}{\lfloor \nu m \rfloor} \;\leq\; \left(\frac{em}{\lfloor \nu m \rfloor}\right)^{\lfloor \nu m \rfloor} \;\leq\; (1.1e/\nu)^{\nu m} \;=\; \exp\left(\nu \ln(1.1e/\nu) m\right)$$

(where we again used $\nu m > 10^4$). Applying this, we have

$$\binom{m}{\lfloor \nu m \rfloor} \cdot \exp[(10^{-4}s)(\beta - 998\nu\rho)m] \;\leq\; \exp[(10^{-4}s)(\beta - 998\nu\rho)m + \nu \ln(1.1e/\nu)m]$$

which is at most $\exp((10^{-4}s)(\beta - 98\rho)m) = \exp((10^{-4}s)(\beta - 5.88\Delta)m)$ as claimed in the Lemma, *provided that*

$$\nu \ln(1.1e/\nu)m \;\leq^?\; 900\nu\rho(10^{-4}s)m \; .$$

Cancelling and plugging in our chosen values, this is equivalent to

$$\ln(55e) \;\leq^?\; 900(.06 \cdot 10^{-4})(s\Delta) \; ,$$

and as $s\Delta \geq 1000$, this inequality holds true. ∎

The next (fairly specific and application-tailored) lemma studies a quantity $U + W - R$, in which the nonnegative random variable R can be lower-bounded when $U + W$ is either "too big" or "too small". We show that then $U + W - R$ will obey an exponential-moment bound noticeably smaller than that of $U + W$. The lemma requires that the exponent s, in initial assumed smallness bounds for U and $U + W$, should be sufficiently large relative to other values. This is our second core motivation for using exponential-moment bounds.

Lemma 6 (U/W-savings lemma)**.** *Suppose nonnegative random variables $U, W \geq 0$ are given, where U is (β, s)-small and $U + W$ is (α, s)-small, for some (α, β, s) with $s > 0$ and $\alpha > \beta \geq 0$. Let $\Delta := \alpha - \beta$.*

Let $S, r > 0$ be given. Define (nonnegative) random variables

$$R_1 \;:=\; \mathbf{1}_{[U+W<S+r]} \cdot W \;, \qquad R_2 \;:=\; \mathbf{1}_{[U+W \geq S]} \cdot (U + W - S) \;.$$

Let us further assume that:

A1. $s\Delta \geq 1000$*,*

A2. $r \geq .95\Delta$*.*

Then the random variable

$$Q \;:=\; U + W - \mathrm{Max}(R_1, R_2)$$

is $(.01s, \alpha - .93\Delta)$-small.

Proof. We always have $U + W - R_2 \leq S$, so Q is certainly $(.01s, S)$-small. If $S \leq \alpha - .93\Delta$, then we are done. So assume $S > \alpha - .93\Delta$, and thus (by assumption A2) $S + r > \alpha + .02\Delta$.

We bound $U + W - R_1$ using the expression

$$\begin{aligned} U + W - R_1 \;&=\; \left(\mathbf{1}_{[U+W<S+r]} \;+\; \mathbf{1}_{[U+W\geq S+r]}\right)(U + W - R_1) \\ &\leq\; \mathbf{1}_{[U+W<S+r]} \cdot U \;+\; \mathbf{1}_{[U+W\geq S+r]} \cdot (U + W) \\ &\leq\; U \;+\; \mathbf{1}_{[U+W\geq \alpha+.02\Delta]} \cdot (U + W) \;. \end{aligned} \tag{4}$$

where in the first inequality we applied the definition of R_1. From Eq. (4), and using that U is (s, β)-small (hence also $(.01s, \beta)$-small, by Prop. 6), we have

$$\mathbb{E}[\exp(.01sQ)] \;\leq\; \mathbb{E}[\exp(.01s(U+W-R_1))] \;\leq\; e^{.01s\beta} + \mathbb{E}[\mathbf{1}_{[U+W\geq\alpha+.02\Delta]} \cdot \exp(.01s(U+W))] \;. \tag{5}$$

By Lemma 4 applied to the (s, α)-small random variable $U + W$ (with $C = 100$, and with α in place of β in that Lemma and $\rho := .02\Delta$), we have

$$\mathbb{E}[\mathbf{1}_{[U+W\geq\alpha+.02\Delta]} \cdot \exp(.01s(U + W))] \;\leq\; e^{.01s(\alpha-99(.02\Delta))} \;\leq\; e^{.01s\beta} \;.$$

Combining this with Eq. (5), and using assumption A1 and $\ln 2 < .7 \leq .01s \cdot (.07\Delta)$, we have

$$\mathbb{E}[\exp(.01sQ)] \;\leq\; 2 \cdot e^{.01s\beta} \;\leq\; e^{(.01s)(.07\Delta)} \cdot e^{(.01s)\beta} \;=\; e^{.01s(\alpha-.93\Delta)} \;,$$

and again Q is $(.01s, \alpha - .93\Delta)$-small. □

2.5 Ensembles

The following definition will be a main object of study. We consider particular triples of related inputs (X, Y, z) to MA_ℓ:

Definition 5 (Ensembles)**.** *Fix $N = 4^\ell$ and values $0 \leq \beta < \alpha \leq 1$, as well as a value $s > 0$, which may be $+\infty$. An (α, β, s)*-ensemble *over N input variables is a tuple (X, Y, z), where:*

1. $z \in [0,1]^N$ *is a fixed vector;*

2. (X, Y) *are a pair of vector-valued random variables, each with support in* $[0,1]^N$*, and obeying* $X \le Y \le z$ *coordinate-wise;*

3. *each coordinate tuple* (X_i, Y_i) *takes finitely many possible values, and is independent of* $\{(X_j, Y_j)\}_{j \neq i}$*, although* X_i *need not be independent of* Y_i*;*

4. *for each such tuple,* $(z_i - X_i)$ *is* (s, α)*-small and* $(z_i - Y_i)$ *is* (s, β)*-small. That is (for the case where* $s < \infty$*),*

$$\mathbb{E}[\exp(s(z_i - X_i))] \ \le \ e^{s\alpha} \qquad \textit{and} \qquad \mathbb{E}[\exp(s(z_i - Y_i))] \ \le \ e^{s\beta} \ .$$

Item 4 constrains X_i and Y_i to be "typically close" to z_i, which in our application is regarded as a "true" value for the i^{th} coordinate.

2.6 Steepeners

In this section we describe our key type of "base case" algorithm (using baseline advice) for approximating MA_ℓ with useful guarantees; we call such an algorithm a "steepener".

Definition 6 (Steepeners)**.** *Assume* $N = 4^\ell \ge 10^{21}$*, and let* $\mathfrak{e} := 2^{30}\sqrt{N} + 1 < N/2$*. Consider an* <u>*all-queries*</u> *algorithm* $A = A^x(y)$ *with baseline advice for a* <u>*fixed*</u> *input size* $N = 4^\ell$*. Assume that* $A^x(y)$*, given advice string* $x \le y$*, makes* $(N - \mathfrak{e})$ *queries to* y*, yielding a partial view* $y^* \in \{[0,1] \cup \{*\}\}^N$*, before making the remaining* $\mathfrak{e}$ *queries.*

Suppose $\alpha > \beta \ge 0$ *and* $s, s' \in (0, +\infty)$*, and* $c \in (0,1)$*.*

We say that A *as above is a* c***-steepener, with respect to the 4-tuple*** (α, β, s, s') *if, for every* (α, β, s)*-ensemble* (X, Y, z)*, when the advice/input pair are generated as* $(x, y) \sim (X, Y)$ *then we have the* s'*-exponential-moment bound*

$$\mathbb{E}[\ \exp(s'(\mathsf{MA}_\ell(z) - LB_{\mathsf{MA}_\ell}(y^*; X))\ ] \ \le \ \ \exp\left(s'\left[\frac{\mathfrak{e} - c}{N} \cdot \alpha + \frac{N - \mathfrak{e} + c}{N} \cdot \beta\right]\right) \ . \qquad (6)$$

Here the expectation is over the randomness in (X, Y) *and any random choices by* A*.*

Thus the algorithm makes all N queries to Y (which is useful in our application), but its steepener property only concerns the partial view after $N - \mathfrak{e}$ queries. We will exhibit steepeners for certain 4-tuples featuring a large *exponent boost*, $s' = 10^{-4}\sqrt{N} \cdot s \gg s$, which will be a key source of progress.

To gain familiarity with the definition, note that a c-steepener for 4-tuple $(\alpha, \beta, s_1, s_1')$ is automatically one for $(\alpha, \beta, s_2, s_2')$ if $s_2 \ge s_1$ and $s_2' \le s_1'$. This follows directly from Proposition 6—the set of ensembles for which the algorithm must succeed only shrinks, *and* the definition of "success" for a given ensemble becomes more lenient.

Def. 6 makes sense for other values of $\mathfrak{e}$, but we will only construct a steepener for $\mathfrak{e} = 2^{30}\sqrt{N} + 1$ (showing that a single construction succeeds for various tuples (α, β, s, s')). This value of $\mathfrak{e}$ is chosen so that $\mathfrak{e} \ll N$ yet also $\frac{\mathfrak{e} - c}{N} \cdot \alpha \gg \alpha/\sqrt{N}$. In our recursive application of the steepener, the latter constraint helps us to avoid facing ensemble parameters (α, β, s) for which our steepener construction may not succeed.

Because of the large constant in our choice of $\mathfrak{c}$ (which still needs to be smaller than N), N needs to be very large; and for other reasons[6], we will take a still-larger value, $N = 4^{10^{10}}$. This certainly weakens the running-time bound in the final application.

2.7 Defect and epochs of query algorithms

The following definitions help us study the incremental progress of query algorithms in computing certified lower bounds for the Max-Average function MA_k.

Definition 7. *Fix $N = 4^k$, and let $A = A(y)$ be a randomized query algorithm on a length-N input $y \in [0,1]^N$.*

- *For $t \in [0, N]$ and input y, let $u^t = \mathrm{view}(y; t) \in \{[0,1] \cup \{*\}\}^N$ be the (random) string describing the partial view of y gained by A after t queries. (If A makes only $t' < t$ queries, then we take $u^t := u^{t'}$).*
- *Again for fixed t, y as above, we define the* ***accrued value after*** t ***steps*** *(with respect to MA_k), a random variable denoted Val_t, as*

$$\mathrm{Val}_t \ := \ LB_{\mathsf{MA}_k}(u^t) \ \in \ [0,1] \ ,$$

the lower-bound certified by A's view after t queries have been made. We also define the ***defect after*** t ***steps*** *by*

$$\mathrm{Def}_t \ := \ \mathsf{MA}_k(y) - \mathrm{Val}_t \ .$$

Note that these form a non-increasing sequence, with $\mathrm{Def}_0 = \mathsf{MA}_k(y)$.

Definition 8 (Epochs and defect bounds)**.** *Let A be as in the previous definition.*

- *If A always (on all inputs and random choices) makes all N queries (in some order), we call A an* ***all-queries algorithm****, similarly to Def. 2 but this time without baseline advice.*
- *If $\mathcal{T} = (t_0, \ldots, t_m)$ with*

$$0 \ = \ t_0 \ < \ t_1 \ < \ \ldots \ < t_m \ = \ N$$

are a collection of values (for some $m \geq 1$), we refer to the values $t_0, \ldots, t_m$ as ***epoch-markers****, and for $j \in [m]$ we regard the $(t_{j-1}+1)^{th}$ through $(t_j)^{th}$ queries made by the algorithm as belonging to the j^{th}* ***epoch*** *with respect to $\mathcal{T}$. (Thus the $m+1$ markers define m epochs.)*

[6](arising from the need for concentration properties of random walks along the formula for MA_ℓ; see Lemmas 16 and 22)

- *Suppose $s_0, \ldots, s_m > 0$ are given (here allowing the value $+\infty$ to appear), and*

$$\mathcal{P} = \{(t_0, s_0, \varepsilon_0),\ \ldots,\ (t_m, s_m, \varepsilon_m)\}$$

*are a collection of triples $(t_j, s_j, \varepsilon_j)$ with $\mathcal{T} = (t_0, \ldots, t_m)$ a valid set of epoch-markers, and that each $\varepsilon_j \in [0,1]$. We say that A **obeys the defect bounds** $\mathcal{P}$ (with respect to MA_k) if for each $j \in [0,m]$ and all inputs $y \in [0,1]^N$, the random variable*

$$\mathrm{Def}_{t_j}$$

(defined with respect to y, over the randomness in the execution of $A(y)$) is (s_j, ε_j)-small as in Def. 4.

2.8 Epoch-based recursive composition of algorithms

Definition 9 (Composed algorithms)**.** *The form of recursive composition we will use applies within the following setting. Let us fix:*

- *Values $N_1 = 4^\ell,\ N_2 = 4^k$, and $m \in [1, N_2]$;*
- *An all-queries algorithm $A_1 = A_1^x(y)$ with baseline advice x and input $y \geq x$, both vectors in $[0,1]^{N_1}$;*
- *An all-queries algorithm $A_2(w)$ making queries to $w \in [0,1]^{N_2}$ (both A_1 and A_2 may be randomized);*
- *A collection $\mathcal{T} = (t_0, \ldots, t_{m-1}, t_m)$, satisfying*

$$0 \ = \ t_0 \ < \ t_1 \ < \ \ldots \ < t_m \ = \ N_2 \ ,$$

of epoch-markers for the N_2-query algorithm A_2.

*We then define the **composed query algorithm** $A' = \mathrm{comp}(A_1, A_2; \mathcal{T})$, an all-queries algorithm working on input $z \in [0,1]^{N_1 \cdot N_2}$, as follows. We first describe the high-level framework, then fully specify the algorithm.*

- *First, we regard z as having the indexing $z = (z^1, \ldots, z^{N_1})$, where each $z^i \in [0,1]^{N_2}$.*
- *A' maintains parallel simulations of the executions of $A_2(z^i)$, for each $i \in [N_1]$. Each such simulation uses an independent source of randomness. Letting $U^t \in \{[0,1] \cup *\}^{N_1 \cdot N_2}$ denote the overall partial view of A' after t queries ($t < N_1 N_2$), the algorithm chooses a **next-query target group** index $j = j(t) \in [N_1]$ as a function of t and U^t.*
- *The simulation of $A_2(z^j)$ is then advanced by a single query; we also describe this as allocating a query to $A_2(z^j)$. (Our rule will only choose the index j if z^j has not yet been fully queried in the simulation.)*
- *This process continues for $N_1 N_2$ steps, after which all of z has been queried.*

It remains to give the selection rule for the next-query target group based upon (t, U^t). The query indices $t \in [N_1 N_2]$ are divided into m ***Phases***,[7] *where Phase $r \in [1, m]$ consists of the queries whose index t lies in $[t_{r-1}N_1 + 1, t_r N_1]$.*

Phase r in turn consists of N_1 ***sub-Phases***, *each consisting of $(t_r - t_{r-1})$ queries. Phase r is described in the box below.*

Phase r of algorithm $A' = \mathrm{comp}(A_1, A_2; \mathcal{T})$:

// Precondition: each of the simulations of $A_2(z^1), \ldots, A_2(z^{N_1})$ have been allocated exactly // t_{r-1} queries so far, yielding some partial view $v^i \in \{[0,1] \cup \{*\}\}^{N_2}$ of z^i for each $i \in [N_1]$.

1. For each $i \in [N_1]$, let $x_i := LB_{\mathsf{MA}_k}(v^i)$;

   // $x_i \in [0, 1]$ is computable from the algorithm's partial view at the outset of Phase r

2. Simulate the complete execution of query algorithm $A_1^x(y)$, taking baseline advice x as defined above, and with values $y \in [0,1]^{N_1}$ computed upon request in the course of the simulation, as follows:

   - If the simulation of $A_1^x(y)$ chooses to query y_i for position $i \in [N_1]$, advance the simulation of $A_2(z^i)$ by $(t_r - t_{r-1})$ queries, yielding a new partial view w^i of z^i; and let $y_i := LB_{\mathsf{MA}_k}(w^i)$.

     // Each such block of $(t_r - t_{r-1})$ queries forms a sub-Phase, of which there are N_1 in Phase r;

     // In each such sub-Phase, the selected index $i \in [N]$ is chosen as the next-query target group $(t_r - t_{r-1})$ times in succession;

     // After Step 2, each z^i has received t_r queries, extending the Precondition.

Definition 10 ("Near-endpoint" epoch-markers for composed algorithms)**.** *Assume $N_1 \geq 10^{21}$. Let $A' = \mathrm{comp}(A_1, A_2; \mathcal{T})$ be a composed algorithm as in Def. 9, an all-queries algorithm on input size $N_1 N_2$, with $\mathcal{T} = (t_0, \ldots, t_{m-1}, t_m)$ a collection of epoch-markers in $[0, N_2]$ for A_2. Let $\mathfrak{e} := 2^{30}\sqrt{N_1} + 1 < N_1/2$.*

Define a collection $\mathcal{T}' := (t'_0, \ldots, t'_m, t'_{m+1})$ of epoch-markers in $[0, N_1 N_2]$ for A' by letting $t'_0 = 0$, $t'_{m+1} := N_1 N_2$, and for $r \in [1, m]$, let

$$t'_r \; := \; t_{r-1}N_1 + (N_1 - \mathfrak{e})(t_r - t_{r-1}) \; = \; (N_1 - \mathfrak{e})t_r + \mathfrak{e} \cdot t_{r-1} \; . \tag{7}$$

Thus if A_2 is equipped (by $\mathcal{T}$) with m epochs, then $\mathcal{T}'$ equips A' with $m+1$ epochs. Note that for $r \in [1, m]$, the $(t'_r)^{th}$ query of A' finishes the $(N_1 - \mathfrak{e})^{th}$ sub-Phase of Phase r, at which point all but $\mathfrak{e}$ of the sub-inputs $z^1, \ldots, z^{N_1}$ have received a block of $(t_r - t_{r-1})$ queries in that Phase.

[7] While we will define epochs to analyze our recursively composed algorithms (Def. 10), the chosen epoch-markers will *not* coincide with the endpoints of these "Phases"—hence the use of a distinct term here.

3 Fast approximation for MA_k from steepeners

3.1 Defect bounds for composition with a steepener

Lemma 7. *Let $A' = \mathrm{comp}(A_1, A_2; \mathcal{T})$ be as in Definition 9, for $N_1 = 4^\ell$, $N_2 = 4^k$; here $\mathcal{T} = (t_0, \ldots, t_m)$ for some $m \in [1, N_2]$. Let $s_0, \ldots, s_m > 0$ be given, with $s_0 = s_m = +\infty$ and with s_r finite for any $1 < r < m$; along with finite values $\tilde{s}_1, \ldots, \tilde{s}_m > 0$ satisfying*

$$\tilde{s}_r \ \leq \ \min(s_{r-1} \ , \ s_r) \ , \qquad \textit{for} \ \ r \ \in [1, m] \ . \tag{8}$$

- *Suppose that A_2 (with respect to MA_k) obeys the defect bounds*

  $$\mathcal{P} \ = \ \{(t_0, s_0, \varepsilon_0), \ (t_1, s_1, \varepsilon_1), \ \ldots, \ (t_m, s_m, \varepsilon_m)\}$$

  with epoch-markers from $\mathcal{T}$, and for values $\varepsilon_0 = 1 > \varepsilon_1 > \ldots > \varepsilon_m = 0$. (Recall that $t_0 = 0, t_m = N_2$, and note that these initial and final defect bounds hold trivially.)

- *Suppose too that there are finite values $s'_1, \ldots, s'_m > 0$ and some $c \in (0, 1)$ such that, for each $r \in [1, m]$, the algorithm A_1 is a c-steepener for $N = N_1$ with respect to the 4-tuple*

  $$(\alpha, \beta, s, s') \ \ := \ \ (\varepsilon_{r-1} \ , \ \varepsilon_r \ , \ \tilde{s}_r \ , \ s'_r) \ .$$

Then, A' (with respect to $\mathsf{MA}_{k+\ell}$) obeys the defect bounds

$$\mathcal{P}' \ = \ \{ \ (t'_0, \varepsilon'_0, s'_0), \ (t'_1, \varepsilon'_1, s'_1), \ \ldots, \ (t'_{m+1}, \varepsilon'_{m+1}, s'_{m+1}) \ \} \ ,$$

where:

- *$s'_1, \ldots, s'_m$ are as given, and we introduce $s'_0 = s'_{m+1} := +\infty$;*
- *$\mathcal{T}' = (t'_0, \ldots, t'_{m+1})$ are as in Definition 10 for A';*
- *$\varepsilon'_0 := 1$, $\varepsilon'_{m+1} := 0$, and for $r \in [1, m]$, we take*

  $$\varepsilon'_r \ := \ \left(\frac{2^{30}\sqrt{N_1} + 1 - c}{N_1} \right) \cdot \varepsilon_{r-1} \ + \ \left(\frac{N_1 - 2^{30}\sqrt{N_1} - 1 + c}{N_1} \right) \cdot \varepsilon_r \ . \tag{9}$$

Proof. The first and last claimed defect bounds in $\mathcal{P}'$ are immediate, concerning the certified lower bounds after no queries and all queries, respectively. Now fix $r \in [1, m]$, and let $z = (z^1, \ldots, z^{N_1}) \in [0, 1]^{N_1 N_2}$ be any fixed input to A'. Let $U^1, \ldots, U^{N_1}$ be the random variables, each supported on $\{[0, 1] \cup \{*\}\}^{N_2}$, describing the partial view for A' of $z^1, \ldots, z^{N_1}$ respectively after Phase $r - 1$. Similarly, let $V^1, \ldots, V^{N_1}$ be random variables describing the views of these strings after Phase r.

Let $X = (X_1, \ldots, X_{N_1})$ be the random variable, supported on $[0, 1]^{N_1}$, where $X_i := LB_{\mathsf{MA}_k}(U^i)$. Similarly let $Y = (Y_1, \ldots, Y_{N_1})$ be given by $Y_i := LB_{\mathsf{MA}_k}(V^i)$. Then:

- Each pair (X_i, Y_i) is independent of $\{(X_j, Y_j)\}_{j \neq i}$. This is because each simulated execution of $A_2(z^j)$ (by A', ranging over $j \in [N_1]$) uses a separate source of randomness, and the number of queries to z^j after Phases $r - 1$ and r are each predetermined;

- For each i, we have $X_i \le Y_i \le \zeta_i$, where $\zeta_i := \mathsf{MA}_k(z^i)$;
- The random variable $(\zeta_i - X_i)$ is $(s_{r-1}, \varepsilon_{r-1})$-small. To see this, note that the simulation of $A_2(z^i)$ has made t_{r-1} queries to z^i after the first $r-1$ Phases, and recall that A_2, by assumption, obeys the defect bounds in $\mathcal{P}$—in particular, that given by $(t_{r-1}, \varepsilon_{r-1}, s_{r-1})$. Then, by our assumption in Eq. (8) and using Prop. 6, $(\zeta_i - X_i)$ is also $(\tilde{s}_r, \varepsilon_{r-1})$-small.

  Similarly, $(\zeta_i - Y_i)$ is (s_r, ε_r)-small, and also $(\tilde{s}_r, \varepsilon_r)$-small.

