Meeting in a Convex World: Convex Consensus with Asynchronous Fallback

Andrei Constantinescu, Diana Ghinea, Roger Wattenhofer, and Floris Westermann

ETH Zürich
{aconstantine, ghinead, wattenhofer, wfloris}@ethz.ch

Abstract

Convex Consensus (CC) allows a set of parties to agree on a value v inside the convex hull of their inputs with respect to a predefined convexity notion, even in the presence of byzantine parties. In this work, we focus on achieving CC in the best-of-both-worlds paradigm, i.e., simultaneously tolerating at most t_s corruptions if communication is synchronous, and at most $t_a \leq t_s$ corruptions if it is asynchronous. Our protocol is randomized, which is a requirement under asynchrony, and we prove that it achieves optimal resilience. In the process, we introduce communication primitives tailored to the best-of-both-worlds model, which we believe to be of independent interest. These are a deterministic primitive, which allows honest parties to obtain intersecting views, and a randomized primitive, leading to identical views (which is impossible to achieve deterministically).

Afterwards, we consider achieving consensus using deterministic protocols, for which the agreement condition must be appropriately relaxed depending on the convexity space. For the relevant case of graph convexity spaces, we find that a previous asynchronous approximate agreement protocol for chordal graphs is incorrect, and hereby give a new protocol for the problem designed for the best-of-both-worlds model and achieving tight point-wise resilience bounds. Finally, we show that asynchronous graph approximate agreement remains unsolvable by deterministic protocols even when corruptions are restricted to at most two crashing nodes and the distance agreement threshold is linear in the size of the graph.

1 Introduction

Arranging a meeting place for a group n of people in a city is a common problem, as determining a location that is convenient and accessible for everyone can often be challenging. Oftentimes, such a location can be determined by its geographic coordinates. In other cases, it may be more convenient to represent the map of the city as a graph, with streets modeled as edges, and intersections as vertices. Participants are initially in different locations, i.e., in different vertices, and they want to agree on a vertex for their meeting point via pair-wise communication channels. Finding such a meeting point, while also taking into account that misunderstandings may occur, or that some of the participants may choose to behave dishonestly and not follow the protocol, describes the Convex Consensus problem (CC).

The CC problem serves as a unifying framework for various agreement problems that deal with different input spaces. Such input spaces may be continuous, such as $\mathbb{R}^D$, or discrete, such as graphs and even lattices. Essentially, CC assumes a publicly available input space V (this could be, for instance, the set of geo-coordinates representing the locations) equipped with a convexity notion $\mathcal{C}$ (roughly meant to formalize which potential meeting points are convenient with respect to the participants' inputs). For example, in the case of $\mathbb{R}^D$, the standard "straight-line" convexity notion can be considered. In contrast, convexity notions for graphs may be defined in various ways: for example, *geodesic convexity*, defined over shortest paths between vertices, or *monophonic convexity*, defined over minimal/chordless paths. For a given convexity

notion, CC is concerned with enabling parties to agree on some value lying in the convex hull of their inputs. This should be achieved even if up to t of the parties involved are corrupted and may exhibit malicious behavior.

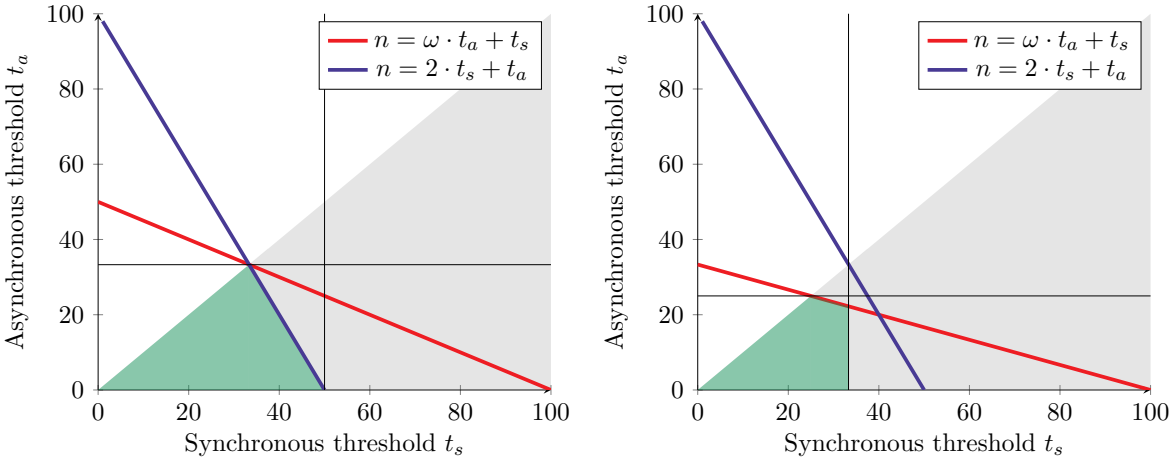
Agreement problems in general have been the subject of a tremendous line of work in Distributed Computing. CC has been introduced by Vaidya and Garg for $\mathbb{R}^D$ in [29], where the authors showed that achieving CC on real values in the so-called synchronous model, where parties have synchronized clocks and messages get delivered within a known amount of time Δ, can be achieved if and only if at most $t < n/(D+1)$ of the parties involved are corrupted. To the best of our knowledge, CC has not been studied in the asynchronous model, which makes no assumptions on the parties' clocks or on message delays, except for all messages getting delivered eventually. In part, this is a consequence of the seminal result of Fischer, Lynch and Paterson [18], implying that the standard definition of CC, or agreement in general, cannot be achieved by deterministic protocols in the asynchronous model. As a result, research has instead focused on Approximate Agreement (AA), introduced in [15], a relaxation of CC requiring that the parties agree up to some error. For the case of $\mathbb{R}^D$, [23, 29] have shown that AA can be achieved in the asynchronous model if and only if the maximum number of corruptions satisfies $t < n/(D+2)$. Nowak and Rybicki [27] have generalized the definitions of CC and AA to general convexity spaces: for an input convexity space with so-called Helly number ω (i.e., $D+1$ for $\mathbb{R}^D$), they have shown that the thresholds of $t < n/\omega$ for the synchronous and $t < n/(\omega+1)$ for the asynchronous setting are required for CC to be solvable when the input space is a convex geometry, which is a restricted class of convexity spaces. For instance, contrary to intuition, the standard convexity notion on $\mathbb{R}^D$ is not a convex geometry. Moreover, they matched the bound for the synchronous setting with a protocol, this time for all convexity spaces. They also considered asynchronous AA for graphs under monophonic convexity under the assumption of at most $t < n/(\omega+1)$ Byzantine parties.

In this work, we primarily investigate whether a randomized CC protocol for arbitrary convexity spaces can be achieved in the so-called *best-of-both-worlds model*. In particular, when running in a synchronous setting, it shall achieve the presumed optimal resilience threshold $t_s < n/\omega$, while if the network is asynchronous, it should still offer resilience up to a lower threshold of $t_a \leq t_s$ corruptions. This network-agnostic approach mitigates important shortcomings of the two classical models: the synchronous model allows for higher resilience thresholds, but in practice, the maximum delay Δ will often be violated during times of increased network load or outages, while the asynchronous model deals with such situations with ease, but at the expense of lower resilience thresholds. Primitives with such network-agnostic resilience guarantees have received increased attention in recent years [4, 7, 10, 13, 19, 20, 25].

1.1 Our Contributions

We answer our main question in the affirmative, by giving a protocol achieving CC that is resilient against t_s corruptions in the synchronous case and t_a corruptions in the asynchronous one, as long as $n > \max(\omega \cdot t_s, \omega \cdot t_a + t_s, 2 \cdot t_s + t_a)$, where $\omega \geq 2$ is the Helly number of the convexity space. The required conditions allow for a trade-off between the two expected optimal "point-wise" resilience bounds: setting $t_a = t_s$ gives an asynchronous protocol resilient to $t_a < n/(\omega+1)$ corruptions, while setting $t_a = 0$ yields a protocol resilient to $t_s < n/\omega$ corruptions if synchronicity holds and no corruptions if it fails, whilst maintaining correctness even in the latter case. It is worth mentioning that the case $\omega = 1$ corresponds to convexity spaces where there exists an element $v \in V$ contained in all non-empty convex sets. In this case, the trivial protocol where parties unconditionally output v achieves CC.

Our protocol is inherently randomized (which is needed, because of [18]), and it makes use of cryptographic setup — namely digital signatures. Assuming setup is necessary to tolerate $t_s < n/\omega$ corruptions in the synchronous model, in particular for $\omega = 2$. This can be easily inferred from various impossibility results (e.g., [29]). When ω increases, however, the quantity



(a) $\omega = 2$ (b) $\omega = 3$

Figure 1: Feasibility of achieving CC resilient against t_s corruptions if the network is synchronous and $t_a \leq t_s$ corruptions if it is asynchronous. For a fixed value of $n = 100$, the two plots depict in green the set of pairs (t_s, t_a) for which a protocol exists as percentages of n:the condition $n > \max(\omega \cdot t_s, \omega \cdot t_a + t_s, 2 \cdot t_s + t_a)$. The two black lines correspond to the point-wise optimal resilience thresholds $n > \omega \cdot t_s$ and $n > (\omega + 1) \cdot t_a$ required in the purely synchronous and asynchronous models respectively. The condition $n > \max(\omega \cdot t_s, \omega \cdot t_a + t_s, 2 \cdot t_s + t_a)$ can be understood as $n > 2 \cdot t_s + t_a$ for $\omega = 2$ and $n > \max(\omega \cdot t_s, \omega \cdot t_a + t_s)$ for $\omega \geq 3$. The two cases are depicted above for $\omega = 2$ and $\omega = 3$.

t_s decreases, which enables us to discard the digital signatures. Hence, in the paper, we will briefly explain how this assumption can be removed when $\omega > 2$.

Moreover, we generalize the aforementioned impossibility results from convex geometries to all convexity spaces, so that $t < n/\omega$ is required for the synchronous case and $t < n/(\omega + 1)$ for the asynchronous one. We note that [27] also gives an impossibility result for general convexity spaces, this time in terms of the distinct Carathéodory number (which in general has no relation to the Helly number that we use to formulate our results), seemingly contradicting our findings. However, upon closer inspection we exhibit what we believe to be an error in the proof, implying that the correct bound is in terms of ω, and not the Carathéodory number. To prove these resilience lower bounds, we introduce what we call *adversarial families*, which are a unified framework for deriving impossibility proofs for convex agreement problems. For the best-of-both-worlds model, we prove that the conditions required by our CC protocol are necessary: $n > \max(\omega \cdot t_s, 2 \cdot t_s + t_a, \omega \cdot t_a + t_s)$ must hold. Together with our protocol, our impossibility results complete the landscape of tractability for the purely synchronous, purely asynchronous, and the best-of-both-worlds model. See Figure 1 for the conditions illustrated.

Our results lead to multiple secondary contributions, which may be of independent interest.

Namely, we design the best-of-both-worlds variants of two communication primitives. The first one is an Agreement on a Core-Set primitive [5, 6], which allows parties to distribute their input values and obtain identical views. This result requires randomization. We also present a weaker, but deterministic variant, namely a best-of-both-worlds implementation of Gather [2, 12], which relaxes the guarantee of identical views to intersecting views. Both primitives provide stronger guarantees than their standard definitions, i.e., if they run in a synchronous network, the views contain all honest parties' inputs.

Moreover, we identify a core issue in the asynchronous AA protocol for chordal graphs with monophonic convexity of [27], and provide an alternative algorithm using our Gather variant as an underlying building block, that additionally achieves best-of-both-worlds resilience guarantees, requiring that $n > \omega \cdot t_s + t_a$ where ω denotes the size of the largest clique in the

input graph (which equals the Helly number in this case). For this case, it is unknown whether the condition $n > \omega \cdot t_s + t_a$ is tight. Note however that achieving a better trade-off when the agreement conditions are relaxed is still an outstanding problem even for $\mathbb{R}^D$, left open in [20]. Notably, the fact that our randomized CC protocol achieves a better resilience trade-off implies that a lower bound of $n > \omega \cdot t_s + t_a$ is unlikely to have scenarios-based proofs, as such proofs often apply to the randomized case with minor modifications.

Finally, we give a simple family of graphs $\{G_d\}_{d \geq 1}$ for which no deterministic asynchronous protocol achieves geodesic convex hull validity and agreement within graph distance d for graph G_d, even when corruptions are restricted to at most two parties crashing. To the best of our knowledge, thus far graph approximate agreement has only been considered when agreement is to be achieved within graph distance 1. This contributes to a long line of research aiming to understand the classes of graphs that admit wait-free AA algorithms under a plethora of agreement and validity conditions, e.g. [3, 22].

1.2 Related Work

Synchronous CC was introduced by Vaidya and Garg in [29] (see also the journal version [24]), on multidimensional real values. The most similar work to ours is that of Nowak and Rybicky [27], which generalize the problem of synchronous CC to abstract convexity spaces, and also its relaxed version AA for the asynchronous model. Our work addresses an open question raised in [27] on whether there exists an input convexity space for which the optimal resilience threshold of AA depends on the Carathéodory number and not on the Helly number, by showing that, in fact, the asynchronous resilience threshold is actually independent of the Carathéodory number; we provide a lower bound based on the Helly number instead. In addition, we identify a core issue in the deterministic algorithm of [27] achieving asynchronous AA on chordal graphs, and we show an alternative correct approach.