From this, we conclude that (X, Y, ζ) is an (α, β, s)-ensemble for $N := N_1$, where $\zeta = (\zeta_1, \ldots, \zeta_{N_1})$ and

$$\alpha \ := \ \varepsilon_{r-1} \ , \qquad \beta \ := \ \varepsilon_r \ , \quad s \ := \ \tilde{s}_r \ .$$

Furthermore, during Phase r, A' precisely simulates the execution of $A_1^X(Y)$ with input Y and baseline advice X. After $t'_r = (N_1 - 2^{30}\sqrt{N_1} - 1)t_r + (2^{30}\sqrt{N_1} + 1) \cdot t_{r-1}$ total queries have been made by A', this simulation has advanced by $(N_1 - 2^{30}\sqrt{N_1} - 1)$ simulated queries. Letting $M \subset [N_1]$ be the $(2^{30}\sqrt{N_1} + 1)$ missing/unmade queries in this simulation, A' has the current view U^i of z^i for $i \in M$ and V^j of z^j for $j \notin M$. The accrued value $\mathrm{Val}_{t'_r}$ for A', that is, the lower bound on $\mathsf{MA}_{k+\ell}(z)$ certified by the current partial view $\overline{U}^{t'_r}$ of the full input z, is therefore

$$\mathrm{Val}_{t'_r} \ = \ \mathsf{MA}_{k+\ell}\left(\overline{U}^{t'_r}\right) \ = \ \mathsf{MA}_\ell(\hat{W}) \ ,$$

where $\hat{W} \in [0,1]^{N_1}$ has $\hat{W}_i := X_i$ for $i \in M$ and $\hat{W}_j := Y_j$ for $j \notin M$. The defect $\mathrm{Def}_{t'_r}$ for A' is

$$\mathrm{Def}_{t'_r} \ = \ \mathsf{MA}_{k+\ell}(z) - \mathrm{Val}_{t'_r} \ = \ \mathsf{MA}_\ell(\zeta) - \mathsf{MA}_\ell(\hat{W}) \ . \tag{10}$$

By our assumptions, $A_1^X(Y)$ is a c-steepener for (α, β, s, s'), with α, β, s as above and $s' := s'_r$. Also, $\mathsf{MA}_k(\hat{W})$ represents the certified lower bound of $A_1^X(Y)$ on $\mathsf{MA}_k(Y)$ after $(N_1 - 2^{30}\sqrt{N_1} - 1)$ simulated queries during Phase r. We thus have

$$\mathbb{E}[\exp(s' \cdot \mathrm{Def}_{t'_r})] \ \le \ \exp\left(s'\left(\kappa \cdot \alpha + (1-\kappa) \cdot \beta\right)\right) \ ,$$

using Eq. (10) and the definition of steepeners. This shows that A' obeys the defect bound (with respect to $\mathsf{MA}_{k+\ell}$) given by $(t'_r, \varepsilon'_r, s'_r)$. As $r \in [1, m]$ was arbitrary and we checked the cases $r \in \{0, m+1\}$, we see that A' obeys all of $\mathcal{P}'$, proving the Lemma. □

3.2 Recursive self-composition of a steepener

In this section we show how a steepener working for a sufficiently broad ensemble class allows us to obtain a query-efficient approximation algorithm for MA_k.

Definition 11. *We say that all-queries algorithm $A = A^x(y)$ with baseline advice, taking fixed input size $N = 4^\ell \ge 10^{21}$, is an* ***admissible steepener*** *if, for some fixed $c \in (0,1)$, it is a c-steepener for all 4-tuples in the family*

$$\{ \ (\alpha, \beta, s, s') \ : \ s\Delta \ = \ 1000 \ , \quad s' \ = \ 10^{-4}\sqrt{N} \cdot s \ \}$$

where for each such tuple we use $\Delta := \alpha - \beta$.

Definition 12. *Say that an all-queries algorithm $A = A^x(y)$ with baseline advice, taking* fixed *input size $N = 4^\ell \geq 10^{21}$, is* ***nicely constructive*** *if it can be simulated by a RAM algorithm with black-box access to x, y, advancing one query to y at a time, in which*

1. *The algorithm makes $O(1)$ additions, subtractions, scalar multiplications, and real-number comparisons, plus $O(1)$ logical-control operations per query;*

2. *If the entries of $x, y \in [0,1]^N$ have decimal expansions of length w, then all numerical values produced in the simulation have decimal expansions of length $w + O(1)$;*

3. *After every t^{th} simulated query yielding partial view $y^t \in ([0,1] \cup \{*\})^N$ of y, the RAM algorithm produces the current certified lower bound $LB_{\mathsf{MA}_\ell}(y^t; x)$ relative to baseline x.*

Theorem 2. *Suppose that an admissible steepener A exists, and is nicely constructive. Then,*

1. *There exists an absolute constant $\delta \in (0, .5)$ and, for every $N = 4^k$, a randomized query algorithm $A^*_N(z)$ which, on input $z \in [0,1]^N$, makes $t = t(N) \leq O(N^{1-\delta})$ queries to z, yielding a partial view $u^t \in ([0,1] \cup \{*\})^N$, and satisfies the expected-approximation guarantee*
$$\mathbb{E}[\ LB_{\mathsf{MA}_k}(u^t)\ ]\ \geq\ \mathsf{MA}_k(z) - O(N^{-\delta})\ .$$
Moreover, these algorithms can be simulated by a fixed *randomized RAM algorithm $M^z(k, w)$ with oracle access to $z \in [0,1]^N$, whose entries are of decimal length $\leq w$. The algorithm $M^z(k, w)$ makes $t(N)$ black-box queries to z and uses $\mathrm{poly}(k, w) \cdot N^{1-\delta}$ computational steps (and a storage-space bound of the same form).*

2. *From u^t as output above, we can also directly compute in time $\mathrm{poly}(k, w) \cdot t(N)$ a description of a partial selection function sel^*, defined on all Max gates $g = \mathrm{Max}(h, h')$ lying on some path P_i with $u^t \neq *$. This function selects an Avg input gate from $\{h, h'\}$ whose value on $[u^t \searrow 0^N]$ is largest among the two.*

   Any selection function sel which is consistent with sel^*, defines an M-tree T which satisfies $\mathrm{Avg}_{i \in T}(z_i) \geq LB_{\mathsf{MA}_k}(u^t)$.

In the remainder of this section we prove Theorem 2. Sections 4-8 will be devoted to building and analyzing the steepener needed to apply this result. Its existence is asserted in Theorem 3, whose combination with Theorem 2 yields Theorem 2's conclusion unconditionally. This is our main result, also described in Theorem 1 from the Introduction; the connection here obtaining near-optimal strategies, uses the natural correspondence between M-trees in the MA_k formula on input z, and strategic-player strategies in the associated game F.

Definition 13 (Compositional families of algorithms)**.** *Suppose $A_1 = A_1^x(y)$ is an all-queries algorithm taking baseline advice for the fixed input size $N = N_1 = 4^\ell \geq 10^{21}$. We use A_1 to define a sequence of (advice-free) all-queries algorithms $\hat{A}_q$ for $q \geq 0$, where $\hat{A}_q(y)$ takes inputs $y \in [0,1]^{N_q}$ with*
$$N_q\ :=\ N_1^q\ =\ 4^{q\ell}\ .$$

These algorithms are defined in conjunction with a family $\{\mathcal{T}_q\}_{q\geq 0}$ *of sets of epoch-markers, where* $\mathcal{T}_q$ *is a set of epoch-markers for* $\hat{A}_q$ *in* $[0, N_q]$.

We let $\hat{A}_0$ *be the algorithm on* $4^0 = 1$ *input coordinate* y_1 *that simply makes its query. We let* $\mathcal{T}_0 := (0, 1)$*, the two epoch-markers designating a single, 1-query epoch for an execution of* $\hat{A}_0$.

We then extend our definition inductively for $q \geq 1$ *by letting*

$$\hat{A}_q \;=\; \mathrm{comp}(A_1, \hat{A}_{q-1}; \mathcal{T}_{q-1}) \ , \tag{11}$$

and letting $\mathcal{T}_q$ *be defined as the derived set* $\mathcal{T}'$ *produced by Def. 10 for the composed algorithm* $A' = \hat{A}_q$ *from its defining expression in Eq. (11).*

Recall that in Def. 10, the derived set of epoch-markers contains one more value than the starting set. Thus each $\mathcal{T}_q$ *has* $q + 2$ *values, which we denote by*

$$\mathcal{T}_q \;=\; (t_{q,0},\ t_{q,1},\ \ldots,\ t_{q,q+1})$$

where $t_{q,0} = 0$ *and* $t_{q,q+1} = N_q$*. With these values* $\mathcal{T}_q$ *delimits* $q + 1$ *epochs for* $\hat{A}_q$*. Also, using Eq. (7), the remaining terms obey the recurrence*

$$t_{q,r} \;=\; (N_1 - 2^{30}\sqrt{N_1} - 1)\cdot t_{q-1,r} + (2^{30}\sqrt{N_1} + 1)\cdot t_{q-1,r-1} \qquad \text{for } q \geq 1\ ,\ r \in [1, q]\ . \tag{12}$$

The next lemma points out the efficient-computability properties of these composed families.

Lemma 8. *Let* $A_1 = A_1^x(y)$ *be as in Def. 13 for fixed input size* $N_1 = 4^\ell$*. Assume that* A_1 *is nicely constructive (Def. 12). Then, there exists an algorithm* $B^z(k, w, t)$ *expecting inputs* $k = q\ell$ *for some* $q \geq 1$*, integers* $w > 0, t \in [1, 4^k]$*, and oracle access to* $z \in [0, 1]^N$ *for* $N = 4^{q\ell}$*, whose entries are of decimal length at most* w*. On these inputs,* $B^z(k, w, t)$ *runs for*

$$T \;\leq\; \mathrm{poly}(q, w)\cdot t$$

computational steps, and simulates the first t *queries of* $\hat{A}_q(z)$*, where* $\hat{A}_q$ *is as induced by* A_1 *in Def. 13.*

Proof. The algorithm B works directly based on the inductive definition $\hat{A}_q = \mathrm{comp}(A_1; \hat{A}_{q-1}; \mathcal{T}_{q-1})$, with reference to Def. 9. First, it explicitly computes all relevant values $\{t_{q',r}\}_{q'\leq q, r'\in[0,q'+1]}$ and associated sets of epoch-markers $\mathcal{T}_0, \ldots, \mathcal{T}_q$, in time $\mathrm{poly}(q)$, using Eq. (12).

A rooted, directed tree $\mathcal{H}$ of nodes is then produced, beginning with a node for $(\hat{A}_q, z)$. We maintain the invariant that, after $t' \in [1, t]$ simulated queries of $\hat{A}_q(z)$, a node is created for every sub-input z' of z of any length $4^{q'\ell}$ (corresponding to the input to a sub-formula of MA_k) for which the corresponding simulated instance of $\hat{A}_{q'}(z')$ has been allocated some number $t'' \geq 1$ of queries so far in the operation of $\hat{A}_q(z)$. We then say that z' is "active". Moreover, each such node $(\hat{A}_{q'}, z')$ is equipped with an explicit simulation of the top level of $\hat{A}_{q'} = \mathrm{comp}(A_1; \hat{A}_{q'-1}; \mathcal{T}_{q'-1})$ on input z', in which t'' queries have been made by A_1 and the corresponding real-valued inputs to A_1 produced.

The node created for such $(\hat{A}_{q'}, z')$ is assigned as children in $\mathcal{H}$ the nodes $(\hat{A}_{q'-1}, z'')$ for sub-inputs z'' of z' of size $4^{(q'-1)\ell}$, as these nodes are created. Nodes are not created for sub-inputs until they become active.

Each top-level query allocated in $\hat{A}_q$ induces $q-1$ more simulated queries along a path in $\mathcal{H}$, leading to the creation of at most $q-1$ additional nodes in $\mathcal{H}$. Each corresponding simulation of some $\hat{A}_{q'}$ can be advanced in $\mathrm{poly}(q, w)$ computational steps, as A_1 is nicely constructive (and using the efficiently-computable action of the composition scheme of Def. 9, and the fact that all real numbers involved in this simulation have decimal expansion length $O(q + w)$). The Lemma follows. □

Next, to help analyze families arising from Def. (13), we will arithmetically define families of parameters $\varepsilon_{q,r}$, along with associated values $\Delta_{q,r}$, $s^0_{q,r}$.

Definition 14. *For the values $N \geq 10^{21}$ and $c \in (0, 1)$ given for the admissible steepener assumed in Theorem 2, define*

$$\kappa \;:=\; \frac{2^{30}\sqrt{N} + 1 - c}{N} \;.$$

Define

$$\{\ \varepsilon_{q,r}\ \}_{q\geq 0, r\in\mathbb{Z}}\ ,$$

as follows. Let $\varepsilon_{0,r} := 1$ for $r \leq 0$ and $\varepsilon_{0,r} := 0$ for $r \geq 1$. Then, for $q \geq 1$, inductively let

$$\varepsilon_{q,r} \;:=\; \kappa \cdot \varepsilon_{q-1,r-1} + (1-\kappa)\varepsilon_{q-1,r}\ . \tag{13}$$

We also define the associated family

$$\Delta_{q,r} \;:=\; \varepsilon_{q,r-1} \;-\; \varepsilon_{q,r}\ , \qquad (q \geq 0, r \in \mathbb{Z})\ .$$

While we define $\varepsilon_{q,r}$ for all $q \geq 0, r \in \mathbb{Z}$, only the range $r \in [0, q+1]$ will have meaning for us; the remaining values are just introduced to handle boundary cases more smoothly. The following facts are easily verified using Eq. (13) and the base case for $q = 0$:

Proposition 7. *1. For each $q \geq 0$, the sequence $\{\varepsilon_{q,r}\}$ is nonincreasing in r.*

2. The family $\{\ \Delta_{q,r}\ \}_{q\geq 0, r\in\mathbb{Z}}$ obeys

$$\Delta_{q,r} \;=\; \kappa\Delta_{q-1,r-1} + (1-\kappa)\Delta_{q-1,r} \qquad \textit{for all}\ \ q \geq 1\ ,\ r \in \mathbb{Z}\ .$$

From this Proposition, one sees that each term $\Delta_{q,r}$ with $r \in [1, q+1]$ gains a nonzero contribution stemming from the base-case term $\Delta_{0,1} = 1 - 0 = 1$, while others receive only 0-contributions from base-case terms $\Delta_{0,r'} = 0$ for $r' \neq 1$. Thus,

Proposition 8. *For each $q \geq 0$, we have $\Delta_{q,r} > 0$ if $r \in [1, q+1]$, and $\Delta_{q,r} = 0$ otherwise.*

Our *goal* in defining the quantities above is to show:

Lemma 9. *If $A_1 = A_{st}$ is chosen as an admissible steepener for value $N = N_1 \geq 10^{21}$, then for each $q \geq 0$, the algorithm $\hat{A}_q$ (as in Def. 13) obeys defect bounds of form*

$$\mathcal{P}_q \;=\; \{\; (t_{q,r}, s_{q,r}, \varepsilon_{q,r}) \;\}_{r\in[0,q+1]} \tag{14}$$

for MA_k (on all inputs), where $k = k_q$ satisfies $N_q = 4^k$, and the $\{t_{q,r}\}$ and $\{\varepsilon_{q,r}\}$ are as in Defs. 13 and 14, respectively; and for some sequence $s_{q,0}, \ldots, s_{q,q+1}$, in which

1. $s_{q,0} = s_{q,q+1} = \;+\infty$,
2. *For $r \in [1, q]$ we have $0 < s_{q,1} < +\infty$,*
3. *For each $r \in [1, q+1]$, we have* $\quad \Delta_{q,r} \cdot \min(s_{q,r-1} \;,\; s_{q,r}) \;\geq\; 1000.$

We will then use this result to prove Theorem 2.

Proof of Lemma 9. We prove the statement using induction on $q \geq 0$. For $q = 0$, the assertion is simply that $\hat{A}_0$ obeys defect bounds $\mathcal{P}_0 = \{(0, 1, +\infty), (1, 0, +\infty)\}$, items 1-3 then being easily checked. This is immediate from the definitions: in any execution of $A_1(y)$ for $y = y_1 \in [0, 1]$, we always have $\mathrm{Def}_0 \leq 1$ and $\mathrm{Def}_1 = 0$.

Now let $q \geq 1$ and assume the statement proved for values $q' < q$. We will apply Lemma 7 to $\hat{A}_q = \mathrm{comp}(A_1, \hat{A}_{q-1}; \mathcal{T}_{q-1})$, and to the epoch bounds $\mathcal{P}_{q-1}$ inductively assumed to hold for $\hat{A}_{q-1}$. For each $r \in [1, q]$ and successive pair $(t_{q-1,r-1}, s_{q-1,r-1}, \varepsilon_{q-1,r-1})$, $(t_{q-1,r}, s_{q-1,r}, \varepsilon_{q-1,r})$ in $\mathcal{P}_{q-1}$, we let

$$\tilde{s}_r \;:=\; 1000/\Delta_{q-1,r} \;,$$

which by item 3 of our inductive assumption satisfies $\tilde{s}_r \;\leq\; \min(s_{q-1,r-1} \;,\; s_{q-1,r})$. Also, define

$$s'_r \;:=\; 10^{-4}\sqrt{N_1} \cdot \tilde{s}_r \qquad \text{for} \;\; r \in [1, q] \;.$$

Note that, as A_1 is an admissible steepener (and using $\Delta_{q-1,r} = \varepsilon_{q-1,r-1} - \varepsilon_{q-1,r}$), it is a c-steepener for each 4-tuple

$$(\varepsilon_{q-1,r-1} \;,\; \varepsilon_{q-1,r} \;,\; \tilde{s}_r \;, s'_r) \;.$$

Thus, the assumptions of Lemma 7 (with $m := q$) are met by

$$A_1 \;, \quad \hat{A}_{q-1} \;, \quad \mathcal{P} \;:=\; \mathcal{P}_{q-1} \;, \quad \{\; \tilde{s}_r, s'_r \;\}_{r\in[1,q]} \;.$$

From that Lemma, we infer that $\hat{A}_q$ obeys defect bounds

$$\mathcal{P}'_q \;=\; \{\; (t_{q,r} \;,\; s'_r \;,\; \varepsilon_{q,r}) \;\}_{r\in[0,q+1]}$$

where $t_{q,r}, \varepsilon_{q,r}$ are as in the desired bounds $\mathcal{P}_q$ (by Eqs. (13) and (9)), and where we let $s'_0 = s'_{q+1} := +\infty$.

We will show that for $r \in [1, q+1]$,

$$\Delta_{q,r} \cdot \min(s'_{r-1} \;,\; s'_r) \;\geq\; 1000 \;. \tag{15}$$

We will then be justified in taking $s_{q,r} := s'_r$ and $\mathcal{P}_q := \mathcal{P}'_q$ as the needed defect bounds for $\hat{A}_q$, meeting the needed item 3 in the Lemma and extending the inductive hypothesis.

Recall that, from Prop. 7, we have

$$\Delta_{q,r} \;=\; \kappa\Delta_{q-1,r-1} + (1-\kappa)\Delta_{q-1,r} \;. \tag{16}$$

We have $2^{30}/\sqrt{N_1} < \kappa < .1$ from the largeness of N_1, so it follows from Eq. (16) that

$$\Delta_{q,r} \;\geq\; \max\left(\, 2^{30}\Delta_{q-1,r-1}/\sqrt{N_1} \;,\; .9\Delta_{q-1,r}\right) \;.$$

Next, if $r \in [2,q]$, then $\Delta_{q-1,r-1} > 0$ and we have

$$1/\Delta_{q,r} \;\leq\; \min\left(\frac{\sqrt{N_1}}{2^{30}} \cdot \frac{1}{\Delta_{q-1,r-1}} \;,\; \frac{1}{.9}\frac{1}{\Delta_{q-1,r}}\right) \;. \tag{17}$$

Using the first bound in Eq. (17), we have

$$\begin{aligned} s'_{r-1} \;&=\; 10^{-4}\sqrt{N_1}\cdot \tilde{s}_{r-1} \\ &=\; .1\sqrt{N_1}/\Delta_{q-1,r-1} \\ &\geq\; (.1\sqrt{N_1})\cdot(2^{30}/\sqrt{N_1})/\Delta_{q,r} \\ &\geq\; 1000/\Delta_{q,r} \;, \end{aligned}$$

as needed. In the case $r=1$, we have $s'_{r-1} = +\infty$ and the above inequality holds as well.

Similarly, from $\kappa < .1$ and Eq. (17) we have $\Delta_{q,r} \geq .9\Delta_{q-1,r}$, so that, if $r \leq q$, we find that

$$\begin{aligned} s'_{r} \;&=\; .1\sqrt{N_1}/\Delta_{q-1,r} \\ &\geq\; (.1\sqrt{N_1})\cdot .9/\Delta_{q,r} \\ &\geq\; 1000/\Delta_{q,r} \;. \end{aligned}$$

If $r = q+1$, then $s'_r = +\infty$ and the above inequality again holds. This establishes Eq. (15) for $r \in [1, q+1]$ and extends the inductive hypothesis to the value q, completing the proof. □

Proof of Theorem 2. **(1.)** First, consider input size N_q, for an arbitrary value $q \geq 1$. The algorithm $\hat{A}_q$ given by q-fold composition of an admissible steepener, as studied in Lemma 9, is an all-queries algorithm, but we will show that the defect bounds $\mathcal{P}_q$ it obeys (as given in that Lemma) are powerful enough that it may simply be halted early to yield the desired fast approximation algorithm for MA_k, where $N_q = 4^k$.

The two recurrences for the parameters $t_{q,r}$ and $\varepsilon_{q,r}$ appearing in $\mathcal{P}_q$ are similar, and for comparison's sake we increase their similarity by "scaling down" the epoch-markers. Using

$$f_{q,r} \;:=\; t_{q,r}/N_q \;=\; t_{q,r}/N_1^q \;,$$

and transforming from Eq. (12), we obtain

$$f_{q,r} \;=\; \frac{2^{30}\sqrt{N}+1}{N_1}\cdot f_{q-1,r-1} \;+\; \frac{N_1 - 2^{30}\sqrt{N}-1}{N_1}\cdot f_{q-1,r} \qquad \text{for } q \geq 1 \;,\; r \in [1,q] \;. \tag{18}$$

Next, let

$$g_{q,r} \ := \ 1 - f_{q,r} \ ,$$

and note that $\{g_{q,r}\}$ obeys the <u>same</u> recurrence as $\{f_{q,r}\}$, but unlike these values, $\{g_{q,r}\}$ shares the same base cases and boundary conditions as $\{\varepsilon_{q,r}\}$ (here restricted to the range $r \in [0, q+1]$): both are instances of the recurrence

$$P_{q,r} \ = \ a \cdot P_{q-1,r-1} \ + \ (1-a) \cdot P_{q-1,r} \qquad \text{for } q \geq 1 \ , \ r \in [1,q] \ . \tag{19}$$

with base cases $P_{0,0} = 1, P_{0,1} = 0$ and obeying boundary conditions $P_{q,0} = 1, P_{q,q+1} = 0$ for $q \geq 1$. They differ only in the value of $a \in (0,1)$—crucially, this value is smaller for $\{\varepsilon_{q,r}\}$.

We now review the classical recurrence of Eq. (19); we use the natural probabilistic interpretation. For $q \geq 1$, consider a random sum

$$S_q \ = \ \sum_{i \in [1,q]} Z_i$$

of q independent 0/1 Bernoulli trials, where each Z_i is a-biased. (We also let $S_0 \equiv 0$.) Define

$$\hat{P}_{q,r} \ := \ \Pr[\ S_q \geq r\ ] \ ,$$

and note that $\{\hat{P}_{q,r}\}_{q \geq 0, r \in [0,q+1]}$ obeys Eq. (19) and the base cases and boundary conditions given for $\{P_{q,r}\}$. Thus, $P_{q,r} \equiv \hat{P}_{q,r}$.

No sum-free, "closed-form" expression seems to be known for these tail probabilities. However, it follows from Chernoff bounds applied to each of the S_q that, for any fixed value of $a \in (0,1)$ and any fixed $\delta > 0$, we have

$$\hat{P}_{q,r} \ \geq \ 1 - \exp(-\Omega_{a,\delta}(q)) \quad \text{if } r \leq (a-\delta)q \ ,$$

$$\hat{P}_{q,r} \quad \leq \quad \exp(-\Omega_{a,\delta}(q)) \qquad \text{if } r \geq (a+\delta)q \ .$$

In the recurrence for $g_{q,r}$, identical to that of $f_{q,r}$ in Eq. (18), we have a constant $a = a_1 = \frac{2^{30}\sqrt{N_1}+1}{N_1}$. In the recurrence of Eq. (13) for $\varepsilon_{q,r}$, we have the value $a = a_2 = a_1 - c/N_1$. Now let

$$r \ = \ R(q) \ := \ \left\lceil \frac{a_1 + a_2}{2} \cdot q \right\rceil \ ,$$

we have

$$\varepsilon_{q,R(q)} \ \leq \ \exp(-\Omega(q)) \ , \qquad \text{while} \qquad g_{q,R(q)} \ \geq \ 1 - \exp(-\Omega(q)) \ .$$

Thus

$$f_{q,R(q)} \ \leq \ \exp(-\Omega(q)) \ , \quad \text{i.e.,}$$

$$t_{q,R(q)} \ \leq \ N_1^q \cdot \exp(-\Omega(q)) \ = \ (N_1^q)^{1-\Omega(1)} \ ,$$

using $N_1 = O(1)$.