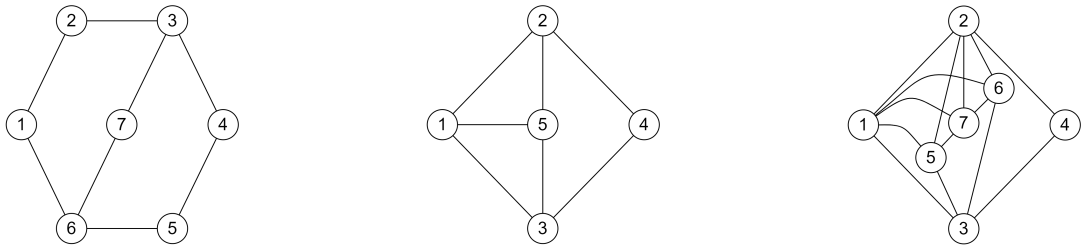
Our work additionally stands out from previous research on CC problems since we strive to achieve protocols with best-of-both-worlds guarantees. This research direction has attracted significant attention in recent years and has led to notable advancements in various areas. For instance, there have been protocols designed for AA on real numbers [19] and its multi-dimensional variant [20], for Byzantine Agreement [7, 13], State-Machine Replication [8] and also Multi-Party Computation [4, 10, 13]. In our work, we build upon and extend techniques from prior works such as [19, 20], but also from prior works focused solely on asynchronous AA [1,23,29]. Most notably, we present the first protocol in the best-of-both-worlds model achieving an optimal resilience trade-off with a non-linear boundary (see Figure 1b).

2 Preliminaries

In the following, given a non-negative integer k, write $[k]$ for the set $\{1, 2, \ldots, k\}$.

2.1 Model

Consider n parties denoted by $P_1, P_2, \ldots, P_n$ running a protocol in a fully-connected network, where links model authenticated channels. A synchronous network ensures that the parties' clocks are perfectly synchronized and that each message is delivered within a publicly known amount of time Δ. If any of these two guarantees fails, then the network is asynchronous. We assume that the parties are not aware a priori of the type of network the protocol is running in. In addition, we assume an adaptive adversary that may corrupt at most t_s parties if the network is synchronous, and at most t_a parties if the network is asynchronous. Corrupted parties permanently become Byzantine, meaning that they deviate arbitrarily, even maliciously, from the protocol. Moreover, the adversary may control the message delivery schedule, subject to the conditions of the network type. We will make use of a public key infrastructure (PKI),



(a) Graph G_1. (b) Graph G_2. (c) Graph G_3.

Figure 2: Illustration of geodesic (G) and monophonic (M) convex hulls. In G_1, $\langle\{1,3\}\rangle = \{1,2,3\}$ for G and $\{1,\ldots,7\}$ for M. In all three graphs, $\langle\{1,4\}\rangle$ consists of all vertices for both G and M, despite not all nodes always lying on a shortest/induced path between 1 and 4.

and a secure signature scheme. For simplicity, we assume that the signatures are perfectly unforgeable.

2.2 Abstract Convexity Spaces

Given a nonempty set V, also called the *universe*, an *abstract convexity space* on V is a family $\mathcal{C}$ of subsets of V such that $\varnothing, V \in \mathcal{C}$ and $\mathcal{C}$ is closed under finite intersections; i.e., whenever $A, B \in \mathcal{C}$, it also holds that $A \cap B \in \mathcal{C}$. The sets in $\mathcal{C}$ are regarded as *convex sets*. For instance, when $V = \mathbb{R}^D$, one possible $\mathcal{C}$ consists of all sets satisfying the condition that the straight-line segment joining any two points in the set is also included in the set. Note that this yields the standard convexity notion on $\mathbb{R}^D$. However, this is not the only way to define a convexity space on $\mathbb{R}^D$ that is consistent with the (arguably minimal) requirements of the definition; e.g., take $\mathcal{C}$ to be the family of "box" subsets of $\mathbb{R}^D$; i.e., subsets of the form $I_1 \times \ldots \times I_D$, where $(I_i)_{i\in[D]}$ are intervals on the real line. A central notion in convexity theory is that of convex hulls. In particular, the *convex hull* of any (not necessarily convex) set $S \subseteq V$ is the intersection $\langle S\rangle$ of all convex sets $C \in \mathcal{C}$ containing S. Note that, as convexity spaces are required to be closed under intersection, the convex hull is, indeed, convex. In $\mathbb{R}^D$ under straight-line convexity, hulls correspond to the usual notion of Euclidean convex hulls, while under "box"-convexity they correspond to so-called "bounding-boxes"; i.e., take the box spanning the region between the minimum and maximum coordinate in the set along each axis. Note that the convex hull operator is idempotent, also called a closure operator, i.e., $\langle\langle S\rangle\rangle = \langle S\rangle$. Moreover, note that a set is convex if and only if $S = \langle S\rangle$.

One relevant notion for our work will be that of *extreme* points. Namely, given a non-necessarily convex set $S \subseteq V$, the set $ex(S) = \{s \in S \mid \langle S \setminus s\rangle \subsetneq \langle S\rangle\}$ is the set of points in S any of whose removal would "shrink" the convex hull. Set S is called *free* if $\langle S\rangle = ex(S)$. Note that free sets are necessarily convex, as $\langle S\rangle = ex(S) \subseteq S \subseteq \langle S\rangle$, from which $\langle S\rangle = S$. Equivalently, S is free if and only if S is convex and $S = ex(S)$.

2.2.1 Graphs Convexity Spaces

Unless stated otherwise, all graphs considered in this paper are finite, undirected, connected and simple. Given a graph $G = (V, E)$, one can define various convexity spaces on V. Similarly to $\mathbb{R}^D$, one would want the convex hull of a set of nodes to represent a set of "good" gathering points. Two prominent examples that have been extensively considered in the literature are the so-called *geodesic* and *monophonic* convexities. We begin with geodesic convexity: a subset $C \subseteq V$ is *geodesically convex* if, for any two vertices $u, v \in C$, and any shortest path P from u to v in G, all vertices in P are in C. This can be thought of as a discrete version of the standard straight-line convexity notion for $\mathbb{R}^D$. Note, however, that unlike in $\mathbb{R}^D$, the shortest path might not be unique. *Monophonic* convexity on G is defined analogously, relaxing the

paths considered from shortest paths to induced paths (also called chordless paths); i.e., paths with no short-circuit edges. In both cases, convex hulls are defined as before, as the intersection of all convex sets containing the set.