The upshot is as follows: on input $z \in [0,1]^{N_q}$ where $N_q = N_1^q = 4^k$, if we halt our algorithm $A_q(z)$ after $t := t_{q,R(q)}$ steps and use the certified lower bound

$$LB_{\mathsf{MA}_k}(u^t)$$

where u^t is the partial view of z after t queries, our query complexity is $N_q^{1-\Omega(1)}$ and our additive error $\mathsf{MA}_k(z) - LB_{\mathsf{MA}_k}(u^t)$ is $(s, \varepsilon_{q,R(q)})$-small for some $s > 0$; in particular, by Jensen's inequality, this implies

$$\mathbb{E}[\mathsf{MA}_k(z) - LB_{\mathsf{MA}_k}(u^t)] \ \leq \ \varepsilon_{q,R(q)} \ \leq \ N_q^{-\Omega(1)} \ .$$

This yields our main goal for input lengths of form $N = N_q$ for $q \geq 1$. This covers all values of form $N = 4^{h\ell}$, where $N_1 = 4^\ell$.

For general input lengths $N = 4^k$, we can write $4^k = 4^{h\ell+h'} = N_h \cdot 4^{h'}$ where $N_1 = 4^\ell$ and $0 \leq h' < \ell = O(1)$. Breaking an input vector z up and writing

$$\mathsf{MA}_k(z) \ = \ \mathsf{MA}_{h\ell+h'}\left(z^1, \ldots, z^{4^{h'}}\right) \ = \ \mathsf{MA}_{h'}\left(\mathsf{MA}_{h\ell}\left(z^1\right), \ldots, \mathsf{MA}_{h\ell}\left(z^{4^{h'}}\right)\right)$$

we use our algorithm A_h on each of the $4^{h'}$ input blocks, yielding certified lower bounds LB_i on each $\mathsf{MA}_{h\ell}(z^i)$, with $\mathbb{E}[LB_i] \leq \exp(-\Omega(h))$; the value

$$LB' \ := \ \mathsf{MA}_{h'}(LB_1, \ldots, LB_{4^{h'}})$$

is then a certified lower bound for $\mathsf{MA}_k(z)$. Also, we have

$$\begin{aligned}
\mathbb{E}[\mathsf{MA}_{h'}(LB_1, \ldots, LB_{4^{h'}})] \ &\geq \ \mathsf{MA}_{h'}(\mathbb{E}[LB_1], \ldots, \mathbb{E}[LB_{4^{h'}}]) \\
&\geq \ \mathsf{MA}_{h'}\left(\mathsf{MA}_{h\ell}(z^1) - \tau, \ldots, \mathsf{MA}_{h\ell}\left(z^{4^{h'}}\right) - \tau\right) \\
&= \ \mathsf{MA}_k(z) - \tau \ ,
\end{aligned}$$

for a value $\tau \leq \exp(-\Omega(h))$. (Above, we used Propositions 1 and 2.) Then, using the fact that $\ell = O(1)$, we also have $\mathbb{E}[LB'] \geq \mathsf{MA}_k(z) - N^{-\Omega(1)}$. Thus we achieve the desired algorithm for general $N = 4^k$.

(2.) This item can be obtained easily by a subsequent computation on u^t which, for every i for which $u_i^t \neq *$, computes every gate value along P_i on input $[u^t \searrow 0^N]$. This computation proceeds directly from the aforementioned input gates i, working toward the root/output values according to the gate definitions; we substitute 0 for the value of every wire not on some such path P_i.

The partial selection function sel* is as described in item 2. Its guarantee in the item follows directly from the definitions. □

4 On the structure of MA_k inputs

Throughout Section 4, let $N = 4^k$, with $k \geq 1$. Here we study the behavior of the MA_k function under changes to its inputs, and give definitions that will help us state and analyze our steepener construction in Section 5.

4.1 Preliminary lemmas on random walks

Our first lemma of this section does not directly treat the MA_k formula, but will be used later to analyze that formula's behavior under random experiments. Our lemma discusses a scenario in which we take a random path down a binary tree having certain vertices "marked". It bounds the probability that our path hits many marked vertices, while simultaneously having only a few "near-misses" in which the path approaches, but does not contain, a marked vertex. The lemma's bound is chosen for simplicity, and doesn't aim to be comprehensive or best-possible.

Throughout this subsection, let $\mathcal{H}$ be a full, balanced binary tree of depth $d \geq 1$ with designated root vertex r. (Thus, $\mathcal{H}$ has 2^d leaves.) For a leaf node v, let $\mathbf{P}_v$ be the unique path in $\mathcal{H}$ from r to v. Say that a vertex v' "hangs from" $\mathbf{P}_v$ if v' is not in $\mathbf{P}_v$, but has a neighbor (parent) on $\mathbf{P}_v$.[8]

Lemma 10. *Let M be a subset of the non-leaf, non-root vertices of $\mathcal{H}$ (M is the "marked" set). For $a, b \geq 0$, let $Q(a, b)$ denote the probability, over a uniformly chosen leaf v, that $\mathbf{P}_v$ contains <u>at least</u> a marked vertices, while <u>at most</u> b marked vertices hang from $\mathbf{P}_v$. Then, $Q(.45d, .1d) \leq (.998)^d$.*

Proof of Lemma 10. Regard v as being selected by sequentially generating $\mathbf{P}_v$ as a random walk down $\mathcal{H}$, starting at the root r. To clarify our reasoning we describe this process being driven by two sequences of i.i.d. Bernoulli random variables, call them $U_1, U_2, \ldots, U_d$ and $V_1, \ldots, V_d$ (each equal to 0 or 1 with equal probability). The walk is generated as follows:

Random-walk experiment on $\mathcal{H}$:

1. Initialize the "current node" $v_c := r$;
2. **While** v_c is non-leaf:
   - **If** v_c has either zero or two marked children, and is the i^{th} such vertex visited, **then** let the next step of $\mathbf{P}_v$ be determined as the left or right child according to U_i (with $0 = $ left, say);
   - **Else** (v_c has exactly one marked child, and is the j^{th} such vertex visited), let the next step of $\mathbf{P}_v$ be determined as the left or right child according to V_j; only <u>this time</u>, with $V_j = 0$ selecting the unmarked child, and 1 the marked child.
3. Let v be the final, leaf-vertex value taken by v_c.

It is clear that the walk is uniform over paths $\mathbf{P}_v$, as desired. Now, if $\mathbf{P}_v$ is to contain at least $.45d$ marked vertices, and have at most $.1d$ hanging marked vertices, then, first of all, the walk must visit at most $.1d$ vertices having two marked children. At least $.35d$ of the marked vertices on $\mathbf{P}_v$ must therefore be reached in a step from a vertex v' having exactly

[8](Later we will discuss "pendant" gates in MA formulas, a closely-related notion held distinct by the different term.)

one marked child. Also, at most $.1d$ steps from a v'' with one marked child can land on the unmarked child. Thus we must have $V_1 + \ldots + V_{\lceil .35d \rceil} \geq .25d$. This occurs with probability at most $(.998)^d$, by a Chernoff bound (item 1 of Lemma 1, with $p = .5,\ \delta > .2$). ☐

The next lemma is a variation, with the same basic reasoning involved. Its statement is again chosen for simplicity.

Lemma 11. *Let M_{good} and M_{bad} be two disjoint subsets of the non-leaf, non-root vertices (the "marked" set). Assume that if $v' \in M_{good}$, and if v'' is the (unique) sibling in $\mathcal{H}$ of v', then at least one child of v'' is in M_{bad}.*

Let $R(a, b)$ denote the probability, over a uniformly chosen leaf v, that $\mathbf{P}_v$ contains at least a vertices from M_{good}, and contains at most b vertices from M_{bad}. Then, $R(a, .1a) \leq (.998)^a$.

Proof. This time we use a different probabilistic experiment to generate $\mathbf{P}_v$ in a sequential fashion, as follows. Namely, each time the current vertex v_c has one or more child in M_{good}, we generate the next *two* successive steps of our path with a random variable $\overline{V}_j$ taking values in $\{0, 1\}^2$; each outcome is equally likely and corresponds to a distinct 2-step continuation, with outcome 00 always made to correspond to a continuation landing on a vertex in M_{bad}.

We use random variable $\overline{V}_j$ for the j^{th} such generative 2-step process. When v_c has no children in M_{good}, we drive the walk for *one* step using U_i-variables, with U_i driving the i^{th} such step, as in the previous Lemma's proof.

One can observe that, for the Lemma's outcome of interest to occur,

1. Each of $\overline{V}_1, \ldots, \overline{V}_{\lceil a/2 \rceil}$ must be used to drive the walk (since each such use acquires at most 2 vertices from M_{good} for $\mathbf{P}_v$, and no step driven by U_i-variables acquires an M_{good} vertex); and,

2. At most $.1a$ of $\overline{V}_1, \ldots, \overline{V}_{\lceil a/2 \rceil}$ may take on the value 00 (since when item 1 holds, each such outcome adds to the number of vertices on $\mathbf{P}_v$ lying in M_{bad}).

Letting $W_i = \mathbf{1}_{[\overline{V}_i = 00]}$, these Bernoulli variables are i.i.d. with expectation $p = .25$. By our observations, $R(a, .1a) \leq \Pr[W_1 + \ldots + W_{\lceil a/2 \rceil} \leq .1a]$, and the Lemma follows from another Chernoff bound (this time item 2 of Lemma 1). ☐

4.2 Slack, dominance, and vanguard coordinates

Recall that $\mathbf{T}$ denotes the formula for MA_k and is regarded as a directed graph whose outgoing edges from a gate g go to each of its input/children gates.

Definition 15. *Fix an input $y \in [0, 1]^N$ to MA_k.*

- *For a gate $g \in \mathbf{T}$, let $g(y) \in [0, 1]$ denote its value in the computation on y.*

- *If g is a* Max *gate with* Avg*-gate children (i.e., inputs) h and h', we define the **slack of g with respect to h (on y)** in a height-dependent way, as*

$$\mathrm{Slack}(g; h, y) \ := \ 2^q \cdot (h(y) - h'(y)) \ ,$$

where $2q$ is the height of g (i.e., g is at distance $2q$ from the input-variable gates in $\mathbf{T}$). Thus $\mathrm{Slack}(g, h'; y) = -\mathrm{Slack}(g; h, y)$.

- *We define the* ***decisiveness*** *of a* Max *gate g as above on y as*

$$\mathrm{Dec}(g;y) \ := \ |\mathrm{Slack}(g;h,y)|$$

(the choice of child h here is irrelevant). We also say gate g is ***δ-decisive on*** *y (for value $\delta > 0$) if $\mathrm{Dec}(g;y) \geq \delta$.*

- *For a path P_j as in Eq. (2), fix attention to a* Max *node $g = p^{j,2s} \in \mathbf{T}$, at height $2(k-s)$ above the variable nodes, and with input* Avg*-gate children h, h'. Here, assume that $h = p^{j,2s+1}$ is the successor of g on the path P_j. For $\delta > 0$, say that P_j is* ***δ-dominant at*** *$p^{j,2s}$ if $\mathrm{Slack}(g;h,y) \geq \delta$, and* ***$\delta$-beaten at*** *$p^{j,2s}$ if $\mathrm{Slack}(g;h,y) \leq -\delta$.*

  We also simply say that P_j is ***dominant at*** *$p^{j,2s}$ if $\mathrm{Slack}(g;h,y) > 0$,* ***beaten at*** *$p^{j,2s}$ if this slack is negative, or* ***tied*** *if the slack is 0.*

- *Define the* ***vanguard*** *of y, denoted*

$$\mathrm{Van}(y) \ \subset \ [N] \ ,$$

as the set of all $j \in [N]$ for which P_j is dominant at each of its k Max *nodes $p^{j,2s}$, for $s \in [0, k-1]$.*

- *For $\delta > 0$, define the* ***δ-decisive vanguard*** *of y, denoted*

$$\mathrm{Van}^{\delta}(y) \ \subseteq \ \mathrm{Van}(y) \ \subset \ [N] \ ,$$

as the set of all $j \in [N]$ for which P_j is δ-dominant at each of its k Max *nodes $p^{j,2s}$, for $s \in [0, k-1]$.*

- *For a path P_j as above and integer $s \in [0, k-2]$, suppose that the* Avg*-node $h = p^{j,2s+1}$ has* Max *nodes $g = p^{j,2(s+1)}$ and some g' (not in P_j) as children in $\mathbf{T}$. We then say that g' is* ***pendant to*** *P_j* ***at*** *h.*

The next propositions are immediate from the definitions.

Proposition 9. $\mathrm{Van}(y)$ *consists of exactly those $i \in [N]$ which are contained in every optimal M-tree for y. In particular, this implies $|\mathrm{Van}(y)| \leq \sqrt{N}$.*

Proposition 10. *Suppose that $i, j \in [N]$ are distinct indices both in the M-tree T, for some $\delta > 0$. Let $m \in [0, k]$ be the largest value for which $p^{i,m} = p^{j,m}$ (the paths P_i, P_j diverge on their $(m+1)^{st}$ steps).*

Then $m = 2t - 1$ is odd ($p^{i,m}$ is an Avg *gate). Moreover, if $m < k - 1$ then the (*Max*) gate $p^{j,m+1}$ is pendant to P_i at $p^{i,m}$.*

4.3 On reducing coordinate values

Definition 16. *For $u \in \{\mathbb{R} \cup \{*\}\}^N$ and $i \in [N], a \in \mathbb{R}$, let $u[i \leftarrow a]$ be u with its i^{th} coordinate set to a.*

We study the effect of decrementing a single coordinate. Our first such lemma concerns the effect of a "sufficiently small" decrement to the i^{th} input coordinate.

Lemma 12. *Let $y \in [0,1]^N$, and let $i \in [N]$ with $y_i > 0$. Let $\delta \in (0, y_i]$.*

1. *If $i \in \mathrm{Van}(y)$ and moreover, all* Max *gates g along P_i have $\mathrm{Dec}(g; y) \geq \delta$ (i.e., if $i \in \mathrm{Van}^{\delta}(y)$), then for each $g = p^{i,2t}$ along P_i ($t \in [0, k-1]$), we have the relation*

$$g(y[i \leftarrow y_i - \delta]) \;=\; g(y) - 2^{-(k-t)} \cdot \delta \ ,$$

Also, P_i is not beaten at g on $y[i \leftarrow y_i - \delta]$, and

$$\mathrm{Dec}(g; y[i \leftarrow y_i - \delta]) \;=\; \mathrm{Dec}(g; y) - \delta \ .$$

2. *Next, suppose $i \notin \mathrm{Van}(y)$, and let $t \in [0, k-1]$ be <u>maximal</u> for which $g = p^{i,2t}$ is not dominant for P_i on y. Suppose that $\delta \leq \mathrm{Dec}(p^{i,2t'}; y)$ for each $t' > t$ (a vacuous requirement if $t = k-1$). Then for all such $t' > t$*

$$p^{i,2t'}(y[i \leftarrow y_i - \delta]) \;=\; p^{i,2t'}(y) - 2^{-(k-t')}\delta$$

and

$$\mathrm{Dec}(p^{i,2t'}; y[i \leftarrow y_i - \delta]) \;=\; \mathrm{Dec}(p^{i,2t'}; y) - \delta \ ,$$

and P_i is not beaten at $p^{i,2t'}$; while for $p^{i,2t}$ we have

$$p^{i,2t}(y[i \leftarrow y_i - \delta]) \;=\; p^{i,2t}(y) \ , \qquad \mathrm{Dec}(p^{i,2t}; y[i \leftarrow y_i - \delta]) \;=\; \mathrm{Dec}(p^{i,2t}; y) + \delta \ ,$$

and P_i is beaten at $p^{i,2t}$.

Moreover, for $s < t$, we have

$$p^{i,2s}(y[i \leftarrow y_i - \delta]) \;=\; p^{i,2s}(y) \ , \qquad \mathrm{Dec}(p^{i,2s}; y[i \leftarrow y_i - \delta]) \;=\; \mathrm{Dec}(p^{i,2s}; y) \ ,$$

and the dominant/tied/beaten status of P_i at $p^{i,s}$ does not change.

Proof. **(1.)** We prove by induction on $d = 0, 1, \ldots, k$ that

$$p^{i,2(k-d)}(y[i \leftarrow y_i - \delta]) \;=\; p^{i,2(k-d)}(y) - 2^{-d}\delta \ .$$

For $d = 0$ this is just the statement that the i^{th} input gate decreases in value by δ, which is immediate. Now let $d > 0$ and assume the statement true for smaller values. Inspecting the formula, we can write the functional relation

$$p^{i,2(k-d)} \;=\; \mathrm{Max}(h_1, h_2) \tag{20}$$

where

$$h_1 \;=\; .5(p^{i,2(k-(d-1))} + g_1)$$

is one of the two Avg gates inputting to the Max gate $p^{i,2(k-d)}$. By assumption in item 2 and our definitions of t' and δ,

$$h_1(y) \;\geq\; h_2(y) + 2^{-d} \cdot \mathrm{Dec}(p^{i,2(k-d)}; y) \;\geq\; h_2(y) + 2^{-d} \cdot \delta \ .$$

Using our inductive assumption (and the fact that g_1 does not depend on input i),

$$\begin{aligned} h_1(y[i \leftarrow (y_i - \delta)]) &= .5\left(p^{i,2(k-(d-1))}(y[i \leftarrow (y_i - \delta)]) + g_1(y[i \leftarrow (y_i - \delta)])\right) \\ &= .5\left(p^{i,2(k-(d-1))}(y) - 2^{-(d-1)}\delta + g_1(y)\right) \\ &= .5\left(p^{i,2(k-(d-1))}(y) + g_1(y)\right) - 2^{-d}\delta \\ &= h_1(y) - 2^{-d}\delta \ , \end{aligned}$$

which is at least as large as $h_2(y) = h_2(y[i \leftarrow y_i - \delta])$. Using Eq. (20), this extends the induction to the value d. At the same time, our analysis showed that P_i is not beaten at $p^{i,2(k-d)}$ on $y[i \leftarrow y_i - \delta]$, and it follows from the calculations that $\mathrm{Dec}(g; y[i \leftarrow y_i - \delta]) = \mathrm{Dec}(g; y) - \delta$. This proves the other parts of item 1 as well.

(2.) The statements for values $t' > t$ are proved using an induction identical to that in item 1, except that we consider only $d = 0, 1, \ldots, k - t - 1$.

For the value t, we write $p^{i,2t} = \mathrm{Max}(h_1, h_2)$, where $h_1 = .5(p^{i,2(t+1))} + g')$ lies on P_i. By our assumption on t in item 2, $h_2(y) \geq h_1(y)$. Decreasing the i^{th} coordinate cannot change this property or the value of h_2, so the value of $p^{i,2t}$ is unchanged, as claimed. Furthermore, our statement for $t' = t+1$ implies that in replacing y_i with $y_i - \delta$, the gate $p^{i,2(t+1))}$ decreases in value by $2^{-(k-(t+1))}\delta$; so that h_1 decreases in value by $2^{-(k-t)}\delta$; this establishes that P_i is beaten at $p^{i,2t}$ on $y[i \leftarrow y_i - \delta]$, and gives the claim about the decisiveness at $p^{i,2t}$.

For $s < t$, our fact that $p^{i,2t}(y[i \leftarrow y_i - \delta]) = p^{i,2t}(y)$ implies that both input values to $p^{i,2s}$ are unchanged, so that both the value and the decisiveness at $p^{i,2s}$ are unaffected, as claimed; as is the dominance status for P_i. This proves item 2. $\square$

Definition 17. *Fix an input $y \in [0,1]^N$ to MA_k. Let $i \in [N]$, and let B_i be the set of* Max *gates $g = p^{i,2s}$ on P_i for which P_i is beaten at g on y. Define*

$$S_i^-(y) \ := \ \sum_{g \in B_i} \mathrm{Dec}(g; y) \ .$$

For occasional use we also define

$$\tilde{S}_i^-(y) \ := \ \sum_{g \in B_i, g \neq p^{i,0}} \mathrm{Dec}(g; y)$$

as the sum omitting the root gate $r = p^{i,0}$ (if P_i is beaten at r).

The next Lemma collects some consequences of Lemma 12 for the effect of a "larger", more general coordinate decrement. It shows that the effect of a decrement to the i^{th} coordinate "splits", in a precise way, between an effect on the MA_k-value, and an effect on the S_i^--value. It also bounds the possible reduction in MA_k-value, and gives sufficient conditions for gates on P_i to remain dominant.

Lemma 13. *Let $y \in [0,1]^N$ and suppose $0 \leq y_i' < y_i$. Then*

1. *We have*

$$y_i - y_i' \ = \ (S_i^-(y[i \leftarrow y_i']) - S_i^-(y)) \ + \ 2^k[\mathsf{MA}_k(y) - \mathsf{MA}_k(y[i \leftarrow y_i'])] \ . \tag{21}$$

2. *If $i \in \mathrm{Van}(y)$, and $\delta_0 > 0$ is the largest value such that $i \in \mathrm{Van}^{\delta_0}(y)$, then $\mathsf{MA}_k(y[i \leftarrow y_i']) \geq \mathsf{MA}_k(y) - \delta_0/2^k$.*

   If $y_i - y_i' \geq \delta_0$, then $\mathsf{MA}_k(y[i \leftarrow y_i']) = \mathsf{MA}_k(y) - \delta_0/2^k$.

   If $i \notin \mathrm{Van}(y)$, then $\mathsf{MA}_k(y[i \leftarrow y_i']) = \mathsf{MA}_k(y)$.

3. *If P_i is dominant at $p^{i,2t}$ for y, with $\mathrm{Dec}(p^{i,2t}; y) > y_i - y_i'$, then P_i is also dominant at $p^{i,2t}$ for $y[i \leftarrow y_i']$.*

4. *If $i \in \mathrm{Van}(y)$ and δ_0 is as in item 1, and if P_i is dominant at $p^{i,2t'}$ for y for $t' = 0, 1, \ldots, t$, and each such t' satisfies $\mathrm{Dec}(p^{i,2t'}; y) > \delta_0$, then P_i is also dominant at $p^{i,2t}$ for $y[i \leftarrow y_i']$, regardless of the magnitude of $(y_i - y_i')$.*

Proof. If $i \in \mathrm{Van}^{\delta}(y)$, where $\delta := y_i - y_i'$, then by applying item 1 of Lemma 12, P_i is beaten at no Max gates on $y[i \leftarrow y_i']$, so that $S_i^-(y[i \leftarrow y_i']) = S_i^-(y) = 0$. Item 1 also tells us that this decrement decreases the value of $p^{i,0}$ by $2^{-k}\delta$, which implies Eq. (21), as well as item 2 above, in this case.

Next, suppose $i \notin \mathrm{Van}^{\delta}(y)$ but that $i \in \mathrm{Van}(y)$ (so that $S_i^-(y) = 0$), and let $p^{i,2t}$ be a Max gate on P_i of minimal decisiveness value $\mathrm{Dec}(p^{i,2t}; y) = \delta_0 > 0$ (this agrees with the definition in item 2). We first consider the "mini-decrement" where we replace y_i with $y_i - \delta_0$. Lemma 12, item 1 tells us that the effect of this mini-decrement is to reduce the value of each gate $p^{i,2t'}$ by $\delta_0 2^{-(k-t')}$; in particular the value of $p^{i,0}$, which computes the MA_k-value of the input, decreases by $\delta_0 2^{-k}$. Also, item 1 tells us that each decisiveness value along P_i reduces by δ_0; in particular, the decisiveness value for $p^{i,2t}$ reduces to 0, so that P_i is not dominant there on $y[i \leftarrow y_i - \delta_0]$, and $i \notin \mathrm{Van}(y[i \leftarrow y_i - \delta_0])$. This vanguard non-membership can only persist after further decrements to the i^{th} coordinate.