A tempting alternative definition of the convex hull, motivated by the slogan "a good gathering point" would just take all vertices that lie on some shortest/induced path between two vertices in the set. This is, however, a distinct notion. For geodesic convexity, consider for instance the graph G_1 in Figure 2a, where all nodes except 7 lie on a shortest path between 1 and 4. However, the set of all nodes excluding 7 is not convex, as 7 is on a shortest path between 3 and 6, so the convex hull actually consists of all nodes. Hence, arranging a meeting point that lies on some shortest path between two parties can not be modelled through convexity alone. Figure 2b gives an example for monophonic convexity where 5 does not lie on any induced path between 1 and 4, but it is in the hull $\langle\{1,4\}\rangle$. Write $F(S)$ for the set of all nodes lying on some shortest/induced path between nodes in S, then, the convex hull $\langle S\rangle$ is the least fixed point of F containing S. Operationally, this means that $\langle S\rangle$ can be computed by starting with S and repeatedly performing $S := F(S)$ until equality is reached; i.e., take the nodes lying on some shortest/induced path and add them to the set, repeating until the set no longer changes. E.g., in Figure 2c, for both convexity notions the set $S = \{1,4\}$ would evolve as follows: $\{1,4\} \to \{1,\ldots,4\} \to \{1,\ldots,6\} \to \{1,\ldots,7\} = \langle\{1,4\}\rangle$.

An important observation, to become instrumental later on, is that free sets correspond to cliques for our two graph convexity notions. The proof is enclosed in Appendix A.

Lemma 1. *Let $G = (V,E)$ be a graph and $S \subseteq V$ be a subset of its vertices. Then, under both geodesic or monophonic convexity S is a free set if and only if S induces a clique in G.*

2.2.2 The Helly Number ω of a Convexity Space

The following seminal result in convexity theory concerns $\mathbb{R}^D$ with straight-line convexity.

Theorem 2 (Helly's Theorem)**.** *Consider a finite collection of at least $D+1$ convex sets in $\mathbb{R}^D$ with straight-line convexity. If every $D+1$ of them intersect, then all of them intersect.*

Helly's Theorem implies that, for instance, any collection of at least four disks in $\mathbb{R}^2$ with triple-wise non-empty intersections has a non-empty intersection. Notice that the same would not hold if $D+1$ was replaced by D; e.g., one can draw three disks in $\mathbb{R}^2$ that pair-wise intersect but have no point common to all three. One might wonder what about box convexity? In that case, $D+1$ can be replaced by 2. For instance, this means that any collection of at least three rectangles in $\mathbb{R}^2$ where any two intersect has a non-empty intersection, in contrast to disks. This number, which is $D+1$ for straight-line convexity and 2 for box convexity is what is known as the Helly number ω of the convexity space. We make this more general in the following: consider a convexity space $\mathcal{C}$. An *m-Helly family* for $\mathcal{C}$ is a collection of m convex sets $C_1, C_2, \ldots, C_m \in \mathcal{C}$ such that their intersection is the empty set, but the intersection of any $m-1$ of them is non-empty. Formally, $\cap_{j=1}^{m} C_j = \varnothing$, but $\cap_{j\neq i} C_j \neq \varnothing$ for any $i \in [m]$. The *Helly number* of $\mathcal{C}$ is the maximum number ω such that there exists an ω-Helly family for $\mathcal{C}$. Note that for some spaces there will exist arbitrarily-large Helly families, in which case the Helly number is undefined.[1]

2.3 Chordal Graphs and Convex Geometries

A graph $G = (V,E)$ is *chordal* if it has no induced cycle of length greater than three. A vertex $v \in V$ is *simplicial* if its neighbors in G form a clique. Chordal graphs admit many equivalent

[1] We do not directly concern ourselves with this case in the statement of our main results, but note that our reasoning often still applies when ω is undefined, for instance when deriving impossibility results. For the rest of this work we assume that the spaces we consider have a well-defined Helly number ω.

definitions, most notably they are the graphs that have a simplicial vertex whose removal yields another chordal graph. They are also the graphs that admit a *perfect elimination order*, which is a total order $\succ$ on V such that any $v \in V$ is simplicial in the subgraph induced by $\{u \in V \mid u \succeq v\}$. One should read $u \succ v$ as "u is eliminated after v." A graph G is *distance-hereditary* if distances in any connected induced subgraph are the same as in the original graph. Equivalently, G is distance-hereditary if every induced path is a shortest path. A graph is *Ptolemaic* if it is chordal and distance-hereditary.

An abstract convexity space $\mathcal{C}$ on universe V is a *convex geometry* if it additionally satisfies that, for all convex sets $C \subsetneq V$, there exists $v \in V \setminus C$ such that $C \cup \{v\}$ is convex. Note that this is a non-trivial requirement; e.g., $\mathbb{R}^D$ with neither straight-line convexity nor box convexity is a convex geometry. However, two notable examples arise when we consider graphs endowed with geodesic or monophonic path convexity. Namely, as shown in [17], the monophonic convexity of a graph G is a convex geometry if and only if G is chordal and the geodesic convexity of G is a convex geometry if and only if G is Ptolemaic. Hence, our work on graph convex geometries will be concerned with chordal graphs. Notice that all graphs for which geodesic convexity is a convex geometry are Ptolemaic, and hence distance-hereditary, meaning that the two convexity notions coincide on such graphs. As a result, our results for chordal graphs will target monophonic convexity. Unless stated otherwise, for chordal graphs by "convex" we mean "monophonically convex."

2.4 Convex Agreement Problems

A *convex agreement problem* is defined for a convexity space $\mathcal{C}$ over a universe V; e.g., $\mathbb{R}^D$ with straight-line convexity, or a graph $G = (V, E)$ with either geodesic or monophonic convexity. Each party P starts with an input $v_{in}^P \in V$ and should produce an output v_{out}^P. Ideally, all outputs should match, and this common output should be in the convex hull of the inputs. However, one has to consider the presence of the Byzantine parties. Hence, an agreement problem is defined by a collection of less strict properties that protocols solving it should satisfy: one validity, one agreement, and one termination condition. Write V_{in} and V_{out} for the set of inputs and respectively outputs of the *honest* parties. A convex agreement problem has the following validity condition:

Convex-Hull Validity: $V_{out} \subseteq \langle V_{in} \rangle$ (honest outputs are in the convex hull of honest inputs).

For the agreement condition, there are multiple natural options to choose from, such as:

Exact Agreement (Consensus): $|V_{out}| \leq 1$ (no two honest parties obtain different outputs).
***k*-Set Agreement**: $|V_{out}| \leq k$ (no $k+1$ honest parties obtain pairwise distinct outputs).

Note that 1-Set Agreement is Consensus. When the convexity space has more structure, one can define more specialized agreement conditions depending on the application. Most naturally, if *dist* is a metric on universe V, then one can define:

Distance-*d* Agreement: $\max_{p,q \in V_{out}} dist(p, q) \leq d$ (no two honest parties obtain outputs more than distance d away from each other).

Two immediate examples are $\mathbb{R}^D$ with straight-line convexity, where the Euclidean distance is the natural choice for *dist*, and a graph $G = (V, E)$ with either geodesic or monophonic path convexity, where graph distance is the most natural choice. Another possible condition introduced in [27] to capture a certain type of semilattice agreement is the following:

Free-Set Agreement: V_{out} is a free set (i.e., $\langle V_{out} \rangle = ex(V_{out})$).

For graphs, by Lemma 1, this corresponds to V_{out} inducing a clique in G, so it is equivalent to Distance-1 Agreement. For $\mathbb{R}^D$ with straight-line convexity, the only free sets are the singletons, so Free-Set Agreement coincides with Consensus, while Distance-1 Agreement is more lenient.

In the original application to semilattice agreement [27], Free-Set Agreement translates to values in V_{out} forming a chain, which none of the other notions can capture.

Finally, let us discuss the termination requirements of the protocol. There are two flavors, one for deterministic protocols, and one for randomized protocols, listed below:

Termination: all honest parties obtain outputs.
Probabilistic Termination: the probability that some honest party has not obtained output after T time units tends to 0 as $T \to \infty$.

In this work, we are mainly concerned with two convex agreement problems: *Convex Consensus* (CC) and *Approximate Agreement* (AA); defined for a convexity space $\mathcal{C}$. A protocol solving CC or AA has to satisfy Convex Hull Validity and Termination (Probabilistic Termination for randomized protocols). For CC, Exact Agreement has to be satisfied, while for AA we require Distance-d Agreement for some parameter d. The convexity space at hand dictates the distance notion; i.e., Euclidean distance for $\mathbb{R}^D$ and graph distance for graphs. For graphs, unless stated otherwise, we are interested in the case $d = 1$, coinciding with Free-Set Agreement.

We say that a protocol Π solving CC or AA is (t_s, t_a)-*resilient* if it satisfies the relevant properties in the presence of at most t_s corrupted parties in the synchronous setting and $t_a \leq t_s$ corrupted parties in the asynchronous one.

3 Tight Resilience Bounds Using the Helly Number

In this section, we introduce the notion of *adversarial families* to prove impossibility results for convex agreement problems. Our main result is that CC in the best-of-both-worlds model requires each of the following conditions to hold: $n > \omega \cdot t_s$, $n > 2 \cdot t_s + t_a$, and $n > \omega \cdot t_a + t_s$.

We begin by showing that the conditions $n > \omega \cdot t$ and $n > (\omega + 1) \cdot t$ are necessary in the synchronous and resp. asynchronous model, where ω is the Helly number of the convexity space. Note that the purely synchronous threshold immediately implies that the condition $n > \omega \cdot t_s$ is necessary in the best-of-both-worlds model. Afterward, we move towards showing that the other conditions are required in the best-of-both-worlds model.

In the next section, we will show that these bounds are tight, by giving CC protocols assuming $n > \max(\omega \cdot t_s, 2 \cdot t_s + t_a, \omega \cdot t_a + t_s)$. All our (possibility and impossibility) results are general, in that they make no assumption on the convexity space other than its Helly number.

We end this section by showing how adversarial families can be used to streamline known impossibility proofs for agreement problems other than consensus, such as AA in $\mathbb{R}^D$ with straight-line convexity. We note that a previously-known resilience bound [27, Theorem 11] given in terms of the distinct Carathéodory number of the space is incompatible with our results, as in general there is no relation between the Carathéodory number and the Helly number. However, this bound appears to be incorrect (detailed discussion in Appendix B.1).

Consider a convexity space $\mathcal{C}$ with Helly number ω defined on a universe V. Consider a family $\mathcal{A} = \{A_1, \ldots, A_m\}$ consisting of m non-empty pairwise-disjoint convex sets $A_i \in \mathcal{C}$ and write $A = \cup\mathcal{A}$. Then, $\mathcal{A}$ is m-*adversarial* if $A_i = \cap_{\ell \neq i} \langle A \setminus A_\ell \rangle$ for all $i \in [m]$.[2] Note that the pairwise-disjoint condition is equivalent to $\cap_{\ell=1}^{m} \langle A \setminus A_\ell \rangle = \varnothing$.[3] The following technical lemma, and the two following it, will be the main tool used to get impossibility results. The techniques used in its proof are similar in spirit to the proofs for $\mathbb{R}^D$ in [24].

Lemma 3. *Let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be an m-adversarial family for convexity space $\mathcal{C}$. Assume $n \geq m$ and that, moreover, $n \leq m \cdot t$ if the network is synchronous and $n \leq (m+1) \cdot t$ if the network is asynchronous. Then, any (deterministic or randomized) n-party protocol satisfying*

[2] This definition requires $m > 1$ to avoid taking the intersection of an empty collection of sets. However, for $m = 1$ all our results will hold if we assume that $\mathcal{A} = \{A\}$ is 1-adversarial for any convex set $A \neq \varnothing$. We will not discuss this technicality further and henceforth assume that $m \geq 1$ is well-defined.

[3] To see this, note that for $i \neq j$ we have $A_i \cap A_j = (\cap_{\ell \neq i} \langle A \setminus A_\ell \rangle) \cap (\cap_{\ell \neq j} \langle A \setminus A_\ell \rangle) = \cap_{\ell=1}^{m} \langle A \setminus A_\ell \rangle$.

Convex-Hull Validity and (Probabilistic) Termination will have a terminating execution where there are honest parties $P_1, \ldots, P_m$ such that the output v^i_{out} of party P_i satisfies $v^i_{out} \in A_i$.

The following two technical lemmas gives similar guarantees, this time for the hybrid, best-of-both-worlds model. The proof of the first is similar to that for $\mathbb{R}$ in [19], while that of the second is an extension of the asynchronous part of Lemma 3.