We then decompose the remaining decrement amount, namely $(y_i - y_i') - \delta_0 > 0$, into a series of "mini-decrements", each small enough for item 2 of Lemma 12 to apply. Each mini-decrement amount is chosen as the largest possible for which that item applies—a mini-decrement which, by inspection, has the effect of *either* increasing the number of non-dominant Max gates along P_i, *or* completing the overall decrement from y_i to y_i'.

Lemma 12's item 2 informs us that each mini-decrement in this process, by an amount δ', has the effect of increasing the S_i^--value by δ'. It also tells us that the MA_k-value (computed at $p^{i,0}$) does not change. If we let $\delta_1, \ldots, \delta_m$ be the mini-decrement amounts for these steps, it follows that the final S_i^--value is $\delta_1 + \ldots + \delta_m$, so that

$$\begin{aligned}(S_i^-(y[i \leftarrow y_i']) - S_i^-(y)) \; + \; 2^k[\mathsf{MA}_k(y) - \mathsf{MA}_k(y[i \leftarrow y_i'])] \;&=\; \sum_{j \geq 1} \delta_i + 2^k(2^{-k}\delta_0) \\ &=\; \sum_{j \geq 0} \delta_j \\ &=\; y_i - y_i' \;,\end{aligned}$$

giving Eq. (21). Item 2 also holds in this case from our observations.

Finally, the case where $i \notin \mathrm{Van}(y)$ is analyzed in the same way as the previous case, except omitting the initial mini-decrement δ_0. The mini-decrements $\delta_1, \ldots, \delta_m$ now contribute to any preexisting S_i^--value on y, as shown by a fully analogous calculation.

This proves items 1 and 2. Item 3 is easily obtained along the way, by noting that (in view of Lemma 12, item 1) each mini-decrement above by an amount δ' decreases the decisiveness at $p^{i,2t}$ by δ', without changing the dominance of P_i at $p^{i,2t}$. For item 4, we note that in the sequence of mini-decrements $\delta_0, \delta_1, \ldots, \delta_m$ described above, the gates $p^{i,2t'}$ (for $t' \in [0, t]$) remain dominant after the initial δ_0 mini-decrement (by Lemma 12, item 1), and thereafter their dominance status is not changed (by repeated application of the last part of Lemma 12, item 2). $\square$

5 Construction of steepeners

5.1 A-trees and the distribution Γ

Recall that a steepener, as in Def. 6, is an all-queries algorithm $A^x_{st}(y)$ with baseline advice that satisfies a certain performance guarantee.

In brief, our candidate steepener construction $A^x_{st}(y)$ will: choose a set I of $N - 2^{30}\sqrt{N}$ coordinates; query y on all but two of them, call these $\mathbf{i}, \mathbf{j}$; and then, as its key choice, adaptively decide which of these two to query next, before querying all remaining coordinates. Here we develop some terminology and give a suitable distribution Γ over possible outcomes to $(I, \mathbf{i}, \mathbf{j})$ as above.

Definition 18. *An **A-tree** is a subset T of $[N]$, regarded as a subset of variable-gate indices. It is specified by a "selection function" $\mathrm{sel} = \mathrm{sel}_T$ giving, for each Avg gate $h = \mathrm{Avg}(g, g')$ in $\mathbf{T}$, a "selected child" $\mathrm{sel}(h) \in \{g, g'\}$. Then $T = T_{\mathrm{sel}}$ is defined as the set of $i \in [N]$ for which every step in P_i from any Avg gate h goes to $\mathrm{sel}(h)$. (The function sel is not unique for a given A-tree T.)*

Definition 19. *A $2^{30}\sqrt{N}$**-extended A-tree** is a subset T' of $[N]$, regarded as a subset of variable-gate indices. It is specified by a (non-unique) "selection function" $\mathrm{sel} = \mathrm{sel}_{T'}$ giving, for each Avg gate $h = \mathrm{Avg}(g, g')$ at distance at least 61 from the root Max gate in $\mathbf{T}$ (i.e., excluding the topmost 30 layers of Avg nodes), a "selected child" $\mathrm{sel}(h) \in \{g, g'\}$. Then $T' = T'_{\mathrm{sel}}$ is defined as the set of $i \in [N]$ for which every step in P_i from any Avg gate h at distance at least 61 from the root gate, goes to $\mathrm{sel}(h)$.*

Thus a $2^{30}\sqrt{N}$-extended A-tree is of size $2^{30}\sqrt{N}$, and as a subset of $[N]$, can be viewed as a particular type of (disjoint) union of 2^{30} ordinary A-trees. Its main useful property for us is the following:

Proposition 11. *If T is any M-tree and T' a $2^{30}\sqrt{N}$-extended A-tree, we have $|T \cap T'| = 2^{30}$.*

Definition 20. *Let $N = 4^k$ and $i, j \in [N]$. We write*

$$i \perp j$$

(a symmetric relation) to indicate that the paths P_i, P_j diverge at the root Max gate $g = \mathrm{Max}(h, h')$. That is, one of these paths continues through the Avg gate h and the other through h'.

We write $S \perp j$ (or, $j \perp S$) to indicate that $i \perp j$ for all $i \in S$. Note that if $S = T$ is an M-tree and a single $i \in T$ satisfies $i \perp j$, then in fact $T \perp j$.

Definition 21. *Define a distribution* $\Gamma = (\mathcal{I}, \mathbf{i}, \mathbf{j})$ *over sets* $I \subset [N]$ *and pairs* $i, j \in I$ *as follows.*

- *Let* $\hat{T}$ *be a* $2^{30}\sqrt{N}$*-extended A-tree on* $[N]$*, chosen by uniformly generating a selection function* sel *on all* Avg *nodes at distance at least 61 from the root in* $\mathbf{T}$.
- *Let* $\mathcal{I} := [N] - \hat{T}$.
- *Let* $\mathbf{i}, \mathbf{j} \in \mathcal{I}$ *be chosen uniformly from* $\mathcal{I}$ *subject to the condition* $\mathbf{i} \perp \mathbf{j}$.

For brevity, we will say that a tuple (I, j, i) *is* Γ***-supported*** *if it is a possible outcome to* $(\mathcal{I}, \mathbf{j}, \mathbf{i})$*; and similarly we'll say that* (I, j) *is* Γ*-supported if it's a possible partial outcome to* $(\mathcal{I}, \mathbf{j})$.

We collect some easy facts about Γ above:

Proposition 12. *In the distribution* Γ*,*

1. $\mathbf{i}, \mathbf{j}$ *are individually uniform over* $[N]$*;*
2. $\mathbf{i}, \mathbf{j}$ *are individually uniform over* I *conditioned on* $[\mathcal{I} = I]$*;*
3. *Conditioned on* $[(\mathcal{I}, \mathbf{i}) = (I, i)]$*, the value* $\mathbf{j}$ *is uniform over the* $(N - 2^{30}\sqrt{N})/2$ *indices* $i \in I$ *for which* $i \perp j$*. The analogous statement holds with* $\mathbf{i}, \mathbf{j}$ *interchanged;*
4. $|\mathcal{I} \cap T| = \sqrt{N} - 2^{30}$ *for any M-tree* T *(using Prop. 11), and any such* T *contains at most one of* $\mathbf{i}, \mathbf{j}$.

5.2 Description of steepener candidate

Let $C_1 := 1000$. Let $\ell := 10^{10}$ and $N := 4^{\ell}$. Let $\mathfrak{e} := 2^{30}\sqrt{N} + 1$.

Algorithm $A_{st} = A^x_{st}(y)$: on input $y \in [0,1]^N$ and advice $x \in [0,1]^N$ (with $x \le y$),

1. Let $(\mathcal{I}, \mathbf{i}, \mathbf{j}) \sim \Gamma$ be chosen (as in Def. 21), independently of x, y. Let A_{st} make its first $N - \mathfrak{e} - 1$ queries (in ascending-index order) to $(y)_{\mathcal{I}-\{\mathbf{i},\mathbf{j}\}}$.

2. Let u be the resulting "hybrid input", agreeing with y on $\mathcal{I}-\{\mathbf{i},\mathbf{j}\}$ and with x elsewhere.

We choose to query coordinate $\mathbf{i}$ at step $N - \mathfrak{e}$ if at least one of the following **"Selection-Conditions"** holds of u and $\mathbf{i}$:

(a) The path $P_{\mathbf{i}}$ is dominant (with respect to input u) at each of its C_1 Max gates $p^{\mathbf{i},0}, p^{\mathbf{i},2}, \ldots, p^{\mathbf{i},2(C_1-1)}$ closest to the output gate $r = p^{\mathbf{i},0}$; **or,**

(b) The path $P_{\mathbf{i}}$ has more than $.51\ell$ Max gates where it is dominant with respect to u; **or,**

(c) There exists a positive value

$$\gamma \ > \ .001 \cdot S^-_{\mathbf{i}}(u) \ \ge \ 0 \tag{22}$$

and a set G of at least $.01\ell$ Max gates g pendant to $P_{\mathbf{i}}$, such that for each $g \in G$ we have

$$\mathrm{Dec}(g; u) \ \in \ [\gamma, 1000\gamma] \ . \tag{23}$$

(Simplifying observation: if such γ, G exist, then γ can be chosen as the smallest value $\mathrm{Dec}(g; u)$ of any Max gate g in our chosen set G.)

Otherwise (if $\mathbf{i}$ fails Selection-Conditions (a), (b), and (c)), we query coordinate $\mathbf{j}$ at step $N - \mathfrak{e}$. We let $\mathbf{k}_1 \in \{\mathbf{i},\mathbf{j}\}$ be the index chosen for the $(N-\mathfrak{e})^{th}$ query, and let $\mathbf{k}_2 \in \{\mathbf{i},\mathbf{j}\}$ be the sole coordinate in $\mathcal{I}$ not queried after $N - \mathfrak{e}$ steps.

3. The remaining $\mathfrak{e}$ queries to y are made in ascending-index order.

We note that the Selection-Conditions were touched upon in the order (b), (c), (a) in Section 1.5.2 (with (a) described as "less central").

5.3 Efficient computability

Claim 1. *$A^x_{st}(y)$ is a nicely constructive all-queries algorithm, in the sense of Def. 12.*

The Claim follows directly from our description of A_{st} and distribution Γ (which can be sampled from in poly(N) time from its definition); and from the definitions of dominance, $S^-_{\mathbf{i}}(u)$, and $\mathrm{Dec}(g; u)$, along with the simplifying observation that allows us to test Selection-Condition (c) by trying a finite number of candidate values γ.

Claim 1 can be refined to more carefully bound the per-query computation costs, but the above statement is enough to help prove the asymptotic results in Claim 8, leading to the efficient-simulation properties in Theorem 2.

6 Steepener analysis: first steps

Our goal in the following sections is to prove:

Theorem 3. *For our choice $N := 4^{10^{10}}$, the algorithm A_{st} in Section 5 is an admissible steepener (as in Def. 11).*

Combined with Theorem 2, this will achieve our main goal of the paper.

To prove Theorem 3, for any given 4-tuple $(\alpha, \beta, s_0, s' = 10^{-4}\sqrt{N}s_0)$ as in Def. 11, where

$$s_0\Delta \;=\; s_0(\alpha - \beta) \;=\; 1000 \;,$$

and for any (α, β, s_0)-ensemble

$$Ens \;=\; (X, Y, z) \;,$$

as in Def. 5, we must show that Eq. (6) holds. (We use "s_0" because we will be considering exponential moments with several different exponents, and wish to emphasize that this value is our starting point.) In the remainder of our work we fix attention upon one such ensemble, and study the behavior of $A_{st}^X(Y)$ on its first $N - \mathfrak{e}$ queries as treated in Eq. (6). We now give some central definitions for our analysis.

Definition 22. *We define some* <u>*random variables*</u>*, determined by the outcomes of (X, Y) and the execution of $A_{st}^X(Y)$:*

1. *Let $y^* \in ([0,1] \cup \{*\})^N$ be the partial view of Y for algorithm $A_{st}^X(Y)$ after $N - \mathfrak{e}$ queries and let*
$$u' \;:=\; u[\mathbf{k}_1 \leftarrow Y_{\mathbf{k}_1}] \;=\; [y^* \searrow X]$$
be its partial view of Y relative to baseline X after $N - \mathfrak{e}$ queries. Note that by definition,
$$\mathsf{MA}_\ell(u') \;=\; LB_{\mathsf{MA}_\ell}(y^*; X) \;.$$

2. *Let*
$$T^* \;\subset\; [N]$$
be an optimal M-tree for input $z \in [0,1]^N$ to MA_ℓ, i.e., one for which $\mathsf{MA}_\ell(z) = \mathrm{Avg}_{e \in T^*}(z_e)$. (We arbitrarily choose and fix one such M-tree throughout the analysis.)

3. *Define*
$$R \;:=\; \sqrt{N} \cdot [\mathsf{MA}_\ell(u') - \mathrm{Avg}_{e \in T^*}(u_e)] \;,$$
noting that $R \geq 0$ since MA_ℓ is monotone, $u' \geq u$, and $\mathsf{MA}_\ell(u) \geq \mathrm{Avg}_{e \in T^*}(u_e)$.

6.1 Exponential quantities of interest; main claimed bounds

Next we lay out a flexible, general definition with which to study exponential moments of different quantities associated with an execution of $A_{st}^X(Y)$:

Definition 23. *1. For $S \subseteq [N]$, and for $(X,Y) \sim Ens$ as presented to A_{st}, we define the random variables*

$$G_S^X \;:=\; \exp\left(10^{-4}s_0\left(\sum_{e\in S}(z_e - X_e)\right)\right) \;\;,\quad G_S^Y \;:=\; \exp\left(10^{-4}s_0\left(\sum_{e\in S}(z_e - Y_e)\right)\right) \;.$$

2. Also define the random variable

$$V \;:=\; G^Y_{(T^*\cap I)-\{\mathbf{i},\mathbf{j}\}} \cdot G^X_{(T^*-I)\cup(T^*\cap\{\mathbf{i},\mathbf{j}\})} \cdot \exp(-10^{-4}s_0 R) \;,$$

noting that, by definition of u, u', R, we have

$$V \;=\; \exp\left(10^{-4}s_0\left(\left(\sum_{e\in T^*}(z_e - u_e)\right) - R\right)\right) \;=\; \exp\left(10^{-4}s_0\sqrt{N}\left(\mathsf{MA}_\ell(z) - \mathsf{MA}_\ell(u')\right)\right) \;.$$

Our **central goal** of establishing Eq. (6) for $A^X(Y)$ on $Ens = (X,Y,z)$, thus corresponds precisely to showing

$$\begin{aligned}
\mathbb{E}[V] \;&\leq\; \exp\left(10^{-4}s_0\sqrt{N}\cdot\left[\frac{N-\mathfrak{e}+c}{N}\cdot\beta + \frac{\mathfrak{e}-c}{N}\cdot\alpha\right]\right) \\
&=\; \exp\left(10^{-4}s_0\left[\left(\sqrt{N} - 2^{30} - \frac{1-c}{\sqrt{N}}\right)\cdot\beta + \left(2^{30} + \frac{1-c}{\sqrt{N}}\right)\cdot\alpha\right]\right) \;. \qquad (24)
\end{aligned}$$

Our proof essentially consists of three parts, aimed at proving the following three bounds.

Claim 2. $\mathbb{E}[V|\mathbf{i},\mathbf{j} \notin T^*] \;\leq\; \exp\left[10^{-4}s_0((\sqrt{N}-2^{30})\beta + 2^{30}\alpha)\right]$.

Claim 3. *We have*

$$\mathbb{E}[V|\mathbf{i}\in T^*] \;\leq\; \exp\left[10^{-4}s_0((\sqrt{N}-2^{30}-1+.65)\beta + (2^{30}+1-.65)\alpha)\right] \;.$$

Claim 4. *We have*

$$\mathbb{E}[V|\mathbf{j}\in T^*] \;\leq\; \exp\left[10^{-4}s_0\left((\sqrt{N}-2^{30}-1+.88)\beta + (2^{30}+1-.88)\alpha\right)\right] \;.$$

The first claim above is simplest, and is proved below. The other two (particularly Claim 3) are more involved; we prove Claim 3 in Section 8 and Claim 4 in Section 7. Proving these claims will allow us to finish the argument, as follows:

Proof of Theorem 3. We claim that

$$\begin{aligned}
\mathbb{E}[V] \;&=\; \Pr[\mathbf{i}\in T^*]\cdot\mathbb{E}[V|\mathbf{i}\in T^*] \;+\; \Pr[\mathbf{j}\in T^*]\cdot\mathbb{E}[V|\mathbf{j}\in T^*] \;+\; \Pr[\mathbf{i},\mathbf{j}\notin T^*]\cdot\mathbb{E}[V|\mathbf{i},\mathbf{j}\notin T^*] \\
&=\; N^{-.5}\cdot\{\,\mathbb{E}[V|\mathbf{i}\in T^*] + \mathbb{E}[V|\mathbf{j}\in T^*]\,\} + (1-2N^{-.5})\cdot\mathbb{E}[V|\mathbf{i},\mathbf{j}\notin T^*] \;. \qquad (25)
\end{aligned}$$

Above, the first equality holds since $\mathbf{i},\mathbf{j}$ cannot both be in T^*, and the second holds because $\mathbf{i},\mathbf{j}$ are individually uniform over I (Prop. 12) and

$$\frac{|T^*\cap I|}{|I|} \;=\; \frac{\sqrt{N}-2^{30}}{N-2^{30}\sqrt{N}} \;=\; N^{-.5} \;.$$

Let $a_{\mathbf{i}}, a_{\mathbf{j}}, a_{\emptyset}$ denote the expected-value upper bounds provided by Claims 3, 4, and 2 respectively; from Eq. (25) we have

$$\begin{aligned}\mathbb{E}[V] \;\le\;& N^{-.5} \cdot \{\, a_{\mathbf{i}} + a_{\mathbf{j}} \,\} + (1 - 2N^{-.5}) \cdot a_{\emptyset} \\ \le\;& 2N^{-.5} a_{\mathbf{i}} + (1 - 2N^{-.5}) a_{\emptyset} \;,\end{aligned} \tag{26}$$

with the last step (which is crude but suffices for our purpose) holding since $a_{\mathbf{i}} \ge a_{\mathbf{j}}$.

Next, we make use of Lemma 3. Writing $a_{\mathbf{i}} = e^y > e^x = a_{\emptyset}$, we have

$$y - x \;=\; 10^{-4} s_0 (.35\Delta) \;\le\; .1 \;,$$

the last step since $s_0 \Delta = 1000$. So we can conclude from Lemma 3 (with $q := 2N^{-.5}$) that

$$\begin{aligned}2N^{-.5} a_{\mathbf{i}} + (1 - 2N^{-.5}) a_{\emptyset} \;\le\;& \exp[1.1qy + (1 - 1.1q)x] \\ =\;& \exp[x + 1.1q(y - x)] \\ =\;& \exp\left[(10^{-4} s_0) \cdot [(\sqrt{N} - 2^{30})\beta + 2^{30}\alpha) + .77\Delta/\sqrt{N}]\right] \\ =\;& \exp\left[(10^{-4} s_0) \cdot [(\sqrt{N} - 2^{30} - .77/\sqrt{N})\beta + (2^{30} + .77/\sqrt{N})\alpha]\right] \;,\end{aligned}$$

which establishes Eq. (24) and therefore Eq. (6), with $c = .33$. □

Now, as promised, we prove Claim 2.

Proof of Claim 2. Fix any outcome $\mathcal{I} = I$ that can occur conditioned on $[\mathbf{i}, \mathbf{j} \notin T^*]$ (actually this conditioning does not affect the support or distribution of $\mathcal{I}$, though we will not need this fact). Under the extended conditioning $[\mathcal{I} = I \wedge \mathbf{i}, \mathbf{j} \notin T^*]$, the random variable V satisfies

$$\begin{aligned}V \;\le\;& \exp\left((10^{-4} s_0) \left(\sum_{e \in T^* \cap I} (z_e - Y_e) + \sum_{e \in (T^* - I)} (z_e - X_e) \right) \right) \\ =\;& \prod_{e \in T^* \cap I} \exp\left[(10^{-4} s_0)(z_e - Y_e)\right] \;\cdot\; \prod_{e \in (T^* - I)} \exp\left[(10^{-4} s_0)(z_e - X_e)\right] \;,\end{aligned}$$

using the fact that $R \ge 0$. Using the independence of the ensemble coordinates, it follows that

$$\begin{aligned}\mathbb{E}[V | \mathcal{I} = I \wedge \mathbf{i}, \mathbf{j} \notin T^*] \;\le\;& \prod_{e \in T^* \cap I} \mathbb{E}[e^{(10^{-4} s_0)(z_e - Y_e)}] \;\cdot\; \prod_{e \in (T^* - I)} \mathbb{E}[e^{(10^{-4} s_0)(z_e - X_e)}] \\ \le\;& \exp\left((10^{-4} s_0)\beta\right)^{|T^* \cap I|} \cdot \exp\left((10^{-4} s_0)\alpha\right)^{|T^* - I|} \;,\end{aligned}$$

since (X, Y, z) is an (α, β, s_0)-ensemble and therefore also (by Prop. 6) a $(\alpha, \beta, 10^{-4} s_0)$-ensemble. The last line is $\exp[(10^{-4} s_0)((\sqrt{N} - 2^{30})\beta + 2^{30}\alpha)]$, since $|T^* \cap I| = \sqrt{N} - 2^{30}$ (Prop. 12).

As $\mathcal{I} = I$ was an arbitrary possible further outcome to $\mathcal{I}$, this proves Claim 2. □

6.2 Conditions for lower-bounding R

We will identify several conditions under which R from Def. 22 can be lower-bounded, which (with further work) will allow us to analyze the random variable V and establish the Steepener property. Here, we give the two most basic such conditions, with which other more specific conditions will be identified later. The first such condition is that our "key" $(N-\mathfrak{e})^{th}$ query is made to a coordinate in T^* (as in Def. 22).

Lemma 14. $R \ \geq\ \mathbf{1}_{[\mathbf{k}_1 \in T^*]} \cdot (Y_{\mathbf{k}_1} - X_{\mathbf{k}_1})$.

Proof. In the event $[\mathbf{k}_1 \in T^*]$ we have $u'_{\mathbf{k}_1} = Y_{\mathbf{k}_1}$, which implies $\mathsf{MA}_\ell(u') - \mathrm{Avg}_{T^*}(u_e) \geq \mathrm{Avg}_{T^*}(u'_e) - \mathrm{Avg}_{T^*}(u_e) = (Y_{\mathbf{k}_1} - X_{\mathbf{k}_1})/\sqrt{N}$. □

The second source of lower-bounds on R is from the event that one of $\mathbf{i}$ or $\mathbf{j}$ lies in T^*, but that this coordinate $\mathbf{i}$ is not in the vanguard set $\mathrm{Van}(u)$ for u, and in fact has positive sum of negative slacks appearing along $P_\mathbf{i}(u)$.

Lemma 15. $R \ \geq\ \mathbf{1}_{[\mathbf{i} \in T^*]} \cdot S^-_\mathbf{i}(u) \ +\ \mathbf{1}_{[\mathbf{j} \in T^*]} \cdot S^-_\mathbf{j}(u)$.