Lemma 4. *Assume a convexity space $\mathcal{C}$ admitting a 2-adversarial family $\mathcal{A} = \{A_1, A_2\}$. Assume $2 \leq n \leq 2 \cdot t_s + t_a$. Let Π denote an arbitrary (deterministic or randomized) protocol achieving Convex-Hull Validity and (Probabilistic) Termination for at most t_s corruptions when the network is synchronous and at most t_a corruptions when it is asynchronous. Then, Π has a terminating execution where the outputs v^1_{out} and v^2_{out} of two honest parties satisfy $v^1_{out} \in A_1$ and $v^2_{out} \in A_2$.*

Lemma 5. *Let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be an m-adversarial family for convexity space $\mathcal{C}$. Assume that $m \leq n \leq m \cdot t_a + t_s$. Then, any (deterministic or randomized) n-party protocol satisfying Convex-Hull Validity and (Probabilistic) Termination for at most t_s corruptions when the network is synchronous and at most t_a corruptions when the network is asynchronous will have a terminating execution where there are honest parties $P_1, \ldots, P_m$ such that the output v^i_{out} of party P_i satisfies $v^i_{out} \in A_i$.*

We now show a relationship between adversarial families and Helly families.

Lemma 6. *Consider a convexity space $\mathcal{C}$, then an m-adversarial family exists if and only if an m-Helly family exists. Hence, the size of the largest adversarial family for a convexity space equals its Helly number ω.*

Proof. First, consider an adversarial family $\mathcal{A} = \{A_1, \ldots, A_m\}$ for $\mathcal{C}$ and as usual write $A = \cup\mathcal{A}$. The family of sets $\langle A \setminus A_i \rangle_{i \in [m]}$ do not intersect, but any $m-1$ of them do, since for any i we assumed that $A_i = \cap_{\ell \neq i} \langle A \setminus A_\ell \rangle$ is non-empty, so it is an m-Helly family. Conversely, consider an m-Helly family; i.e., convex sets $C_1, \ldots, C_m \in \mathcal{C}$ that do not intersect, but any $m-1$ of them do. Define the family of non-empty convex sets $\mathcal{A} = \{A_1, \ldots, A_m\}$ where $A_i = \cap_{\ell \neq i} C_\ell$. Notice that for $i \neq j$ we have $A_i \cap A_j = \cap_{\ell \in [m]} C_\ell = \varnothing$, so the sets are pairwise disjoint. To show that $\mathcal{A}$ is an m-adversarial family, it remains to show that for all i it holds that $A_i = \cap_{\ell \neq i} \langle A \setminus A_\ell \rangle$. To see this, note that $A \setminus A_\ell = \cup\{A_1, \ldots, A_{\ell-1}, A_{\ell+1}, \ldots, A_m\}$ and that $A_{\ell'} \subseteq C_\ell$ for all $\ell' \neq \ell$, so $A \setminus A_\ell \subseteq C_\ell$. Since C_ℓ is convex, this means that $\langle A \setminus A_\ell \rangle \subseteq \langle C_\ell \rangle = C_\ell$. As a result, $\cap_{\ell \neq i} \langle A \setminus A_\ell \rangle \subseteq \cap_{\ell \neq i} C_\ell = A_i$. To also show that $A_i \subseteq \cap_{\ell \neq i} \langle A \setminus A_\ell \rangle$ just notice that $A_i \subseteq A \setminus A_\ell \subseteq \langle A \setminus A_\ell \rangle$ for all $\ell \neq i$. □

Note that a more restrictive definition of adversarial families where all the sets are singletons would not suffice to prove the previous, as in some spaces no singletons are convex.

To access the full power of Lemmas 3 and 5, which require n to be at least the size of the adversarial family, we would like that adversarial families of a certain size imply the existence of adversarial families of all smaller sizes. We show this in the following lemma.

Lemma 7. *Given a convexity space, if there exists an m-Helly family, then there exist m'-Helly families for any $1 \leq m' < m$. The same holds if "Helly" is replaced by "adversarial."*

Proof. It suffices to consider $m' = m-1$. If $C_1, \ldots, C_m$ is an m-Helly family, then one can check that $C_1, \ldots, C_{m-2}, (C_{m-1} \cap C_m)$ is an $(m-1)$-Helly family. For the latter, apply Lemma 6. □

We now leverage Lemmas 3 and 6 to get the following result, generalizing those in [27] by removing the strong requirement of a convex geometry.

Theorem 8. *Consider a convexity space $\mathcal{C}$ with Helly number ω. Assume $n \leq \omega \cdot t$ if the network is synchronous and $n \leq (\omega+1) \cdot t$ if the network is asynchronous. Then, there is no (deterministic or randomized) n-party protocol satisfying Convex-hull Validity and (Probabilistic) Termination such that the set of outputs of the honest parties is guaranteed to have size at most* $\min(n, \omega) - 1$.

Proof. Write $m = \min(n, \omega)$. By Lemma 6, there is an ω-adversarial family for $\mathcal{C}$. Since $m \leq \omega$, using Lemma 7, let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be an m-adversarial family for $\mathcal{C}$. Consider a protocol Π satisfying convex-hull validity and termination. By Lemma 3, there is a terminating execution of Π where the set of honest outputs contains $\{a_1, \ldots, a_m\}$ where $a_i \in A_i$. As sets in $\mathcal{A}$ are pairwise disjoint, this set has cardinality m, implying the conclusion. □

By similarly leveraging Lemmas 4 and 5, we similarly get the following impossibilities for the best-of-both-worlds model.

Theorem 9. *Consider a convexity space $\mathcal{C}$ with Helly number $\omega \geq 2$. Assume $2 \leq n \leq 2 \cdot t_s + t_a$ or $2 \leq n \leq \omega \cdot t_a + t_s$. Then, there is no (deterministic or randomized) n-party protocol satisfying Convex-Hull Validity, (Probabilistic) Termination and Exact Agreement can simultaneously tolerate at most t_s corruptions when the network is synchronous and at most t_a corruptions when the network is asynchronous. For the case $2 \leq n \leq \omega \cdot t_a + t_s$, the same holds even for the weaker condition of agreeing on at most* $\min(n, \omega) - 1$ *values.*

We conclude by showing that adversarial families can also be used to recover known impossibility results for $\mathbb{R}^D$ with straight-line convexity. The same can also be done to recover the requirement of $n > 2 \cdot t_s + t_a$ for $\mathbb{R}$ in the best-of-both-worlds model [19].

Theorem 10. *Consider $\mathbb{R}^D$ with straight-line convexity and let $d > 0$ be arbitrary. Assume $n \leq (D+1) \cdot t$ if the network is synchronous and $n \leq (D+2) \cdot t$ if the network is asynchronous. Then, there is no (deterministic or randomized) n-party protocol satisfying Convex-Hull Validity and Termination such that no two honest outputs are more than Euclidean distance d apart.*

4 Achieving Convex Consensus

We now describe a construction achieving CC in the best-of-both-worlds model that matches our previous resilience lower bounds. Concretely, we focus on proving the following theorem.

Theorem 11. *Assume that $t_a \leq t_s$ and $n > \max(\omega \cdot t_s, 2 \cdot t_s + t_a, \omega \cdot t_a + t_s)$. Then, there is a protocol achieving (t_s, t_a)-resilient CC.*

To set up the intuition for our construction, we recall the outline of the synchronous protocol on $\mathbb{R}^D$ of [29]. The synchronous model offers powerful communication primitives (i.e., Synchronous Reliable Broadcast [16]), that enable the parties to reliably distribute their values and obtain an *identical view*. This view consists of a set of value-sender pairs, out of which $n - t_s$ correspond to honest parties. Then, honest parties may derive a *safe area* inside their inputs' convex hull by intersecting the convex hulls of all subsets of $n - t_s$ values received, as defined below. For the convenience of avoiding working with multisets, our protocols will work with sets of value-sender pairs. For this purpose, we extend the convex hull operator to such sets straightforwardly: ignore party identities and take the convex hull of the values.

Definition 12 (Safe Area)**.** *Let $\mathcal{M}$ denote a set of value-sender pairs. For a given k, $safe_k(\mathcal{M}) := \bigcap_{M \in restrict_k(\mathcal{M})} \langle M \rangle$, where $restrict_k(\mathcal{M}) := \{M \subseteq \mathcal{M} : |M| = |\mathcal{M}| - k\}$.*

Specifically, if parties received the (same) set $\mathcal{M}$ of $n - t_s + k$ value-sender pairs, they compute their safe area as $safe_k(\mathcal{M})$. We will later show (in a more general form) that, since $n > \omega \cdot t_s$, the safe area obtained is non-empty. Therefore, any value in the common safe area is a valid choice. Hence, parties may output any such value chosen by some deterministic criterion.

Identical Views in Asynchrony. Building towards our solution achieving best-of-both worlds guarantees, we first identify the challenges posed by translating the outline above to the purely asynchronous model (where $t_s = t_a$ and $n > (\omega + 1) \cdot t_a$). Assuming a primitive that enables parties to obtain an identical vrew of $n - t_a$ value-sender pairs, CC can be achieved in a similar manner. Out of the set $\mathcal{M}$ of $n - t_a$ pairs agreed upon, at most t_a may be corrupted. Then, honest parties derive a safe area inside their inputs' convex hulls by computing $safe_{t_a}(\mathcal{M})$, and may afterward take a deterministic decision to obtain the same output.

While achieving an identical view deterministically is impossible in this model [18], allowing randomization leads to a simple solution. Namely, we employ a primitive introduced in [5] achieving *Agreement on a Core-Set* (ACS) when up to $t_a < n/3$ of the parties involved are corrupted, which suffices for our case of $\omega \geq 2$. Roughly speaking, an ACS protocol assumes that each party holds a value meant to be distributed, and enables the parties to obtain the same set $\mathcal{M}$ of $n - t_a$ value-sender pairs. By utilizing the (randomized) ACS protocol presented in [6, Section 4], we achieve asynchronous CC with optimal resilience, in constant expected number of rounds, proving the lower bound $n > (\omega + 1) \cdot t_a$ to be tight as well.

Maintaining the Advantages of Synchrony. While the standard definition of ACS makes obtaining identical views possible in an asynchronous network as well, the synchronous network still has an advantage that will be crucial for matching the point-wise optimal resilience thresholds for CC. Namely, the key insight on why CC can be achieved up to $t_s < n/\omega$ corruptions in the synchronous model, while $t_a < n/(\omega+1)$ is necessary in the asynchronous one, is that the former ensures all honest values are delivered. Intuitively, in the asynchronous setting, t_a corrupted parties may *replace* t_a honest parties: the messages of these honest parties get delayed for sufficiently long, while the t_a corrupted parties follow the protocol correctly, but with inputs of their choice. Hence, to achieve hybrid CC under the resilience condition $n > \max(\omega \cdot t_s, 2 \cdot t_s + t_a, \omega \cdot t_a + t_s)$, we require an additional property from ACS: if the network is synchronous, all honest values must be included in the output set agreed upon. Consequently, we propose the following definition for ACS in the best-of-both-worlds model.

Definition 13 (Agreement on a Core Set)**.** *Let* Π *be a protocol where every party* P *holds an input* v_P *and may output a set of value-sender pairs* $\mathcal{M}_P$*. We consider the following properties.*

- ***Validity****: Let* P *and* P' *be two honest parties. If* $(v', P') \in \mathcal{M}_P$*, then* $v' = v_{P'}$*.*
- ***Consistency****: Let* P *and* P' *be two honest parties. If* $(v_1'', P'') \in \mathcal{M}_P$ *and* $(v_2'', P'') \in \mathcal{M}_{P'}$*, then* $v_1'' = v_2''$*.*
- ***Exact Agreement****: Let* P *and* P' *be two honest parties, and assume they obtain outputs* $\mathcal{M}_P$ *and* $\mathcal{M}_{P'}$ *respectively. Then,* $\mathcal{M}_P = \mathcal{M}_{P'}$*.*
- ***T-Output Size****: If an honest party* P *obtains output* $\mathcal{M}_P$*, then* $\left|\mathcal{M}_P\right| \geq n - T$*.*
- ***Honest Core****: If an honest party* P *obtains output* $\mathcal{M}_P$*, then* $(v_{P'}, P') \in \mathcal{M}$ *for every honest party* P'*.*

Then, we say that Π *is a* (t_s, t_a)*-resilient* ACS *protocol if it achieves the following:*

- *Validity, Consistency, Exact Agreement,* t_s*-Output Size, Probabilistic Termination and Honest Core when running in a synchronous network and at most* t_s *parties are corrupted;*
- *Validity, Consistency, Exact Agreement,* t_s*-Output Size,*[4] *Probabilistic Termination when running in an asynchronous network and at most* t_a *parties are corrupted.*

This way, if parties use an ACS protocol to distribute their values, they obtain the same set $\mathcal{M}$ of $n - t_s + k$ value-sender pairs. If the network is asynchronous, at most t_a of these values are corrupted. In contrast, if the network is synchronous, the Honest Core property will ensure that at most k of these values are corrupted. To take both cases into account, parties locally

[4]This is intentional: we do not require the stronger property of t_a-Output Size.

compute their safe areas as $S := safe_{\max(k,t_a)}(\mathcal{M})$ and deterministically decide on an output $s \in S$. Note that, by definition of the safe area, S is indeed inside the honest inputs' convex hull. In addition, since the input space has Helly number ω and $n > \max(\omega \cdot t_s, 2 \cdot t_s + t_a, \omega \cdot t_a + t_s)$, the safe area obtained is still non-empty, so such s can be chosen. The proof of this result, stated below, is enclosed in Appendix D.