Proof. We first consider the $\mathbf{i}$ coordinate. By Lemma 13, with $u[\mathbf{i} \leftarrow z_\mathbf{i}]$ taking the role of y and $y' := u$, we have

$$\begin{aligned} z_\mathbf{i} - X_\mathbf{i} \ &=\ (S^-_\mathbf{i}(u) - S^-_\mathbf{i}(u[\mathbf{i} \leftarrow z_\mathbf{i}])) \ +\ 2^\ell[\mathsf{MA}_\ell(u[\mathbf{i} \leftarrow z_\mathbf{i}]) - \mathsf{MA}_\ell(u)] \\ &\geq\ S^-_\mathbf{i}(u) \ +\ 2^\ell[\mathsf{MA}_\ell(u[\mathbf{i} \leftarrow z_\mathbf{i}]) - \mathsf{MA}_\ell(u)] \ . \end{aligned} \tag{27}$$

Then,

$$\begin{aligned} \mathsf{MA}_\ell(u') \ &\geq\ \mathsf{MA}_\ell(u) \\ &=\ \mathsf{MA}_\ell(u[\mathbf{i} \leftarrow z_\mathbf{i}]) - [\mathsf{MA}_\ell(u[\mathbf{i} \leftarrow z_\mathbf{i}]) - \mathsf{MA}_\ell(u)] \\ &\geq\ \Big[\mathsf{MA}_\ell(z) + (z_\mathbf{i} - X_\mathbf{i})/\sqrt{N} - \mathrm{Avg}_{e \in T^*}(z_e - u_e)\Big] \ -\ [\mathsf{MA}_\ell(u[\mathbf{i} \leftarrow z_\mathbf{i}]) - \mathsf{MA}_\ell(u)] \\ &\geq\ [\mathsf{MA}_\ell(z) \ - \mathrm{Avg}_{e \in T^*}(z_e - u_e)] \ +\ S^-_\mathbf{i}(u)/\sqrt{N} \ , \end{aligned}$$

where we used Eq. (27) in the last step. Thus,

$$R \ \geq\ \mathbf{1}_{[\mathbf{i} \in T^*]} \cdot S^-_\mathbf{i}(u) \ .$$

The same reasoning applies to the coordinate $\mathbf{j}$, giving the analogous inequality with $\mathbf{j}$ in place of $\mathbf{i}$; then, as $\mathbf{i}, \mathbf{j}$ can never both lie in T^* (since $\mathbf{i} \perp \mathbf{j}$ and T^* is an M-tree), we infer the stronger statement of the Lemma. □

6.3 Contexts

Here we describe natural partial outcomes to the coordinates of $(X, Y) \sim \mathit{Ens}$.

Definition 24. *1. For $i \in [N]$, an i-**context** is a pair of vectors (x', y'), each with coordinates indexed by the set $[N] - \{i\}$, for which the joint outcome*

$$(X)_{[N]-\{i\}} \ =\ x' \ , \quad (Y)_{[N]-\{i\}} \ =\ y'$$

is supported under $(X, Y) \sim \mathit{Ens}$.

2. *For contrast, by an Ens-**outcome** we will simply mean a full outcome (x, y) to (X, Y), supported under Ens.*

3. *For an Ens-outcome (x, y), the i-**context associated with** (x, y) is the projection*

$$x' \;:=\; (x)_{[N]-\{i\}}\ , \quad y' \;:=\; (y)_{[N]-\{i\}}\ .$$

4. *For brevity, in an execution of $A^X_{st}(Y)$, we denote conditioning on a joint outcome to $(\mathcal{I}, \mathbf{j}) = (I, j)$ and upon the j-context (x', y') with the event notation*

$$[I, j, x', y'] \;=\; \Big[\; (\mathcal{I}, \mathbf{j}) \;=\; (I, j) \;\wedge\; (X)_{[N]-\{j\}} \;=\; x' \quad \wedge \quad (Y)_{[N]-\{j\}} \;=\; y' \;\Big]\ .$$

We chiefly use it in conditional expectations, e.g. $\mathbb{E}[U|I, j, x', y']$. We also sometimes condition additionally on $[\mathbf{i} = i]$, e.g. using $\mathbb{E}[U|I, j, i, x', y']$. We use similar notation when (x', y') is an i-context (not a j-context), when this is clear from the surrounding discussion.

5. *Also for brevity, when an index i is clear from the discussion, we write $(x', y') \sim Ens$ to indicate that the i-context (x', y') is sampled as $((X)_{[N]-\{i\}}, (Y)_{[N]-\{i\}})$.*

7 Steepener analysis: the "j-critical" case

In the execution of $A^X(Y)$ on (X, Y) drawn from *Ens*, let us fix an index

$$j \;\in\; T^*$$

and a Γ-supported partial outcome $[(\mathcal{I}, \mathbf{j}) = (I, j)]$; we refer to such a partial outcome, in which $\mathbf{j} \in T^*$, as **"j-critical"**. Given, additionally, a j-context (x', y'), we will give a sufficient condition on possible further outcomes $[\mathbf{i} = i]$ under which it is forced that $[\mathbf{j} = \mathbf{k}_1]$, that is, under which $\mathbf{i}$ is not chosen for the $(N - \mathfrak{e})^{th}$ query. The conclusion $[\mathbf{j} = \mathbf{k}_1]$ in turn implies that $R \geq (Y_{\mathbf{j}} - X_{\mathbf{j}})$, by Lemma 14 and the fact that $j \in T^*$. Then, we will argue that, regardless of the choice of (x', y') as above, *almost all possible values* for i obey this condition—a finding that will play a key role in establishing Claim 4.

7.1 Variant-conditions

Definition 25. *Given a Γ-supported pair (I, j) with $j \in T^*$, a j-context (x', y'), and for a given $i \in I$ with $i \perp T^*$, we define a set of **"Variant-Conditions"** for i. (These are variants of the Selection-Conditions from the algorithm A_{st}.)*

First, arbitrarily fix any further outcome $[X_j = \tilde{x}_j]$ possible under the ensemble Ens, and let $\hat{u}'$ be the input defined (independent of i) by $\hat{u}'_j := \tilde{x}_j$, and that elsewhere agrees with y' on index set $I - \{j\}$ and with x' on other coordinates. We define the Variant-Conditions in terms of this $\hat{u}'$. Let us say that i obeys the Variant-Condition:

(a)' if P_i is dominant with respect to $\hat{u}'$ at gates $p^{i,2t}$ for $t \in [1, C_1 - 1]$;[9]

[9]Note that here we exclude the root $p^{i,0}$, in distinction to Selection-Condition (a) in A_{st}.

(b)' if P_i is dominant with respect to $\hat{u}'$ on at least $.51\ell - 1$ Max *gates among $p^{i,2t}$ for $t \in [1, \ell - 1]$;*

(c)' if (recalling notation from Def. 17) there exists a positive value

$$\gamma \ > \ .001 \cdot \tilde{S}_i^-(\hat{u}') \ \geq \ 0 \ ,$$

and a set G of at least $.01\ell$ Max *gates pendant to P_i, such that for each $g \in G$,*

$$\mathrm{Dec}(g; \hat{u}') \ \in \ [\gamma, 1000\gamma] \ . \tag{28}$$

Definition 26. *[Type-1 bad index sets] Given Γ-supported pair (I, j) with $j \in T^*$ and a j-context (x', y'), define the set*

$$B^1_{I,j,x',y'} \ := \ \{i \in I : \ \ i \perp j \ \ \wedge \ \ i \text{ obeys one or more of Variant-Conditions (a)'-(c)'}\} \ ,$$

the Conditions defined with respect to (I, j, x', y').

Claim 5. *Suppose that for Γ-supported (I, j) with $j \in T^*$ and j-context (x', y') we have $i \notin B^1_{I,j,x',y'}$. Then in the execution of $A^X_{st}(Y)$, we have the implication*

$$[(I, j, x', y') \ \wedge \ \mathbf{i} = i] \quad \Longrightarrow \quad R \ \geq \ Y_{\mathbf{j}} - X_{\mathbf{j}} \ ,$$

Proof. By Lemma 14, it suffices to show that under our assumptions,

$$[(I, j, x', y') \ \wedge \ \mathbf{i} = i] \quad \Longrightarrow \quad [\mathbf{k}_1 = \mathbf{j}] \ . \tag{29}$$

Suppose that some possible joint outcome to (X, Y) and execution of $A^X_{st}(Y)$ satisfies

$$[(I, j, x', y') \wedge \mathbf{i} = i] \ \wedge \ [\mathbf{k}_1 = \mathbf{i}] \ .$$

We will argue that i obeys one or more of Variant-Conditions (a)'-(c)', contrary to the assumption $i \notin B^1_{I,j,x',y'}$. This will prove Eq. (29) since $\mathbf{k}_1 \in \{\mathbf{i}, \mathbf{j}\}$.

The condition $[\mathbf{k}_1 = \mathbf{i}]$ is triggered for $A^X_{st}(Y)$ precisely when the pair $(u, \mathbf{i})$ obeys one of Selection-Conditions (a)-(c), with u as in the algorithm definition. Let $(X_j, Y_j) = (x_j, y_j)$ in the outcome under consideration. Letting u be as in $A^X_{st}(Y)$ in this execution, define

$$\hat{u} \ := \ u[i \leftarrow y'_i] \ ,$$

an input obtained by incrementing u on coordinate i from x'_i to y'_i. As the formula defining MA_ℓ is monotone, any gate g along the path P_i takes on at least as large a value under input $\hat{u}$ as under u; while any gate g not lying on P_i takes the same values on both inputs. Thus for every Max gate g on P_i for which P_i is dominant on input u, P_i is also dominant at g on $\hat{u}$. Similarly, we also have that $S_i^-(\hat{u}) \leq S_i^-(u)$ (the contribution to this sum from any gate g cannot increase by passing from u to $\hat{u}$).

Next, note that, for $\hat{u}'$ defined with respect to I, j, x', y' as in Def. 25, we have

$$\hat{u}' \ = \ \hat{u}[j \leftarrow \tilde{x}_j] \ ,$$

a change in a single coordinate (that may be an increase or decrease). As $i \perp j$, the paths P_i and P_j have only the root gate $r = p^{i,0}$ in common; changing the input from $\hat{u}$ to $\hat{u}'$ can affect the dominance of P_i for its Max gates only at this root gate. It follows directly that, if (u, i) obey Selection-Condition (a) (respectively, (b)) in the execution of $A^X_{st}(Y)$, then i obeys Variant-Condition (a)' (respectively, (b)') for (I, j, x', y'). (Observe that in each case the possible loss of a dominant root gate for P_i is tolerated—the root gate is ignored by Variant-Condition (a)', while the threshold number of dominant gates is lower by one in Variant-Condition (b)' compared to Selection-Condition (b).)

The quantity $\tilde{S}^-_i(\hat{u}')$ from Def. 17, which ignores the root gate, is unaffected by the change—we have

$$\tilde{S}^-_i(\hat{u}') \;=\; \tilde{S}^-_i(\hat{u}) \;\leq\; S^-_i(\hat{u}) \;\leq\; S^-_i(u) \;.$$

And, all Max gates g' *pendant* to P_i have no dependence upon coordinates i and j, and thus satisfy $\mathrm{Dec}(g'; u) = \mathrm{Dec}(g'; \hat{u}')$. From this, we see that, if (u, i) obey Selection-Condition (c) as witnessed by gate set G and value γ, the same pair (G, γ) witness that $\hat{u}'$ obeys Variant-Condition (c)'. This completes the proof. □

Lemma 16. *For (I, j, x', y') as in Def. 26 we have*

$$|B^1_{I,j,x',y'}| \;\leq\; .01(N - 2^{30}\sqrt{N})/2 \;.$$

Proof. The number of $i \in I$ with $i \perp j$ that obey Variant-Condition (a)' for (I, j, x', y') is bounded by the total number of $i \in [N]$ with $i \perp j$ whose paths P_i are dominant with respect to $\hat{u}'$ (an input independent of i) on gates $p^{i,2t}$ for $t \in [1, C_1 - 1]$. Such paths must pass from this $p^{i,2t}$ to a child Avg gate with strictly larger value on $\hat{u}'$ (of the two children). Then, simple counting reveals there are at most $2^{-(C_1-1)} \cdot N/2 \leq 2^{-(C_1-2)} \cdot (N - 2^{30}\sqrt{N})/2$ such i, where we used the largeness of $N \geq 10^{21}$.

Similarly, the number of $i \in I, i \perp j$ obeying Variant-Condition (b)' is at most $\rho N/2 \leq 2\rho(N - 2^{30}\sqrt{N})/2$, where $\rho \in (0, 1)$ is defined, for i.i.d. .5-biased Bernoulli variables $W_1, \ldots, W_{\ell-1}$, by

$$\rho \;=\; \Pr[W_1 + \ldots + W_{\ell-1} \;\geq\; .51\ell - 1] \;.$$

As $\ell > 500$, we have $.51\ell - 1 > .505(\ell - 1)$. By Lemma 1, item 1 (with $n = \ell - 1, p = .5, \delta \geq .01$), we get

$$\rho \;\leq\; \left[\frac{e^{.01}}{(1.01)^{1.01}}\right]^{.5(\ell-1)} \;<\; (1 - 10^{-5})^{\ell-1} \;.$$

Variant-Condition (c)' needs more work to analyze. For our fixed (x', y', I, j), if i obeys Variant-Condition (c)' and $t \in [0, \ell - 2]$, say that (c)' is **"triggered by t"** (an event we also denote E_t) if in Variant-Condition (c)', the set G of Max gates (pendant to P_i) can be chosen so that its closest gate $g_0 \in G$ to the root r, is pendant to P_i at the Avg node $p^{\hat{i},2t+1}$. If (c)' is to hold for i, it must clearly be triggered by at least one such t.

We now upper-bound $\Pr[E_t]$ over an index $\hat{i} \in [N]$ chosen uniformly subject to $\hat{i} \perp j$, for each t. A union bound over these t will lead to an upper-bound on the number of i obeying (c)'. (The additional constraint in the definition of $B^1_{I,j,x',y'}$ that $i \in I$ cannot increase the overall count.)

So fix such a t, consider our $\hat{i} \perp j$ as chosen by a descending random walk, and condition on any possible outcome for the Max gate $p^{\hat{i},2(t+1)}$, which determines $p^{\hat{i},2t+1}$ and the Max gate $g_0 \in G$ with which (c)' is triggered by t. Let $\gamma_0 := \mathrm{Dec}(g_0; \hat{u})$. Say that a Max gate g in $\mathbf{T}$ is "g_0-similar" if

$$\mathrm{Dec}(g; \hat{u}) \ \in \ \left[\frac{\gamma_0}{1000},\ 1000 \cdot \gamma_0\right] \ .$$

Say that $P_{\hat{i}}$ "approaches" a Max gate g if the parent of g' lies in P_i. Say that $P_{\hat{i}}$ "runs afoul" of Max gate g if g lies on $P_{\hat{i}}$, and $P_{\hat{i}}$ is beaten at g with respect to $\hat{u}$. If (c)' is to be triggered by t (under our conditioning on $p^{\hat{i},2(t+1)}$), then the random path $P_{\hat{i}}$ must, after passing through $p^{\hat{i},2(t+1)}$, approach at least $.01\ell$ Max gates which are g_0-similar.

However, the path $P_{\hat{i}}$ must also run afoul of <u>few such gates</u>. For, each such step increases the sum $\tilde{S}^{-}_{\hat{i}}(\hat{u})$ by at least $\gamma_0/1000$. If $P_{\hat{i}}$ runs afoul of more than 10^6 such gates, this rules out any possible value of γ in (c)' for a set G that is to include g_0.

For our above settings of t and $p^{\hat{i},2(t+1)}$, let us define $\mathcal{H}$ as the full *subtree*, rooted at $p^{\hat{i},2(t+1)}$, of the MA_ℓ formula tree $\mathbf{T}$. Now,

- Let M_{good} be the set of all Max gates in $\mathcal{H}$ that are *sibling* to a g_0-similar Max gate;
- Let M_{bad} be the set of all Avg gates h in $\mathcal{H}$ whose parent gate $g = \max(h, h')$ is g_0-similar, with $h(\hat{u}) < h'(\hat{u})$.

Note that these two marked sets obey the assumptions of Lemma 11. Also, our work above implies that for (c)' to be triggered by t, when $P_{\hat{i}}$ passes through the node $p^{\hat{i},2(t+1)}$, the path $P_{\hat{i}}$ (whose restriction to $\mathcal{H}$ is a uniform path to a leaf) must contain at least $a := .01\ell$ vertices of M_{good}, and at most 10^6 elements of M_{bad} (which is $< .1a$, since $\ell = 10^{10}$). By Lemma 11, this occurs with probability $\leq (.998)^{.01\ell}$. As $p^{\hat{i},2(t+1)}$ was arbitrary, we conclude $\Pr[E_t] \leq (.998)^{.01\ell}$.

Then, the probability that (c)' is triggered by *any* $t \in [0, \ell - 2]$ is $\leq \ell(.998)^{.01\ell}$. The *number* of $i \in I, i \perp j$ obeying (c)' is therefore at most $\leq \ell(.998)^{.01\ell}(N/2) \leq 2\ell(.998)^{.01\ell}(N - 2^{30}\sqrt{N})/2$. Summing our bounds for (a)', (b)', (c)' , the total number of indices $i \in B^1_{I,j,x',y'}$ is at most

$$\left[\ 2^{-(C_1-2)} + (1 - 10^{-5})^{\ell-1} + 2\ell(.998)^{.01\ell}\ \right] \cdot (N - 2^{30}\sqrt{N})/2 \ < \ .01(N - 2^{30}\sqrt{N})/2 \ ,$$

the inequality holding since $C_1 = 1000,\ \ell = 10^{10}$. This proves the Lemma. □

7.2 Exponential-moment bounds

Lemma 17. *Let (I, j) be Γ-supported with $j \in T^*$. Consider a j-context sampled as $(x', y') \sim$ Ens, and for each $i \in I$ with $i \perp j$, define*

$$\zeta_i^{I,j} \ := \ \mathbb{E}_{(x',y')\sim Ens}\left[\ \mathbf{1}[i \in B^1_{I,j,x',y'}\ ] \cdot G^Y_{(T^*\cap I)-\{j\}} \cdot G^X_{T^*-I}\ \right] \ . \tag{30}$$

<u>Then</u>, averaging over i, we have

$$\frac{2}{N - 2^{30}\sqrt{N}} \sum_{i \in I: i \perp T^*} \zeta_i^{I,j} \ \leq \ .01 \cdot \exp\left(10^{-4} s_0 \left[(\sqrt{N} - (2^{30}+1))\beta + 2^{30}\alpha\right]\right) \ .$$

Proof. The quantity $G^Y_{(T^*\cap I)-\{j\}} \cdot G^X_{T^*-I}$ is identical for every $i \in I$ with $i \perp T^*$; thus

$$\begin{aligned}\sum_{i\in I: i\perp T^*} \zeta_i^{I,j} &= \mathbb{E}_{(x',y')\sim Ens}\left[|B^1_{I,j,x',y'}| \cdot G^Y_{(T^*\cap I)-\{j\}} \cdot G^X_{T^*-I}\ \right] \\ &\le .01\frac{(N-2^{30}\sqrt{N})}{2} \prod_{e\in(T^*\cap I)-\{j\}} \mathbb{E}[\exp(10^{-4}s_0(z_e - Y_e))] \cdot \prod_{e\in(T^*-I)} \mathbb{E}[\exp(10^{-4}s_0(z_e - X_e))]\ ,\end{aligned}$$

using our size bound from the conclusion of Lemma 16 and the ensemble independence property. As an (α, β, s_0)-ensemble, and therefore (Prop. 6) an $(\alpha, \beta, 10^{-4}s_0)$-ensemble, the above is at most

$$.01\frac{(N-2^{30}\sqrt{N})}{2} \cdot \exp\left(10^{-4}s_0\left[(\sqrt{N}-(2^{30}+1))\beta + 2^{30}\alpha\right]\right)\ ,$$

from which the conclusion follows. ☐

Proof of Claim 4. Conditioned on $[\mathbf{j} \in T^*]$, the pair $(\mathcal{I}, \mathbf{j})$ is uniform over its possible outcomes for which $\mathbf{j} \in T^* \cap I$. Letting Γ^* be the induced distribution on such pairs, we have

$$\mathbb{E}[V|\mathbf{j} \in T^*] \ = \ \mathbb{E}_{(I,j)\sim\Gamma^*}[\ \mathbb{E}[V|(\mathcal{I},\mathbf{j}) = (I,j)]\ ]\ . \tag{31}$$

Fix any such (I, j). Conditioned on this pair, $\mathbf{i}$ is uniform over $i \in I$ for which $i \perp T^*$. Thus

$$\mathbb{E}[V|(\mathcal{I},\mathbf{j}) = (I,j)] \ = \ \frac{2}{N-2^{30}\sqrt{N}} \sum_{i\in I, i\perp T^*} \mathbb{E}[V|(\mathcal{I},\mathbf{i},\mathbf{j}) = (I,i,j)]\ , \tag{32}$$

since $|\{i \in I : i \perp T^*\}| = |I|/2 = (N - 2^{30}\sqrt{N})/2$.

We fix $i \in I$ with $i \perp T^*$, and further expand $\mathbb{E}[V|(\mathcal{I},\mathbf{i},\mathbf{j}) = (I,j,i)]$, which we also denote $\mathbb{E}[V|I,j,i] = \mathbb{E}[V|W]$ where we let $W := [(\mathcal{I},\mathbf{i},\mathbf{j}) = (I,j,i)]$. Note that conditioned on W, we have

$$V \ = \ AB\ , \qquad \text{where}$$

$$A \ := \ G^Y_{(T^*\cap I)-\{j\}} \cdot G^X_{T^*-I}\ , \qquad B \ := \ \exp(10^{-4}s_0(z_j - x_j - R))\ .$$

Preparing for an application of Lemma 2 (under our fixed I, j, i), let us consider the auxiliary random variable $\mathbf{t} := (x', y')$ giving the j-context associated with the outcome $(X, Y) \sim Ens$. Let $S := \{(x', y') : i \in B^1_{I,j,x',y'}\}$. This $\mathbf{t}$ is independent of W. We have, by our definitions,

$$\mathbb{E}[\mathbf{1}[(x',y') \in S] \cdot A|W] \ = \ \zeta_i^{I,j}\ ,$$

as defined in Lemma 17, using that the expectation in Eq. (30) is of a variable unaffected by $(\mathcal{I},\mathbf{j},\mathbf{i})$ and hence by W.

Let us suppose for the moment that our fixed i satisfies

$$\zeta_i^{I,j} \ \le \ .1\left(\prod_{e\in(T^*\cap I)-\{j\}} \mathbb{E}[\exp(10^{-4}s_0(z_e - Y_e))]\right)\left(\prod_{e\in(T^*-I)} \mathbb{E}[\exp(10^{-4}s_0(z_e - X_e))]\right)\ ,$$

which if true also gives $\zeta_i^{I,j} \leq p \cdot a$, where

$$p \ := \ .1 \ , \qquad a \ := \ \exp\left(10^{-4}s_0((\sqrt{N}-2^{30}-1)\beta + 2^{30}\alpha)\right) \ ,$$

and where we note $\mathbb{E}[A|W] \leq a$ by the ensemble conditions. The random variable B has expected value at most $b := e^{10^{-4}s_0\alpha}$ conditioned on $[W \wedge \mathbf{t} = (x', y')]$ for any j-context, and at most $b' := e^{10^{-4}s_0\beta}$ conditioned upon $[W \wedge \mathbf{t} = (x', y')]$ for any j-context (x', y') satisfying $i \notin B^1_{I,j,x',y'}$ (by Lemma 5). Moreover, using the ensemble property, A, B are independent conditional upon $[W \wedge \mathbf{t} := (x', y')]$, for any outcome to $\mathbf{t}$, which determines the value of A. We can thus use Lemma 2 applied to $\mathbb{E}[AB|W] = \mathbb{E}[V|I, j, i]$ with the values given, to find

$$\mathbb{E}[V|I,j,i] \ \leq \ a[pb + (1-p)b'] \ . \tag{33}$$

Now letting $b = e^f, \ b' = e^g$, we have $f - g = (10^{-4}s_0)\Delta = (10^{-4}s_0)(\alpha - \beta) \leq .1$. By Lemma 3 and the definitions of b, b', we have

$$pb + (1-p)b' \ \leq \ \exp\left[10^{-4}s_0\,(1.1(.1)\alpha + (1-1.1(.1))\beta)\right] \ .$$

Then from Eq. (33) and our value of a, we have

$$\begin{aligned}
\mathbb{E}[V|I,j,i] \ &\leq \ \exp\left[10^{-4}s_0((\sqrt{N}-(2^{30}+1))\beta + 2^{30}\alpha)\right] \cdot \exp\left[10^{-4}s_0\,(.11\alpha + .89\beta)\right] \\
&= \ \exp\left[10^{-4}s_0((\sqrt{N}-2^{30}-.11)\beta + (2^{30}+.11)\alpha)\right] \ .
\end{aligned} \tag{34}$$

Recall that we assumed in the above that $\zeta_i^{I,j} \leq .1$. However, by Lemma 17 and a Markov bound, at most a .1 fraction of all $i \perp T^*$ with $i \in I$ satisfy $\zeta_i^{I,j} > .1$. And even for such i, we have

$$\begin{aligned}
\mathbb{E}[V|I,j,i] \ &\leq \ \mathbb{E}\left[G^Y_{(T^*\cap I)-\{j\}} \cdot G^X_{(T^*-I)\cup\{j\}}\right] \\
&\leq \ \exp\left[10^{-4}s_0((\sqrt{N}-2^{30}-1)\beta + (2^{30}+1)\alpha)\right] \ .
\end{aligned}$$

Combining this with Eqs. (32) and (34), we have

$$\mathbb{E}[V|(\mathcal{I}, \mathbf{j}) = (I, j)] \ \leq \ .1e^{\mu} + .9e^{\nu} \tag{35}$$

where

$$\mu \ := \ 10^{-4}s_0((\sqrt{N}-2^{30}-1)\beta + (2^{30}+1)\alpha) \ , \quad \nu \ := \ 10^{-4}s_0((\sqrt{N}-2^{30}-.11)\beta + (2^{30}+.11)\alpha) \ ,$$

and $0 < \mu - \nu \leq 10^{-4}s_0\Delta \leq .1$. By another application of Lemma 3, from Eq. (35) we get

$$\begin{aligned}
\mathbb{E}[V|(\mathcal{I}, \mathbf{j}) = (I, j)] \ &\leq \ \exp(.11\mu + .89\nu)) \\
&\leq \ \exp\left[10^{-4}s_0\left((\sqrt{N}-2^{30}-.22)\beta + (2^{30}+.22)\alpha\right)\right] \\
&= \ \exp\left[10^{-4}s_0\left((\sqrt{N}-2^{30}-1+.88)\beta + (2^{30}+1-.88)\alpha\right)\right] \ .
\end{aligned}$$

The outcomes I and $j \in T^* \cap I$ were arbitrary. Thus, from Eq. (31), we get the same upper bound on $\mathbb{E}[V|\mathbf{j} \in T^*]$, as needed, proving Claim 4. $\square$

8 Steepener analysis: the "i-critical" case

Our goal in this Section is to prove Claim 3.

For Γ-supported outcomes $[(\mathcal{I}, \mathbf{j}) = (I, j)]$ with

$$j \perp T^* ,$$

our focus is on analyzing further possible partial outcomes $[\mathbf{i} = i]$ with $i \in T^* \cap I$. Next we lay out some important terms for this study.

Definition 27. *Let (I, j, i) be Γ-supported and satisfy*

$$i \in T^* \cap I .$$

<u>Relative to a given i-context</u> (x', y'), we define the following (using Def. 15):

1. *Let*

   $$\tilde{u} \in [0, 1]^N$$

   be the input agreeing with y' on index set $I - \{i, j\}$, with x' on index set $\{j\} \cup ([N] - I)$, and setting $\tilde{u}_i := z_i$.

In the <u>case</u> where $i \in \mathrm{Van}(\tilde{u})$, we define additional quantities as follows:

2. *let $0 \le d_0 \le d_1 \le \ldots \le d_{\ell-1}$ be the multiset of values $\{ \mathrm{Dec}(p^{i,2t}; \tilde{u}) \}_{t \in [0, \ell-1]}$, listed in ascending value (<u>not</u> according to their order of appearance along P_i).*

3. *Let $\tilde{\gamma}$ be a <u>maximal</u> value having an associated set G of at least $.01\ell$ Max gates pendant to P_i, with each $g \in G$ satisfying*

   $$\mathrm{Dec}(g; \tilde{u}) \in [\tilde{\gamma}, 1000\tilde{\gamma}] .$$

   If no such value exists, let $\tilde{\gamma} := \perp$.

We now establish several conditions in which the quantity R from Def. 22 can be lower-bounded in the "**i**-critical" case under discussion.

Lemma 18. *Let $(I, j, i, x', y', \tilde{u})$ be as in Def. 27. Conditioned on*

$$(\mathcal{I}, \mathbf{i}, \mathbf{j}) = (I, i, j) \ , \ (X)_{[N]-\{i\}} = x' \ , \ (Y)_{[N]-\{i\}} = y' ,$$

we have the following items.

0. *If $i \notin \mathrm{Van}(\tilde{u})$, then $R \ge z_i - X_i$.*

Else, for the remaining items assume $i \in \mathrm{Van}(\tilde{u})$. For $d_0, \ldots, d_\ell$ as in Def. 27, we have:

1. $R \ \ge \ \mathbf{1}_{[z_i - X_i \ge d_0]} \cdot (z_i - X_i - d_0)$.

2. $R \ \ge \ \mathbf{1}_{[z_i - X_i < d_f]} \cdot (Y_i - X_i)$, *where $f := \lceil .49\ell \rceil$;*

3. *If d_0 is not equal to* $\mathrm{Dec}(p^{i,2t};\tilde{u})$ *for any* $t \in [0, C_1 - 1]$ *(i.e., the minimum decisiveness along* P_i *does not appear among the* C_1 Max *gates nearest the root), then* $R \geq (Y_i - X_i)$.

4. *Regardless of whether the condition in item 3 holds, if* $\tilde{\gamma} \neq \bot$*, then we have* $R \geq \mathbf{1}_{[z_i - X_i < d_0 + 1000\tilde{\gamma}]} \cdot (Y_i - X_i)$.

Proof. In all the items, we will use the basic "coordinate-decrement" relation

$$u \;=\; \tilde{u}[i \leftarrow X_i] \;, \tag{36}$$

valid under the conditioning given by $[I, j, i, x', y']$.

(0.) If $i \notin \mathrm{Van}(\tilde{u})$, then $\mathsf{MA}_\ell(u) = \mathsf{MA}_\ell(\tilde{u})$ by Lemma 13, item 2. Now,

$$\mathsf{MA}_\ell(\tilde{u}) \;\geq\; \mathrm{Avg}_{e \in T^*}(\tilde{u}_e) \;=\; \mathrm{Avg}_{e \in T^*}(u_e) \;+\; (z_i - X_i)/\sqrt{N} \;.$$

Also, we always have $\mathsf{MA}_\ell(u') = \mathsf{MA}_\ell(u[\mathbf{k}_1 \leftarrow Y_{\mathbf{k}_1}]) \geq \mathsf{MA}_\ell(u)$. Thus $R \geq z_i - X_i$, as claimed.

Henceforth we assume $i \in \mathrm{Van}(\tilde{u})$.

(1.) Suppose $z_i - X_i \geq d_0$. We have

$$\mathsf{MA}_\ell(u') \;\geq\; \mathsf{MA}_\ell(u) \;\geq\; \mathsf{MA}_\ell(\tilde{u}) - d_0/\sqrt{N} \;\geq\; \mathrm{Avg}_{e \in T^*}(u_e) + (\tilde{u}_i - u_i - d_0)/\sqrt{N} \;,$$

where for the second inequality we used Lemma 13, item 2, with $\delta_0 := d_0 > 0$. From this we get

$$R \;\geq\; \tilde{u}_i - u_i - d_0 \;=\; z_i - X_i - d_0 \;,$$

as needed.

(2.) Now suppose $z_i - X_i < d_f$. Applying Lemma 13, item 3 to the coordinate decrement in Eq. (36), we see that each Max gate $p^{i,2t}$ on P_i for which $\mathrm{Dec}(p^{i,2t};\tilde{u}) \geq d_f$, the path P_i is still dominant on $p^{i,2t}$ for input u. By our choice of f, fewer than $.49\ell$ Max gates on P_i are non-dominant on u.

Thus under the conditioning $[I, i, j, x', y']$ (and under our additional assumption in item 1), Selection-Condition (b) holds of u and $\mathbf{i}$ in the execution of $A^X_{st}(Y)$, so that we have $[\mathbf{k}_1 = \mathbf{i}]$. It then holds that

$$\mathsf{MA}_\ell(u') \;\geq\; \mathrm{Avg}_{e \in T^*}(u'_e) \;=\; \mathrm{Avg}_{e \in T^*}(u_e) \;+\; (Y_i - X_i)/\sqrt{N} \;.$$

$R \geq Y_i - X_i$ follows by work analogous to that in item 0.

(3.) By Lemma 13, item 4, our assumptions imply that P_i is dominant at each Max gate $p^{i,2t}$ for $t \in [0, C_1 - 1]$ not only on input $\tilde{u}$, but on $u = \tilde{u}[i \leftarrow X_i]$ as well. Thus, under our conditioning and assumptions, Selection-Condition (a) holds for u and $\mathbf{i}$ in the execution of $A^X_{st}(Y)$, so that $[\mathbf{k}_1 = \mathbf{i}]$, and then the lower bound on R follows as in item 2 above.

(4.) Suppose $z_i - X_i < d_0 + 1000\tilde{\gamma}$. By Lemma 13, item 1 applied to $u = \tilde{u}[i \leftarrow Z_i]$,

$$\begin{aligned} z_i - X_i \;&=\; (S^-_i(u) - S^-_i(\tilde{u})) + \sqrt{N}(\mathsf{MA}_\ell(\tilde{u}) - \mathsf{MA}_\ell(u)) \\ &=\; S^-_i(u) + \sqrt{N}(\mathsf{MA}_\ell(\tilde{u}) - \mathsf{MA}_\ell(u)) \;, \end{aligned} \tag{37}$$

since $i \in \mathrm{Van}(\tilde{u})$ implies that $S_i^-(\tilde{u}) = 0$.

If $z_i - X_i \le d_0$ then $z_i - X_i = \sqrt{N}(\mathsf{MA}_\ell(\tilde{u}) - \mathsf{MA}_\ell(u))$ by Lemma 12, item 1 applied to the root gate $p^{i,0}$. Then, Eq. (37) yields $S_i^-(u) = 0$. Otherwise, $\sqrt{N}(\mathsf{MA}_\ell(\tilde{u}) - \mathsf{MA}_\ell(u)) = d_0$ by Lemma 13, item 2; so that $S_i^-(u) < 1000\tilde{\gamma}$, i.e. $\tilde{\gamma} > .001 \cdot S_i^-(u)$.

Now, both the gates in G, and the subformulas rooted at these gates, don't depend on input i and so take the same values on u as on $\tilde{u}$. Thus, on u with $[\mathbf{i} = i]$ they fulfill the assumptions of Selection-Condition (c) in $A_{st}^X(Y)$ under our conditioning. As in earlier items, this implies $[\mathbf{k}_1 = \mathbf{i}]$ and $R \ge Y_i - X_i$. $\square$

8.1 Bad type-2 contexts

For a fixed $i \in T^* \cap I, i \perp j$, we now define a criterion for an i-context (x', y') to be considered "bad", thus defining a subset $\mathfrak{B}^2_{I,j,i}$ of such contexts. Essentially, the bad contexts we identify are those for which none of Lemma 18's items directly give a useful lower bound on R.

Our approach here should be compared with that used in Section 7. In each case we are using contexts (partial outcomes) as a useful "backdrop" to highlight a critical coordinate and identify cases in which the conditional expectation of V is nontrivially small. The backdrops' details differ, however, as we identify different sources of savings. In Section 7 we directly defined a set of (type-1) bad *index set* associated with a given j-context. Here we first define a set of bad i-contexts—but will also use this definition to induce a definition of (type-2) bad index sets associated with full *Ens*-outcomes.

Definition 28 (Bad (type-2) contexts and sets)**.** *Let (I, j, i) be Γ-supported and satisfy $i \in T^* \cap I$. We define $\mathfrak{B}^2_{I,j,i}$, the set of **"bad" (type-2) i-contexts** relative to (I, j, i), as the set of i-contexts (x', y') for which <u>all</u> of the following are true:*

0. $i \in \mathrm{Van}(\tilde{u})$;

1. $d_0 > \alpha - .93\Delta$;

2. d_0 <u>does</u> appear as $d_0 = \mathrm{Dec}(p^{i,2t}; \tilde{u})$ for some $t \in [0, C_1 - 1]$;

3. $d_f - d_0 < .95\Delta$;

4. <u>either</u> $\tilde{\gamma} = \perp$, <u>or</u> $1000\tilde{\gamma} < .95\Delta$.

*Also, for each Ens-outcome (x, y), we define the **(type-2) "bad" set***

$$B^2_{I,j,x,y} \subseteq T^* \cap I$$

as the set of indices $i \in T^* \cap I$ for which the associated i-context $(\hat{x}^i, \hat{y}^i) = ((x)_{[N]-\{i\}}, (y)_{[N]-\{i\}})$ satisfies $(\hat{x}^i, \hat{y}^i) \in \mathfrak{B}^2_{I,j,i}$.

The main significance of Def. 28 for bounding exponential moments is as follows:

Lemma 19. *Let (I, j, i) be Γ-supported, with $i \in T^* \cap I$. Let (x', y') be an i-context, and let $\tilde{u}$ be as in Def. 27 relative to (I, j, i, x', y'). Define the random variable*

$$\hat{Q} := z_i - X_i - R .$$

For conditional expectations of form

$$\mathbb{E}\left[\exp(10^{-4}s_0\hat{Q})|I,j,i,x',y'\right] \;=\; \mathbb{E}\left[\exp(10^{-4}s_0\hat{Q})\middle|(\mathcal{I},\mathbf{j},\mathbf{i})=(I,j,i)\ ,\ (X)_{[N]-\{i\}}=x'\ ,\ (Y)_{[N]-\{i\}}=y'\right]$$

we have the implication

$$(x',y')\ \notin\ \mathfrak{B}^2_{I,j,i}\quad\Longrightarrow\quad \mathbb{E}\left[\exp(10^{-4}s_0\hat{Q})|I,j,i,x',y'\right]\ \le\ \exp\left[10^{-4}s_0(\alpha-.93\Delta)\right]\ .$$

Proof. Given $(x',y')\ \notin\ \mathfrak{B}^2_{I,j,i}$, we ask which is the lowest-numbered item in Def. 28 to be violated by (x',y'). The options are analyzed as follows.

(0.) First, suppose $i\notin\mathrm{Van}(\tilde{u})$. From item 0 of Lemma 18, under the given conditioning we have $z_i-X_i-R\le 0$, from which the conclusion certainly follows (using that Ens is an (α,β,s_0)-ensemble and therefore also an $(\alpha,\beta,10^{-4}s_0)$-ensemble.

Henceforth, we assume $i\in\mathrm{Van}(\tilde{u})$.

(1.) If $d_0\le\alpha-.93\Delta$, then from item 1 of Lemma 18, under the given conditioning we have $z_i-X_i-R\le d_0$, and again the conclusion follows.

(2.) From item 3 of Lemma 18, under the given conditioning we have $z_i-X_i-R\le z_i-Y_i$, and the latter quantity, which is unaffected by the conditioning, is $(10^{-4}s_0,\beta)$-small.

(3.) Under no conditioning, define random variables

$$U:=z_i-Y_i\ ,\qquad W\ :=Y_i-X_i\ ,$$

Considering s_0-exponential moments now (rather than $10^{-4}s_0$-exponential ones), we have

$$\mathbb{E}[\exp(s_0U)]\le e^{s_0\beta}\ ,\qquad \mathbb{E}[\exp(s_0(U+W))]\le e^{s_0\alpha}$$

by the ensemble assumptions on X_i,Y_i.

Next, with reference to the fixed pair $\tilde{u},i$, we define values

$$S\ :=\ d_0\ ,\quad r\ :=\ d_f-d_0\ ,$$

where $f:=\lceil .49\ell\rceil$. Also define random variables

$$R_1\ :=\ \mathbf{1}_{[z_i-X_i<d_f]}\cdot(Y_i-X_i)\ =\ \mathbf{1}_{[U+W<S+r]}\cdot W$$

and

$$R_2\ :=\ \mathbf{1}_{[z_i-X_i\ge d_0]}\cdot(z_i-X_i-d_0)\ =\ \mathbf{1}_{[U+W\ge S]}\cdot(U+W-S)\ .$$

Let

$$Q\ :=\ U+W-\mathrm{Max}(R_1,R_2)\ .$$

As $s_0\Delta=1000$, and by our assumption that (x',y') violates item 3 (so that $r\ge .95\Delta$), the assumptions A1-A2 of Lemma 6 are met. Thus,

$$\mathbb{E}[\exp(.01s_0Q)]\ \le\ \exp\left[.01s_0(\alpha-.93\Delta)\right]\ .$$

We now pass to a lower exponent: by Prop. 6, we then also have

$$\mathbb{E}[\exp(10^{-4}s_0 Q)] \;\le\; \exp\left[10^{-4}s_0(\alpha - .93\Delta)\right] \;. \tag{38}$$

Now, we need to apply our unconditioned analysis of Q to the analysis of $\hat{Q}$ under conditioning $[I,i,j,x',y']$. Note that (as $\tilde{u},i$ are fixed) Q is a function solely of the random variables X_i,Y_i. Also, the conditioning $[I,i,j,x',y']$ does not affect the distribution of X_i,Y_i (by the independence property of our ensemble (X,Y)), and recalling that the i-coordinate is omitted in the i-context (x',y')).

Now under this conditioning, we have $\hat{Q} = U + W - R$. By Lemma 18, items 1 and 2, under the same conditioning we have $R \ge \mathrm{Max}(R_1,R_2)$. Also, $s_0 > 0$, so we conclude that

$$\mathbb{E}[\exp(10^{-4}s_0\hat{Q})|I,j,i,x',y'] \;\le\; \mathbb{E}[\exp(10^{-4}s_0 Q)|I,j,i,x',y'] \;=\; \mathbb{E}[\exp(10^{-4}s_0 Q)] \;.$$

With Eq. (38), this proves item 3.

(4.) The proof is completely analogous to that of item 3, with U,W defined as before, except that this time we instead define

$$S \;:=\; d_0\;, \qquad r \;:=\; 1000\tilde{\gamma}\;,$$

and we let

$$R_1 \;:=\; \mathbf{1}_{[z_i - X_i < d_0 + 1000\tilde{\gamma}]}\cdot(Y_i - X_i) \;=\; \mathbf{1}_{[U+W<S+r]}\cdot W$$

and

$$R_2 \;:=\; \mathbf{1}_{[z_i - X_i \ge d_0]}\cdot(z_i - X_i - d_0) \;=\; \mathbf{1}_{[U+W\ge S]}\cdot(U+W-S)\;.$$

We use these terms as before, but with items 1 and 4 of Lemma 18 instead of items 1-2. $\square$

8.2 Exceptional outcomes and contexts

In Lemma 19 we have established useful bounds under conditioning upon **i**-critical partial outcomes (I,j,i) and i-contexts which are not "bad". Now we wish to show that such bad contexts are in a useful sense "rare". Our basic approach for doing so is to show that for a "large" set[10] of *full* *Ens*-outcomes (x,y), the following favorable property holds: *most* of the possible choices $i \in T^* \cap I$ induce an associated i-context $(x',y') = ((x)_{N-\{i\}},(y)_{N-\{i\}})$ that is not in the "bad" set $\mathfrak{B}^2_{I,j,i}$ from Def. 28.

Certain "exceptional" *Ens*-outcomes and contexts may not have the favorable property described above, and pose a challenge for our analysis; we define them next, and by showing them to be "small" under an exponential measure, we will bound their impact on our overall analysis.

Definition 29 (Exceptional outcomes and contexts)**.** *Fix any* Γ*-supported outcome* $(\mathcal{I},j) = (I,j)$ *for* Γ *that satisfies* $j \perp T^*$*.*

[10](large under an appropriate exponential measure, that is)

1. *Let us say that an Ens-outcome to (X, Y) given by (x, y) is (I, j)-**exceptional**, and write $(x, y) \in \mathfrak{X}_{I,j}$ (a subset of outcomes), if there are at least $1 + .02(\sqrt{N} - 2^{30} - 1)$ coordinates $e \in T^* \cap I$ for which*

$$z_e - y_e \ > \ \beta + .06\Delta \ .$$

2. *Also, for $i \in T^* \cap I$ with $i \perp j$ and an i-context (x', y'), let us say that (x', y') is (I, j, i)-**exceptional**, and write $(x', y') \in \mathfrak{X}_{I,j,i}$, if there are at least $.02(\sqrt{N} - 2^{30} - 1)$ coordinates $e \in (T^* \cap I) - \{i\}$ for which $z_e - y'_e > \beta + .06\Delta$.*

Note that if Ens-outcome (x, y) is (I, j)-exceptional, then its associated i-context is (I, j, i)-exceptional; the converse may fail.

Next, for fixed Γ-supported (I, j) with $j \perp T^*$, and for $i \in T^* \cap I$, we will upper-bound the contribution to the expected value of the quantity $G^Y_{(T^* \cap I)-\{i\}} \cdot G^X_{T^*-I}$, from i-contexts which are exceptional. Note that the above quantity is fully determined by the i-context associated with (X, Y).

Lemma 20. *For any fixed pair (I, j) possible under Γ with $j \perp T^*$ and $i \in T^* \cap I$, we have*

$$\begin{aligned} &\mathbb{E}_{(x',y') \sim Ens} \left[\mathbf{1}[(x', y') \in \mathfrak{X}_{I,j,i}] \cdot G^Y_{(T^* \cap I)-\{i\}} \cdot G^X_{T^*-I} \right] \ \leq \\ &\qquad 10^{-10} \cdot \ \exp[10^{-4} s_0 (\beta(\sqrt{N} - 2^{30} - 1) + 2^{30} \cdot \alpha)] \ . \end{aligned} \tag{39}$$

Proof. The quantity $G^X_{T^*-I}$ is (using the independence property of Ens) independent of $G^Y_{(T^* \cap I)-\{i\}}$ and of the membership condition $\mathbf{1}[(x', y') \in \mathfrak{X}_{I,j,i}]$, which is determined by variables with coordinates in $(T^* \cap I) - \{i\}$. We have (again by independence)

$$\mathbb{E}[G^X_{T^*-I}] \ = \ \prod_{e \in T^*-I} \mathbb{E}[\exp(10^{-4} s_0 (z_e - x'_e))] \ \leq \ (e^{10^{-4} s_0 \alpha})^{|T^*-I|} \ = \ e^{10^{-4} s_0 (2^{30} \alpha)} \ . \tag{40}$$

Next, we apply Lemma 5 to the family $\{ \ U_e = (z_e - y'_e) \ \}_{(T^* \cap I)-\{i\}}$, a set of size $m := \sqrt{N} - 2^{30} - 1$. Note that the event $[H \geq .02m]$ in this case corresponds precisely to the outcome $[(x', y') \in \mathfrak{X}_{I,j,i}]$. By Lemma 5, then, we have

$$\mathbb{E} \left[\mathbf{1}[(x', y') \in \mathfrak{X}_{I,j,i}] \cdot G^Y_{(T^* \cap I)-\{i\}} \right] \ \leq \ \exp[10^{-4} s_0 (\beta - 5.88\Delta)(\sqrt{N} - 2^{30} - 1)] \ ,$$

which combined with Eq. (40) and the independence property gives

$$\begin{aligned} &\mathbb{E}_{(x',y') \sim Ens} \left[\mathbf{1}[(x', y') \in \mathfrak{X}_{I,j,i}] \cdot G^Y_{(T^* \cap I)-\{i\}} \cdot G^X_{T^*-I} \right] \ \leq \\ &\exp[-10^{-4} s_0 (5.88\Delta)(\sqrt{N} - 2^{30} - 1)] \ \cdot \ \exp[10^{-4} s_0 (\beta(\sqrt{N} - 2^{30} - 1) + 2^{30} \cdot \alpha)] \ . \end{aligned} \tag{41}$$

As $s_0 \Delta = 1000$ and $\sqrt{N} > 10^{10}$, the first factor appearing on the right-hand side is tiny: $\exp[-10^{-4} s_0 (5.88\Delta)(\sqrt{N} - 2^{30} - 1)] < 10^{-10}$. This yields Eq. (39). □

8.3 Size bounds for (type-2) bad sets

The membership of an index $i \in T^*$ in the bad set $B^2_{I,j,x,y}$ is determined by properties of the input $\tilde{u}$ (which depends on i and z_i). Since such inputs are distinct for each i, it is tricky to reason directly about the size of $B^2_{I,j,x,y}$. Thus we introduce a more convenient "surrogate" set below called $\hat{B}_{I,j,x,y}$ for the case when (x, y) are non-exceptional (the only case where we seek a bound on $|B^2_{I,j,x,y}|$), and relate it to our bad set in Lemma 21.

Definition 30. *Let (I, j) be Γ-supported, with $j \perp T^*$.*

- *Fix any non-(I, j)-exceptional Ens-outcome, $(x, y) \notin \mathfrak{X}_{I,j}$. We define*[11]

$$\hat{u} \ \in \ [0,1]^N$$

as the input which agrees with x on $\{j\} \cup ([N] - I)$, and with y on $I - \{j\}$.

- *Define*

$$\hat{B}_{I,j,x,y} \ \subseteq \ T^* \cap I$$

as the set of $i \in T^* \cap I$ for which all of the following conditions hold (these are related to the conditions of Def. 28):

0. $i \in \mathrm{Van}(\hat{u})$;

For the next items, we assume $i \in \mathrm{Van}(\hat{u})$ *and for such i, we let*

$$\hat{d}_{i,0} \ \le \ \hat{d}_{i,1} \ \le \ \ldots \ \le \ \hat{d}_{i,\ell-1}$$

denote the values $\{ \ \mathrm{Dec}(p^{i,2t}; \hat{u}) \ \}_{t\in[0,\ell-1]}$ listed in ascending order of value.

1. *The smallest value $\hat{d}_{i,0} \ge .01\Delta$;*

2. *The value $\hat{d}_{i,0}$ appears as $\hat{d}_{i,0} = \mathrm{Dec}(p^{i,2t}; \hat{u})$ for some $t \in [0, C_1 - 1]$;*

3. *For $f := \lceil .49\ell \rceil$, we have*

$$\hat{d}_{i,f} - \hat{d}_{i,0} \ < \ .95\Delta \ .$$

4. *There are fewer than $.01\ell$ Max gates g pendant to P_i whose decisiveness value satisfies*

$$\mathrm{Dec}(g; \hat{u}) \ \in \ [\hat{d}_{i,0}, 1000\hat{d}_{i,0}] \ .$$

Lemma 21. *For (I, j, x, y) as in Def. 30, we have*

$$B^2_{I,j,x,y} \ \subseteq \ \hat{B}_{I,j,x,y} \ \cup \ \{i \in T^* \cap I : z_i - y_i > \beta + .06\Delta\} \ .$$

[11](while the definition is closely similar to the input $\hat{u}$ considered in proving Lemma 5 in Section 7, the surrounding details are different and the notation's current scope is restricted to Section 8)

Proof. In the proof, we fix any $i \in T^* \cap I$ for which

$$z_i - y_i \ \leq \ \beta + .06\Delta \ ,$$

and assume that $i \notin \hat{B}_{I,j,x,y}$. We'll show the associated i-context $(x', y') := ((x)_{[N]-\{i\}}, (y)_{[N]-\{i\}})$ satisfies $(x', y') \notin \mathfrak{B}^2_{I,j,i}$, and thus (by definition) $i \notin B^2_{I,j,x,y}$, which proves the Lemma. Let $\tilde{u}$ be as determined by $(x', y'), i$ as in Def. 27. In what follows, we will repeatedly use the coordinate-decrement relation, in which the i^{th} coordinate is reduced by an amount $z_i - y_i$:

$$\hat{u} \ = \ \tilde{u}[i \leftarrow y_i] \ .$$

Now, we do case analysis, based on the lowest-numbered item violated by i in Def. 30:

(0.) First, suppose $i \notin \mathrm{Van}(\hat{u})$. Then, we claim that on input $\tilde{u}$, either $i \notin \mathrm{Van}(\hat{u})$, or, at least one Max gate $p^{i,2t}$ along P_i has $\mathrm{Dec}(p^{i,2t}; \tilde{u}) \leq \beta + .06\Delta = \alpha - .94\Delta$. In either case (x', y') violates one of items 0-1 of Def. 28.

To show the claim, suppose not. Then after the decrement of coordinate i from z_i (on $\tilde{u}$) to y_i (on $\hat{u}$), for which $z_i - y_i \leq \beta + .06\Delta$ by initial assumption, each such $p^{i,2t}$ would remain dominant for P_i on $\hat{u}$. This is seen by appealing to item 1 of Lemma 12. We then have $i \in \mathrm{Van}(\hat{u})$, which is a contradiction.

In the remaining items, we assume $i \in \mathrm{Van}(\hat{u})$, so that $\hat{d}_{i,0} \leq \ldots \leq \hat{d}_{i,\ell-1}$ are defined.

(1.) Next, assume $\hat{d}_{i,0} < .01\Delta$. If $p^{i,2t}$ is a Max gate on P_i with $\mathrm{Dec}(p^{i,2t}; \hat{u}) = \hat{d}_{i,0}$, we find (again by Lemma 12, item 1) that $\mathrm{Dec}(p^{i,2t}; \tilde{u}) = \hat{d}_{i,0} + (z_i - y_i) < .01\Delta + \beta + .06\Delta = \alpha - .93\Delta$, so that the value d_0 associated with $\tilde{u}$ is also less than $\alpha - .93\Delta$, and item 1 of Def. 28 is violated by (x', y').

(2.) Next, assume the smallest decisiveness value along P_i on input $\hat{u}$, namely $\hat{d}_{i,0}$, does *not* appear as $\hat{d}_{i,0} = \mathrm{Dec}(p^{i,2t}; \hat{u})$ for any $t \in [0, C_1 - 1]$. By Lemma 12, item 1, the values $\mathrm{Dec}(p^{i,2t}; \tilde{u})$ are equal to $\mathrm{Dec}(p^{i,2t}; \hat{u}) + (z_i - y_i)$ for each $t \in [0, \ell - 1]$, and considered in the ascending order, we also have

$$d_h \ = \ \hat{d}_{i,h} + (z_i - y_i) \ . \tag{42}$$

In particular the smallest such value, d_0, cannot appear as $\mathrm{Dec}(p^{i,2t}; \hat{u})$ for any $t \in [0, C_1 - 1]$. Thus item 2 of Def. 28 is violated by (x', y').

(3.) Next, assume $\hat{d}_{i,f} - \hat{d}_{i,0} \geq .95\Delta$. Then in light of Eq. (42), which holds here for the same reason as in case 2, we also have $d_f - d_0 \geq .95\Delta$, and item 3 of Def. 28 is violated by (x', y').

(4.) Finally, assume i obeys item 1 but fails item 4 of Def. 30: there exist at least $.01\ell$ Max gates g pendant to P_i whose decisiveness value satisfies $\mathrm{Dec}(g; \hat{u}) \in [\hat{d}_{i,0}, 1000\hat{d}_{i,0}]$. Let γ_0 be the minimum such value over all these g; by our assumption $\gamma_0 \geq .01\Delta$.

For each such g (which is pendant to, but not on P_i) we have $\mathrm{Dec}(g; \tilde{u}) = \mathrm{Dec}(g; \hat{u})$ since $\tilde{u}$ and $\hat{u}$ are equal on the subformula rooted at g. Thus, in Def. 27, we have $\tilde{\gamma} \neq \perp$ and $\tilde{\gamma} \geq \gamma_0 \geq .01\Delta$, so that $1000\tilde{\gamma} \geq 10\Delta$, and item 4 of Def. 28 is violated by (x', y'). $\square$

Lemma 22. *For (I, j, x, y) as in Def. 30, we have*

$$\left|\hat{B}_{I,j,x,y} \cup \{i \in T^* \cap I : z_i - y_i > \beta + .06\Delta\}\right| \;\leq\; .021 \cdot |T^* \cap I| \;=\; .021(\sqrt{N} - 2^{30}) \ .$$

By Lemma 21, the same upper bound holds for $|B^2_{I,j,x,y}|$.

Proof. Let $Z := \{i \in T^* \cap I : z_i - y_i > \beta + .06\Delta\}$. As $(x, y) \notin \mathfrak{X}_{I,j}$, we have (by definition)

$$|Z| \;<\; 1 + .02(\sqrt{N} - 2^{30} - 1) \ .$$

Our strategy to bound the size of $\hat{B}_{I,j,x,y} \subseteq T^* \cap I$ will involve reasoning about an index e selected uniformly from T^*. This distribution, while not exactly uniform on $T^* \cap I$, is quite close since $|T^* - I| = 2^{30} \lll \sqrt{N} = |T^*|$, and we will be able to apply our findings.

It will be convenient to define an auxiliary tree rather than working with the full MA_ℓ-formula tree $\mathbf{T}$, since our e as above is constrained to lie in T^*. Let $\mathcal{H}$ be the full binary tree of height $d := \ell$ whose leaf vertices are all input variables with coordinates $e \in T^*$, and whose non-leaf vertices are Max gates g lying on some path P_e with $e \in T^*$. For any such $g = \mathrm{Max}(h, h')$, as children of g in $\mathcal{H}$ we take g_1 and g_2 where, if $h = \mathrm{sel}_{T^*}(g)$ is the selected child of g in T^*, we have $h = \mathrm{Avg}(g_1, g_2)$.

To analyze $\mathcal{H}$, we use the terminology given for such trees in Sec. 4.1 before Lemma 10. First, we have a direct correspondence between paths P_e in with $e \in T^*$, and paths $\mathbf{P}_e$ in $\mathcal{H}$; a uniformly chosen $e \in T^*$ can equivalently sampled by taking a uniformly random path $\mathbf{P}_e$ in $\mathcal{H}$ from the root to a leaf input gate, of index e. Moreover, the Max gates encountered on this path in $\mathcal{H}$ will be precisely those lying on $\mathbf{P}_e$. Also, the Max gates hanging from $\mathbf{P}_e$ in $\mathcal{H}$, are precisely those Max gates pendant to P_e in $\mathbf{T}$. These observations will help us to get useful bounds by applying Lemma 10.

For a non-leaf node v in $\mathcal{H}$ (corresponding to some Max gate g in $\mathbf{T}$), say that v is **deep** if g is at distance more than $2C_1$ from the root gate in $\mathbf{T}$ (and so, g is not one of the first C_1 Max gates encountered on the path from the root to g itself). For each deep v, we define a quantity

$$D_{min}(v) \;:=\; \min\{\ \mathrm{Dec}(p^{e,2t}; \hat{u})\ \}_{t \in [0, C_1 - 1]} \ ,$$

where $e \in T^*$ is an index for which P_e passes through g; which such e is used does not affect our definition.

Next, we define some "marked" subsets of (non-leaf) vertices in $\mathcal{H}$. Let K be defined as the set of non-leaf nodes v in $\mathcal{H}$ (corresponding to some Max gate g in $\mathbf{T}$) for which at least one of the following conditions hold:

1. g is a non-root gate, and there exists some Max gate g' appearing *above* g on a path $P_{e'}$ with $e' \in T^*$ that passes through g, such that $P_{e'}$ is not dominant at g' with respect to $\hat{u}$; or,

2. $\mathrm{Dec}(g; \hat{u}) < .01\Delta$; or,

3. v is deep, and $\mathrm{Dec}(g; \hat{u}) < D_{min}(v)$.

Claim 6. *If $e \in (T^* \cap I)$, and the associated path $\mathbf{P}_e$ in $\mathcal{H}$ contains any non-leaf vertex $v \in K$, then $e \notin \hat{B}_{I,j,x,y}$.*

Proof of Claim 6. If any v on $\mathbf{P}_e$ fails item 1 in the definition of K, then P_e must pass through a vertex v' corresponding to a Max gate g', for which P_e is non-dominant with respect to $\hat{u}$. Thus e fails item 0 in the membership criteria for $\hat{B}_{I,j,x,y}$.

The same reasoning and conclusion holds if we simply assume $e \notin \mathrm{Van}(\hat{u})$; so we may henceforth assume that $e \in \mathrm{Van}(\hat{u})$, with associated values $\hat{d}_{e,0}, \ldots, \hat{d}_{e,\ell-1}$ as in Def. 30. If option 2 in the definition of K holds for some g on $\mathbf{P}_e$, then $\hat{d}_{e,0} \le \mathrm{Dec}(g; \hat{u}) < .01\Delta$, so that e must fail item 1 in the membership criteria for $\hat{B}_{I,j,x,y}$, and again $e \notin \hat{B}_{I,j,x,y}$.

Finally, suppose option 3 defining K holds for g on $\mathbf{P}_e$. Then, $\hat{d}_{i,0} \le \mathrm{Dec}(g; \hat{u}) < D_{min}(v)$, and by the definition of $D_{min}(v)$ we see that e fails item 2 for membership in $\hat{B}_{I,j,x,y}$. □

Define a second set M (possibly overlapping with K) of marked non-leaf nodes v in $\mathcal{H}$ (again, each with a corresponding Max gate g in $\mathbf{T}$), by

$$M \ := \ \{v \ : \ v \text{ is deep, and } \mathrm{Dec}(g; \hat{u}) \in [D_{min}(v), 101 \cdot D_{min}(v)] \ \} \ .$$

Claim 7. *Assume $e \in (T^* \cap I)$. The conclusion $e \notin \hat{B}_{I,j,x,y}$ then follows from either one of the following further conditions:*

1. *$\mathbf{P}_e$ passes through fewer than $.49\ell - C_1$ vertices in M;* <u>*or*</u>*,*
2. *More than $.01\ell$ vertices in M are hanging from $\mathbf{P}_e$.*

Proof of Claim 7. If $\mathbf{P}_e$ contains any $v \in K$ then we have the desired conclusion by Claim 6. So assume in the remainder that $\mathbf{P}_e$ does not intersect K.

In particular:

- The absence on $\mathbf{P}_e$ of any v for which item 1 defining K is met, implies that $e \in \mathrm{Van}(\hat{u})$, and $\hat{d}_{e,0}, \ldots, \hat{d}_{e,\ell-1}$ are defined;
- The fact that item 2 defining K is not met by any non-leaf vertex on $\mathbf{P}_e$, implies that $\hat{d}_{e,0} \ge .01\Delta$;
- The fact that item 3 defining K is not met on $\mathbf{P}_e$, implies that
$$D_{min}(v) \ = \ \hat{d}_{e,0} \tag{43}$$
for any deep (non-leaf) v on $\mathbf{P}_e$.

With these points in hand, we consider the two possible further assumptions of our Claim in turn:

(1.) Let J_1 be the set of Max gates g on P_e with

$$\mathrm{Dec}(g; \hat{u}) \ \in \ [\hat{d}_{e,0} \ , \ \hat{d}_{e,0} + \Delta] \ .$$

As $\hat{d}_{e,0} \ge .01\Delta$, the interval above is contained in $[\hat{d}_{e,0} \ , \ 101 \cdot \hat{d}_{e,0}]$. Any (deep, non-leaf) $v \in M$ appearing on $\mathbf{P}_e$, has $D_{min}(v) = \hat{d}_{e,0}$. If the corresponding Max gate g lies in J_1, we then find that $\mathrm{Dec}(g; \hat{u}) \in [D_{min}(v), 101 \cdot D_{min}(v)]$, and $v \in M$.

Thus $|J_1| \leq |M| + C_1$, after accounting for the first C_1 Max gates g along P_e, whose corresponding vertices on $\mathbf{P}_e$ might lie in $J_1 - M$. Our assumption in item 1 then implies that

$$|J_1| \ < \ .49\ell \ .$$

For $f = \lceil .49\ell \rceil$ as in Def. 30, we conclude that

$$\hat{d}_{e,f} \ > \ \hat{d}_{e,0} + \Delta \ ,$$

so that e violates item 3 in Def. 30, and $e \notin \hat{B}_{I,j,x,y}$ as claimed.

(2.) Under the alternative assumption in item 2 of Claim 7, let $M' \subseteq M$ be the more than $.01\ell$ vertices in M hanging from $\mathbf{P}_e$. Each $v' \in M'$ is deep and, as it hangs from $\mathbf{P}_e$ (and using Eq. (43)), satisfies $D_{min}(v') = \hat{d}_{e,0}$. Letting G be the set of Max gates corresponding to M' in $\mathbf{T}$, we observe that each $g \in G$ is pendant to P_e and satisfies

$$\mathrm{Dec}(g; \hat{u}) \ \in \ [\hat{d}_{e,0}, 101\hat{d}_{e,0}] \ ,$$

which implies that e violates item 4 of Def. 30 for membership in $\hat{B}_{I,j,x,y}$. □

Using Claim 7, we finish the proof of Lemma 22. First, recall that a uniformly chosen leaf of $\mathcal{H}$ corresponds to a uniformly selected e from T^*. Lemma 10 (with $d := \ell$) tells us that the probability $\mathbf{P}_e$ contains at least $.45\ell$ marked vertices, while having at most $.1\ell$ marked vertices hanging from $\mathbf{P}_v$, is $\leq (.998)^\ell$, which for our large $\ell = 10^{10}$ is $\leq .0001$.

Thus, the *number* of such e is at most $.0001\sqrt{N}$, so by Claim 7, $|\hat{B}_{I,j,x,y}| \leq .0001\sqrt{N}$. Then (using $N \geq 10^{21}$),

$$|\hat{B}_{I,j,x,y} \cup Z| \ \leq \ .0001\sqrt{N} + .02(\sqrt{N} - 2^{30} - 1) \ \leq \ .021(\sqrt{N} - 2^{30}) \ .$$

□

8.4 Exponential-moment bounds

Lemma 23. *For each Γ-supported (I, j, i) with $i \in T^* \cap I$, consider the i-context*

$$(\hat{x}^i, \hat{y}^i) \ := \ ((x)_{[N]-\{i\}}, (y)_{[N]-\{i\}})$$

associated with Ens-outcome (x, y), and define

$$\eta_i^{I,j} \ := \ \mathbb{E}_{(x,y)\sim Ens} \left[\ \mathbf{1}[(\hat{x}^i, \hat{y}^i) \in \mathfrak{B}^2_{I,j,i} - \mathfrak{X}_{I,j,i} \ ] \cdot G^Y_{(T^*\cap I)-\{i\}} \cdot G^X_{T^*-I} \ \right] \ . \tag{44}$$

Then, averaging over $i \in T^* \cap I$, we have the bound

$$\frac{1}{\sqrt{N} - 2^{30}} \sum_{i \in T^*\cap I} \eta_i^{I,j} \cdot \mathbb{E}[\exp(10^{-4} s_0(z_i - Y_i))] \ \leq$$

$$.021 \left(\prod_{e \in T^*\cap I} \mathbb{E}[\exp(10^{-4} s_0(z_e - Y_e))] \right) \left(\prod_{e \in T^*-I} \mathbb{E}[\exp(10^{-4} s_0(z_e - X_e))] \right) \ .$$

Proof. For our fixed I, j, we sample the outcome $(X, Y) = (x, y) \sim Ens$, and simultaneously for each $i \in T^* \cap I$ we consider the associated i-context $(\hat{x}^i, \hat{y}^i)$ as above.

The quantity $G^Y_{(T^*\cap I)} \cdot G^X_{T^*-I}$ is equal to $G^Y_{(T^*\cap I)-\{i\}} \cdot G^X_{T^*-I} \cdot \exp(10^{-4}s_0(z_i - y_i))$ for each $i \in T^* \cap I$. Both $G^Y_{(T^*\cap I)-\{i\}}$ and $(\hat{x}^i, \hat{y}^i)$ are independent of y_i, so from Eq. (44) defining $\eta_i^{I,j}$ we obtain

$$\mathbb{E}_{(x,y)\sim Ens}\left[\ \mathbf{1}[(\hat{x}^i, \hat{y}^i) \in \mathfrak{B}^2_{I,j,i} - \mathfrak{X}_{I,j,i}\ ] \cdot G^Y_{(T^*\cap I)} \cdot G^X_{T^*-I}\ \right] \ = \ \eta_i^{I,j} \cdot \mathbb{E}[\exp(10^{-4}s_0(z_i - Y_i))]\ . \tag{45}$$

By our initial observations after Def. 29, we have

$$(\hat{x}^i, \hat{y}^i) \ \notin \ \mathfrak{X}_{I,j,i} \quad \Longrightarrow \quad (x, y) \ \notin \ \mathfrak{X}_{I,j}\ .$$

Then, using Def. 28 defining type-2 bad contexts and sets, it holds too that

$$(\hat{x}^i, \hat{y}^i) \ \in \ \mathfrak{B}^2_{I,j,i} - \mathfrak{X}_{I,j,i} \quad \Longrightarrow \quad \left[(x, y) \ \notin \ \mathfrak{X}_{I,j} \ \wedge \ i \ \in \ B^2_{I,j,x,y}\right]\ .$$

Thus, from Eq. (45),

$$\eta_i^{I,j} \cdot \mathbb{E}[\exp(10^{-4}s_0(z_i - Y_i))] \ \leq \ \mathbb{E}\left[\ \mathbf{1}[(x, y) \ \notin \ \mathfrak{X}_{I,j} \ \wedge \ i \ \in \ B^2_{I,j,x,y}] \cdot G^Y_{(T^*\cap I)} \cdot G^X_{T^*-I}\ \right]\ .$$

Averaging over $i \in T * \cap I$,

$$\begin{aligned}
&\frac{1}{\sqrt{N} - 2^{30}} \sum_{i\in T^*\cap I} \eta_i^{I,j} \cdot \mathbb{E}[\exp(10^{-4}s_0(z_i - Y_i))] \ \leq \\
&\qquad \frac{1}{\sqrt{N} - 2^{30}} \sum_{i\in T^*\cap I} \mathbb{E}\left[\ \mathbf{1}[(x, y) \ \notin \ \mathfrak{X}_{I,j} \ \wedge \ i \ \in \ B^2_{I,j,x,y}] \cdot G^Y_{(T^*\cap I)} \cdot G^X_{T^*-I}\ \right] \\
&\qquad = \frac{1}{\sqrt{N} - 2^{30}}\ \mathbb{E}\left[G^Y_{(T^*\cap I)} \cdot G^X_{T^*-I} \cdot \sum_{i\in T^*\cap I} \mathbf{1}[(x, y) \ \notin \ \mathfrak{X}_{I,j} \ \wedge \ i \ \in \ B^2_{I,j,x,y}]\right] \\
&\qquad \leq \frac{1}{\sqrt{N} - 2^{30}}\ \mathbb{E}\left[G^Y_{(T^*\cap I)} \cdot G^X_{T^*-I} \cdot |B^2_{I,j,x,y}|\right] \\
&\qquad \leq .021 \cdot \ \mathbb{E}[G^Y_{(T^*\cap I)} \cdot G^X_{T^*-I}] \\
&\qquad\quad \text{(by Lemma 22)} \\
&\qquad = .021 \prod_{e\in T^*\cap I} \mathbb{E}[\exp(10^{-4}s_0(z_e - Y_e))] \ \cdot \prod_{e\in T^*-I} \mathbb{E}[\exp(10^{-4}s_0(z_e - X_e))]\ ,
\end{aligned}$$

as claimed (using independence of coordinates in the last step). □

Proof of Claim 3. After conditioning on $[\mathbf{i} \in T^*]$, the pair $(\mathcal{I}, \mathbf{j})$ is uniform over its possible outcomes for which $\mathbf{j} \perp T^*$, and letting Γ' be the induced distribution on such pairs, we have

$$\mathbb{E}[V|\mathbf{i} \in T^*] \ = \ \mathbb{E}_{(I,j)\sim\Gamma'}[\ \mathbb{E}[V|(\mathcal{I}, \mathbf{j}) = (I, j) \wedge \mathbf{i} \in T^*]\ ]\ . \tag{46}$$

Fix any such (I, j) with $j \perp T^*$. Conditioned on $[(\mathcal{I}, \mathbf{i}) = (I, j) \wedge \mathbf{i} \in T^*]$, the value $\mathbf{i}$ is uniform over $T^* \cap I$, a set of size $\sqrt{N} - 2^{30}$. Thus,

$$\mathbb{E}[V|(\mathcal{I}, \mathbf{j}) = (I, j) \wedge \mathbf{i} \in T^*] \ = \ \frac{1}{\sqrt{N} - 2^{30}} \sum_{i\in T^*\cap I} \mathbb{E}[V|(\mathcal{I}, \mathbf{j}, \mathbf{i}) = (I, j, i)]\ . \tag{47}$$

We fix $i \in T^* \cap I$ and further expand $\mathbb{E}[V|(\mathcal{I},\mathbf{j},\mathbf{i}) = (I,j,i)]$, hereafter denoted $\mathbb{E}[V|I,j,i]$. Note that under this conditioning (using the fact that $i \in T^* \cap I, j \in I - T^*$), we have

$$V \; = \; G^Y_{(T^*\cap I)-\{i\}} \cdot G^X_{T^*-I} \cdot \exp(10^{-4} s_0 (z_i - x_i - R)) \; .$$

With an eye toward Lemmas 20 and 23, we let (x', y') be the i-context associated with outcome (X, Y), and we define

- $a_{i,1} \; := \; \mathbb{E}\left[\mathbf{1}[(x',y') \in \mathfrak{X}_{I,j,i}] \cdot G^Y_{(T^*\cap I)-\{i\}} \cdot G^X_{T^*-I}\right]$,
- $a_{i,2} \; := \; \mathbb{E}\left[\mathbf{1}[(x',y') \in \mathfrak{B}^2_{I,j,i} - \mathfrak{X}_{I,j,i}] \cdot G^Y_{(T^*\cap I)-\{i\}} \cdot G^X_{T^*-I}\right] \; = \; \eta_i^{I,j}$,
- $a_{i,3} \; := \; \mathbb{E}\left[\mathbf{1}[(x',y') \notin (\mathfrak{B}^2_{I,j,i} \cup \mathfrak{X}_{I,j,i})] \cdot G^Y_{(T^*\cap I)-\{i\}} \cdot G^X_{T^*-I}\right]$,

noting that the quantity satisfies

$$a_{i,1} + a_{i,2} + a_{i,3} \; = \; \mathbb{E}[G^Y_{(T^*\cap I)-\{i\}} \cdot G^X_{T^*-I}] \; \le \; \exp[10^{-4} s_0 ((\sqrt{N} - 2^{30} - 1)\beta + 2^{30}\alpha)] \; .$$

Now, under *any* conditioning on the i-context (x', y'), the ensemble property (which extends to exponent $10^{-4}s_0$, by Prop. 6) and the nonnegativity of R tell us that

$$\mathbb{E}[\exp(10^{-4}s_0(z_i - x_i - R))|I,j,i,x',y'] \; \le \; \mathbb{E}[\exp(10^{-4}s_0(z_i - X_i))] \; \le \; \exp(10^{-4}s_0\alpha) \; . \quad (48)$$

Further, upon conditioning on any $(x',y') \notin (\mathfrak{B}^2_{I,j,i} \cup \mathfrak{X}_{I,j,i})$, we have

$$\mathbb{E}[\exp(10^{-4}s_0(z_i - x_i - R))|I,j,i,x',y'] \; \le \; \exp[10^{-4}s_0(\alpha - .93\Delta)] \; ,$$

by Lemma 19.

By Lemma 20, we have

$$a_{i,1} \; \le \; 10^{-10} \exp[10^{-4} s_0 ((\sqrt{N} - 2^{30} - 1)\beta + 2^{30}\alpha)] \; .$$

Let us now suppose that

$$\begin{aligned} &\eta_i^{I,j} \cdot \mathbb{E}[\exp(10^{-4}s_0(z_i - X_i))] \; \le \\ &\qquad \sqrt{.021} \prod_{e \in T^*\cap I} \mathbb{E}[\exp(10^{-4}s_0(z_e - y'_e))] \; \cdot \prod_{e\in T^*-I} \mathbb{E}[\exp(10^{-4}s_0(z_e - x'_e))] \end{aligned} \qquad (49)$$

or equivalently $a_{i,2} = \eta_i^{I,j} \le \sqrt{.021}\cdot\mathbb{E}[G^Y_{(T^*\cap I)-\{i\}} \cdot G^X_{I-T^*}]$, which is at most $\sqrt{.021}\cdot\exp(10^{-4}s_0[(\sqrt{N} - 2^{30} - 1)\beta + 2^{30}\alpha])$. Then,

$$\mathbb{E}[\mathbf{1}[(x',y') \in \mathfrak{B}^2_{I,j,i} \cup \mathfrak{X}_{I,j,i}] \cdot G^Y_{(T^*\cap I)-\{i\}} \cdot G^X_{I-T^*}] \; \le \; (10^{-10} + \sqrt{.021}) \exp(10^{-4}s_0[(\sqrt{N} - 2^{-30} - 1)\beta + 2^{30}\alpha]) \; .$$

We can then apply Lemma 2, to the event $W = [(\mathcal{I},\mathbf{j},\mathbf{i}) = (I,j,i)]$ and random variables

$$\mathbf{t} \; := \; ((X)_{[N]-\{i\}}, (Y)_{[N]-\{i\}}) \; , \quad A \; := \; G^Y_{(T^*\cap I)-\{i\}} \cdot G^X_{T^*-I} \; , \quad B \; := \; \exp(10^{-4}s_0(z_i - x_i - R)) \; ,$$

noting that A, B are independent conditional upon $[W \wedge (x', y')]$, for any outcome to $\mathbf{t}$ (which determines the value of A). We take as the set S, for Lemma 2, the set of i-contexts $\mathfrak{B}^2_{I,j,i} \cup \mathfrak{X}_{I,j,i}$ (a subset of the set $\mathrm{supp}(\mathbf{t})$ of all i-contexts supported by Ens), and with

$$a \;:=\; \exp(10^{-4} s_0((\sqrt{N} - 2^{30} - 1)\beta + 2^{30}\alpha)) \;,$$

$$b \;:=\; e^{10^{-4} s_0 \alpha} \;, \quad b' \;:=\; e^{10^{-4} s_0(\alpha - .93\Delta)} \;,$$

and

$$p \;:=\; 10^{-10} + \sqrt{.021} \;<\; .145 \;,$$

to find that

$$\begin{aligned} \mathbb{E}[V|I,j,i] &\;=\; \mathbb{E}[AB|W] \\ &\;\leq\; a[pb + (1-p)b'] \;. \end{aligned}$$

Regarding b, b' as e^f, e^g respectively, we have $f - g = (10^{-4} s_0 \alpha) - 10^{-4} s_0(\alpha - .93\Delta) = 10^{-4}(.93) s_0 \Delta < .1$, since $s_0 \Delta = 1000$. Thus, we can apply Lemma 3 to the above, to find that (for i satisfying Eq. (49))

$$\begin{aligned} \mathbb{E}[V|I,j,i] &\;\leq\; a \cdot \exp[1.1p(10^{-4} s_0 \alpha) + (1 - 1.1p)(10^{-4} s_0(\alpha - .93\Delta))] \\ &\;=\; a \cdot \exp[10^{-4} s_0(\alpha - (1 - 1.1p)(.93\Delta))] \\ &\;\leq\; a \cdot \exp[10^{-4} s_0(\alpha - .78\Delta)] \;. \end{aligned} \tag{50}$$

Let $b'' := \exp[10^{-4} s_0(\alpha - .78\Delta)]$.

Now by Lemma 23 combined with a Markov bound, at least a .85 fraction of all $i \in T^* \cap I$ satisfy Eq. (49) and thus Eq. (50). And, all other $i \in T^* \cap I$ at least satisfy

$$\mathbb{E}[V|I,j,i] \;\leq\; \mathbb{E}[G^Y_{(T^* \cap I)} \cdot G^X_{(T^* - I) \cup \{i\}}] \;\leq\; \exp[s((\sqrt{N} - 2^{30})\beta + (2^{30}+1)\alpha)] \;=\; ab \;,$$

since $i \in T^*, j \notin T^*$. From these bounds and Eq. (47), then, we infer

$$\begin{aligned} \mathbb{E}[V|(\mathcal{I}, \mathbf{j}) = (I, j) \wedge \mathbf{i} \in T^*] &\;\leq\; .15ab + .85ab'' \\ &\;=\; a[.15b + .85b''] \;. \end{aligned}$$

Making another application of Lemma 3, we have

$$\begin{aligned} a[.15b + .85b''] &\;\leq\; a \cdot \exp\left[(1.1 \cdot .15)(10^{-4} s_0 \alpha) + (1 - 1.1 \cdot .15)(10^{-4} s_0(\alpha - .78\Delta))\right] \\ &\;=\; a \cdot \exp\left[10^{-4} s_0(\alpha - .835(.78\Delta))\right] \\ &\;\leq\; a \cdot \exp\left[10^{-4} s_0(\alpha - .65\Delta)\right] \\ &\;=\; \exp[10^{-4} s_0((\sqrt{N} - 2^{30} - 1 + .65)\beta + (2^{30} + 1 - .65)\alpha)] \;, \end{aligned}$$

Then from Eq. (46), we infer the same upper bound on $\mathbb{E}[V|\mathbf{i} \in T^*]$. This proves Claim 3. $\square$

9 Reduction of MA_k to LIS

Here we describe an approximation-preserving reduction mentioned in the Introduction. It is modeled on a reduction from the Tropical Tensor Similarity problem to the Longest Common Subsequence (LCS) problem in [AR18], and guided by [AR18] and a comment in [RSSS19] on a connection between LCS and LIS.

First we briefly sketch the reduction of [AR18] applied to MA_k. For finite strings u, v let $len_{LCS}(u, v)$ denote the length of the longest common (not necessarily contiguous) subsequence of u, v. The base transformation on single bits mapping $[x_i = 1]$ to the pair of strings $(u, v) = (i, i)$ and $[x_i = 0]$ to the pair $(\perp_A, \perp_B)$, where $\perp_A, \perp_B$ are distinct "null" symbols, outputs a pair satisfying $len_{LCS}(u, v) = \mathbf{1}_{[x_i=1]}$. For an inductive step, one uses the transformation

$$(u, v)\ ,\ (u', v') \quad \to \quad (u \circ u', v \circ v')$$

which satisfies

$$len_{LCS}(u \circ u', v \circ v') \ = \ len_{LCS}(u, v) + len_{LCS}(u', v')$$

provided that $len_{LCS}(u, v') = len_{LCS}(u', v) = 0$. This is combined with the transformation

$$(u, v)\ ,\ (u', v') \quad \to \quad (u \circ u', v' \circ v)$$

which under the same assumption as above satisfies

$$len_{LCS}(u \circ u', v' \circ v) \ = \ \max\{len_{LCS}(u, v)\ ,\ len_{LCS}(u', v')\}$$

Using these steps in alternation, one obtains from a string $x \in \{0, 1\}^{4^k}$ a pair $u, v \in \{[4^k] \cup \{\perp_A, \perp_B\}\}^{4^k}$ such that $len_{LCS}(u, v) = 2^k \cdot \mathsf{MA}_k(x)$.

In the Longest Increasing Subsequence (LIS) problem, one is given as input a list $L = (a_1, \ldots, a_N)$ of integers, and wishes to find a longest, strictly increasing subsequence (which need not be contiguous, and may be a single element). Let $len_{LIS}(L) \in [1, N]$ denote the length of a LIS for L. We mimic the above reduction for LCS, with additive shifts playing a role analogous to string reordering.

Let LIS$^{\perp}$ be the same problem, except that the a_i are allowed to take the value "$\perp$", and such an entry is not permitted in any increasing subsequence. Let $len_{LIS,\perp}(L) \in [0, N]$ be the length of a longest such increasing subsequence.

We first describe a simple, "nearly-perfect" reduction from LIS$^{\perp}$ to LIS. If $L \in (\mathbb{Z} \cup \ \perp)^N$ is given and a lower bound $\min_i a_i \geq c \in \mathbb{Z}$ is known, one can replace any occurrence in L of $a_i = \perp$ by a new entry $a'_i = c - i$. This yields a revised list $L' \in \mathbb{Z}^N$, and one sees that $val_{LIS}(L') = \max\{1, val_{LIS,\perp}(L)\}$. Thus the reduction exactly preserves the optimum value in the case where $L \neq \perp^N$.

Next, we reduce the MA_k problem for Boolean inputs to LIS$^{\perp}$. Following [AR18], the reduction is inductive. For $k = 0$ and Boolean inputs to $\mathsf{MA}_k(x_1) = x_1$, we map $x_1 = 0 \to a_1 = \perp$ and $x_1 = 1 \to a_1 = 1$. Clearly for the obtained 1-element sequence L, we have $len_{LIS,\perp}(L) = x_1$.

Now say $k \geq 1$ and inductively assume we have a mapping such that for all $y \in \{0, 1\}^{4^{k-1}}$, the output list $L = L(y) \in \mathbb{Z}^{4^{k-1}}$ satisfies

$$len_{LIS,\perp}(L(y)) \ = \ 2^{k-1} \cdot \mathsf{MA}_{k-1}(y)\ ,$$

with known upper and lower bounds $a_i \in [1, e_{k-1}]$ for all non-$\perp$ entries holding across all such y. (We may take $e_0 = 1$.)

Suppose input $x = (x^a, x^b, x^c, x^c)$ of length 4^k is given, so that

$$\mathsf{MA}_k(x) \ = \ \max\{\mathrm{Avg}(\mathsf{MA}_{k-1}(x^a), \mathsf{MA}_{k-1}(x^b)), \mathrm{Avg}(\mathsf{MA}_{k-1}(x^c), \mathsf{MA}_{k-1}(x^d))\} \ .$$

Let L^a, L^b, L^c, L^d, each in $\{\mathbb{Z} \cup \ \perp\}^{4^{k-1}}$ be the respective output reductions for x^a through x^d on size 4^{k-1}. Let

$$L'_a \ := \ L_a + 2e_{k-1} \ , \qquad L'_b \ := \ L_b + 3e_{k-1} \ , \qquad L'_d \ := \ L_d + e_{k-1}$$

where for $e \in \mathbb{Z}$ we use $L + e = (a_i)_{i \le t} + e$ to denote the sequence $(a_i + e)_{i \le t}$ obtained by entrywise addition, with $\perp + e = \perp$. Using $\circ$ to denote concatenation of sequences, let

$$L(x) \ := \ L'_a \circ L'_b \circ L_c \circ L'_d$$

be the output of the reduction for input x. We can take $e_k := 4e_{k-1}$, and all non-$\perp$ entries of $L(x)$ are in $[1, e_k]$.

Observe that if s_a, s_b are (possibly empty) increasing subsequences in L_a and in L_b respectively, then the union of their corresponding elements in $(L'_a \circ L'_b)$ is an increasing subsequence of $L(x)$. Similarly, a pair s_c, s_d of increasing subsequences in L_c, L_d induces one in $(L_c \circ L'_d)$, hence in $L(x)$. One also checks that no increasing subsequence can contain an element from both the first and second half of $L(x)$, and so unions as above are the only increasing subsequences. We conclude that

$$\begin{aligned} len_{LIS,\perp}(L(y)) \ &= \ \max\{len_{LIS,\perp}(L^a) + len_{LIS,\perp}(L^b) \ , \ len_{LIS,\perp}(L^c) + len_{LIS,\perp}(L^d)\} \\ &= \ 2 \cdot \max\{\mathrm{Avg}(2^{k-1}\mathsf{MA}_{k-1}(x^a), 2^{k-1}\mathsf{MA}_{k-1}(x^b)), \mathrm{Avg}(2^{k-1}\mathsf{MA}_{k-1}(x^c), 2^{k-1}\mathsf{MA}_{k-1}(x^d))\} \\ &= \ 2^k \cdot \mathsf{MA}_k(x) \ . \end{aligned}$$

One can then combine the above with the "nearly-perfect" reduction of $\mathrm{LIS}^\perp$ to LIS. The resulting reduction carries any $x \in \{0,1\}^{4^k}$ to an integer sequence L of length $N = 4^k$, and satisfies $len_{LIS}(L) = 2^k \mathsf{MA}_k(x)$ provided x is not all-zero. More strongly, there is a natural correspondence between increasing subsequences in L of length $d \ge 2$, and sets of d indices $i \in [4^k]$ residing in a common M-tree and satisfying $x_i = 1$, collectively certifying $\mathsf{MA}_k(x) \ge d/2^k$.

10 Discussion: SampleEvalSSAT on fully-alternating games

Here, as promised in the Introduction, we explain the strong limitations of the SampleEvalSSAT algorithm of [LMP01] for the fully alternating binary games against Nature we study. This algorithm, which is described for stochastic satisfiability (approximating the value of games where the payoff function $F = F(w^1, r^1, \ldots, w^k, r^k)$ and move-order are given by a Boolean formula with Exists and For-Random quantifiers), makes sense for general functions F and associated games against Nature as well. In "generic" form (prior to the specific choice of parameter T below, which affects the approximation quality), the algorithm proceeds by the following outline:

1. Uniformly select some number T of move-sequences $\overline{r}^{(i)}$ for the Nature player;

2. Output, as an estimate to the value of F, the *maximum* empirical performance of any strategy for the non-random player, against the sample $\{\overline{r}^{(i)}\}$. (The empirical performance of a strategy S is its average payoff against the move-sequences $\{\overline{r}^{(i)}\}$.)

For the implementation of step 2, see [LMP01]; the algorithm inspects every entry of F in which $r = \overline{r}^{(i)}$ for some i, so the query complexity of the algorithm is $T \cdot 2^m$, where m is the total bit-length of the non-random player's moves, i.e. $|w^1| + \ldots + |w^k|$.

Consider now the fully-alternating case, where each move w^j and r^j are 1-bit and $k = m = n/2$. Then for the query complexity to be asymptotically better than brute force, we need $T = T(n) = o(2^{n/2})$. Fix any such $T(n)$. If this many samples $\overline{r}^{(i)}$ are drawn uniformly, then there is a pair of functions $h(n) = \omega(n)$ and $\delta(n) = o(1)$, for which the following **"lonely" property** holds with probability $1 - \delta(n)$:

- Only a $\delta(n)$ fraction of strings $\overline{r}^{(i)}$ have a "close partner" $\overline{r}^{(i')}$, for which $\overline{r}^{(i)}$ and $\overline{r}^{(i')}$ are distinct but agree on all but their final $h(n)$ bits.

This is because the *expected* number of close partners to each i is $o(1)$, if our parameters are chosen appropriate to the rate of decay of $T(n)/n$.

We now define a specific input function $F = F(w_1, r_1, \ldots, w_{n/2}, r_{n/2})$, which is a minor extension of an example studied for similar reasons in [LMP01, App. B]. This function is based on a "guessing game" for the non-random player: it outputs 1 if $w_i = r_i$ for each $i > n/2 - h(n)$, and outputs 0 otherwise.

The true value of F is $2^{-h(n)} = o(1)$, since no strategy can predict the final $h(n)$ unseen bits (which in the true game, are uniform) with probability better than this. On the other hand, if $\{\overline{r}^{(i)}\}$ obey the lonely property above, we can use it to define a strategy S which, having viewed the first $n - h(n)$ bits of r, sets the bits $w_{n-h(n)+1}, \ldots, w_n$ according to the unique extension of r to some $\overline{r}^{(i)}$ (when such an extension exists and is unique).

It follows from the lonely property that the empirical performance of S on $\{\overline{r}^{(i)}\}$ is at least $1 - \delta(n) = 1 - o(1)$. And, such an S exists with probability $1 - o(1)$ over the initial sample. Thus, SampleEvalSSAT reliably and drastically overestimates the value of this F, when run with $T(n) = o(2^{n/2})$.

References

[AB17] Amir Abboud and Arturs Backurs. Towards hardness of approximation for polynomial time problems. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 11:1–11:26, 2017.

[Alb03] Susanne Albers. Online algorithms: a survey. *Mathematical Programming*, 97(1):3–26, Jul 2003.

[AR18] Amir Abboud and Aviad Rubinstein. Fast and deterministic constant factor approximation algorithms for LCS imply new circuit lower bounds. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 35:1–35:14, 2018.

[AW09] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *ACM Trans. Comput. Theory*, 1(1):2:1–2:54, 2009.

[Bab85] László Babai. Trading group theory for randomness. In Robert Sedgewick, editor, *17th ACM STOC*, pages 421–429. ACM, 1985.

[BDG00] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Stochastic dynamic programming with factored representations. *Artif. Intell.*, 121(1-2):49–107, 2000.

[BDH99] Craig Boutilier, Thomas L. Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *J. Artif. Intell. Res.*, 11:1–94, 1999.

[CGL$^+$19] Lijie Chen, Shafi Goldwasser, Kaifeng Lyu, Guy N. Rothblum, and Aviad Rubinstein. Fine-grained complexity meets IP = PSPACE. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1–20. SIAM, 2019.

[DP09] Devdatt Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[GKPV03] Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored MDPs. *J. Artif. Intell. Res.*, 19:399–468, 2003.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[GS86] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *18th STOC*, pages 59–68, 1986.

[HRTP11] Brahim Hnich, Roberto Rossi, S. Armagan Tarim, and Steven Prestwich. *A Survey on CP-AI-OR Hybrids for Decision Making Under Uncertainty*, pages 227–270. Springer New York, 2011.

[HSSS19] MohammadTaghi Hajiaghayi, Masoud Seddighin, Saeed Seddighin, and Xiaorui Sun. Approximating LCS in linear time: Beating the $\sqrt{\text{n}}$ barrier. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1181–1200, 2019.

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[Les18] Maya Leshkowitz. Round complexity versus randomness complexity in interactive proofs. In *RANDOM 2018*, pages 49:1–49:16, 2018.

[LFKN92] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[Lit99] Michael L. Littman. Initial experiments in stochastic satisfiability. In *AAAI/IAAI*, pages 667–672, 1999.

[LMP01] Michael L. Littman, Stephen M. Majercik, and Toniann Pitassi. Stochastic boolean satisfiability. *Journal of Automated Reasoning*, 27(3):251–296, Oct 2001.

[LWJ18] Nian-Ze Lee, Yen-Shi Wang, and Jie-Hong R. Jiang. Solving exist-random quantified stochastic boolean satisfiability via clause selection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1339–1345, 2018.

[ML99] Stephen M. Majercik and Michael L. Littman. Contingent planning under uncertainty via stochastic satisfiability. In *AAAI/IAAI*, pages 549–556, 1999.

[MNS+16] Frédéric Magniez, Ashwin Nayak, Miklos Santha, Jonah Sherman, Gábor Tardos, and David Xiao. Improved bounds for the randomized decision tree complexity of recursive majority. *Random Struct. Algorithms*, 48(3):612–638, 2016.

[MS85] Burkhard Monien and Ewald Speckenmeyer. Solving satisfiability in less than 2^{n} steps. *Discrete Applied Mathematics*, 10(3):287–295, 1985.

[Pap85] Christos H. Papadimitriou. Games against nature. *J. Comput. Syst. Sci.*, 31(2):288–301, 1985.

[PPZ99] Ramamohan Paturi, Pavel Pudlák, and Francis Zane. Satisfiability coding lemma. *Chicago J. Theor. Comput. Sci.*, 1999.

[PT87] Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[Put94] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

[RSSS19] A. Rubinstein, S. Seddighin, Z. Song, and X. Sun. Approximation algorithms for LCS and LIS with truly improved running times. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1121–1145, 2019.

[San] Rahul Santhanam. Personal communication.

[San95] Miklos Santha. On the monte carlo boolean decision tree complexity of read-once formulae. *Random Struct. Algorithms*, 6(1):75–88, 1995.

[Sch99] Uwe Schöning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 410–414, 1999.

[Sha92] Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

[Sni85] Marc Snir. Lower bounds on probabilistic linear decision trees. *Theoretical Computer Science*, 38:69 – 82, 1985.

[SS17] Michael E. Saks and C. Seshadhri. Estimating the longest increasing sequence in polylogarithmic time. *SIAM J. Comput.*, 46(2):774–823, 2017.

[SW86] Michael E. Saks and Avi Wigderson. Probabilistic boolean decision trees and the complexity of evaluating game trees. In *27th IEEE FOCS*, pages 29–38, 1986.

[SW15] Rahul Santhanam and Ryan Williams. Beating exhaustive search for quantified boolean formulas and connections to circuit complexity. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 231–241, 2015.

[TF10] Tino Teige and Martin Fränzle. Resolution for stochastic boolean satisfiability. In *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 625–639, 2010.

[TF11] Tino Teige and Martin Fränzle. Generalized Craig interpolation for stochastic boolean satisfiability problems. In *Tools and Algorithms for the Construction and Analysis of Systems - 17th International Conference, TACAS*, pages 158–172, 2011.

[Wal02] Toby Walsh. Stochastic constraint programming. In *15th European Conference on Artificial Intelligence (ECAI)*, pages 111–115, 2002.

[Wil16] Ryan Williams. Strong ETH breaks with Merlin and Arthur: Short non-interactive proofs of batch evaluation. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 2:1–2:17, 2016.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il