Lemma 14. *Assume $n > \max(\omega \cdot t_s, \omega \cdot t_a + t_s)$, and that $\mathcal{M}$ is a set of $n - t_s + k$ value-party pairs, where $0 \leq k \leq t_s$. Then, $safe_{\max(k,t_a)}(\mathcal{M}) \neq \varnothing$.*

Achieving ACS. We now focus on describing a concrete construction of a (t_s, t_a)-resilient ACS protocol. In particular, we describe the proof of the theorem below.

Theorem 15. *Let n, t_s, t_a be such that $n > 2 \cdot t_s + t_a$ and $t_a \leq t_s$. Then, there is a protocol Π_{ACS} achieving (t_s, t_a)-resilient ACS.*

When $t_s, t_a < n/3$ (which is suitable for $\omega \geq 3$), one can achieve this stronger variant of ACS in the best-of-both-worlds model even without using a public key infrastructure, by making a few adjustments to the ACS protocol of [6]. We present this protocol in detail in Appendix C.2.

Building such a primitive under the weaker resilience assumption $n > 2 \cdot t_s + t_a$ is however significantly more challenging. Simple adjustments to the outline of [6] will not suffice anymore, even when employing building blocks with optimal-resilience best-of-both-worlds guarantees. The insight behind this claim is that such primitives would only be able to provide synchronous guarantees if all conditions are met, i.e., if honest parties are ready to join these subprotocols simultaneously. On the other hand, the outline of [6] cannot ensure this premise.

Hence, we use a different approach for achieving ACS with Honest Core when $n > 2 \cdot t_s + t_a$. Our approach hinges on two building blocks. The first is a hybrid Byzantine Agreement (BA) protocol, as defined below.

Definition 16 (Byzantine Agreement)**.** *Let Π be a protocol where every party P holds a bit as input and may output a bit. We consider the following properties:*

- ***Weak Validity**: If all honest parties hold input b, no honest party outputs $b' \neq b$.*
- ***Exact Agreement**: Let P and P' be two honest parties, and assume they obtain outputs b and b' respectively. Then, $b = b'$.*

Then, Π is a (t_s, t_a)-resilient BA protocol if it achieves Weak Validity, Exact Agreement, and Probabilistic Termination when up to t_s of the parties are corrupted if it runs in a synchronous network, and when up to t_a of the parties are corrupted if running in an asynchronous network.

Concretely, we make use of the BA protocol of Blum, Katz and Loss [7].

Theorem 17 ([7])**.** *Let n, t_s, t_a be such that $n > 2 \cdot t_s + t_a$ and $t_a \leq t_s$. Then, there is a protocol Π_{BA} achieving (t_s, t_a)-resilient BA.*

The second building block will be a best-of-both-worlds version of a primitive known as *Gather* (GTHR) [2,12]. A GTHR protocol is a slightly weaker, but deterministic variant of ACS. Namely, GTHR relaxes the Exact Agreement property by only requiring that honest parties' output sets have at least $n - t_s$ values in common. We provide our best-of-both-worlds definition of GTHR below. Notice that our definition also requires the previous Honest Core property to hold under synchrony. Moreover, we require that parties obtain outputs simultaneously if the network is synchronous.

Definition 18 (Gather)**.** *Let Π be a protocol where every party P holds an input v_P and may output a set of value-sender pairs $\mathcal{M}_P$. We consider the following properties, additionally to those in Definition 13.*

- ***T*-Common Core**: *If all honest parties terminate, then* $\left|\bigcap_{P\ honest} \mathcal{M}_P\right| \geq n - T$.
- ***Simultaneous Termination***: *All the honest parties terminate and obtain an output at the same time.*

Then, we say that Π *is a* (t_s, t_a)*-resilient* GTHR *protocol if it achieves:*

- *Validity, Consistency, Honest Core*[5] *and Simultaneous Termination when running in a synchronous setting where at most* t_s *of the parties involved are corrupted;*
- *Validity, Consistency,* t_s*-Common Core and Termination when running in an asynchronous setting where at most* t_a *of the parties involved are corrupted.*

In Section C.3 we provide a construction achieving best-of-both-worlds GTHR, as stated below. Our protocol follows the outline of the initialization protocol Π_{init} of [20, Section 5], while making use of insights from [2], which focuses on the asynchronous definition of GTHR. The formal construction and its analysis are enclosed in Section C.4 of the Appendix.

Theorem 19. *Let* n, t_s, t_a *be such that* $n > 2 \cdot t_s + t_a$ *and* $t_a \leq t_s$*. Then, there is a protocol* Π_{GTHR} *achieving* (t_s, t_a)*-resilient* GTHR*.*

These building blocks enable us to sketch the proof of Theorem 15.

Proof Sketch of Theorem 15. The Π_{ACS} protocol proceeds as follows: parties distribute their values using Π_{GTHR}, obtaining consistent sets of value-party pairs that intersect in least $n - t_s$ pairs. If the network is synchronous, this common core will contain all honest value-sender pairs by the Honest Core property.

When party P obtains output $\mathcal{M}_{\mathsf{GTHR}}$ from Π_{GTHR}, it joins n invocations of Π_{BA}, one for each party P'. Party P inputs 1 in the invocation for P' if $\mathcal{M}_{\mathsf{GTHR}}$ contains some value from P' and 0 otherwise. Note that, if the network is synchronous, Π_{GTHR} provides Simultaneous Termination, hence honest parties join Π_{BA} simultaneously, and therefore the guarantees of Π_{BA} hold.

Then, regardless of the type of network, honest parties agree on a bit for each party. They will output the values sent by parties for whom the bit agreed upon is 1. The t_s-Common Core property of Π_{GTHR} ensures that there are at least $n - t_s$ parties for whom all honest parties join Π_{BA} with input 1, and therefore agree on output 1 (due to Weak Validity).

However, note that obtaining output 1 in the Π_{BA} invocation for P' does not mean that all honest parties have received a value from P' via Π_{GTHR}. Instead, the Weak Validity property of Π_{BA} only ensures that at least one honest party P has joined this invocation with input 1, and hence has received a value from P' via Π_{GTHR}. We will then make use of an additional property provided by our implementation of Π_{GTHR}: if parties wait for sufficiently long, they will receive the missing values as well.[6]

This way, all honest parties output the same set of at least $n - t_s$ values, meaning that Probabilistic Termination, Exact Agreement and t_s-Output Size hold. Note that Validity and Consistency follow immediately from Π_{GTHR} achieving these properties. In addition, if the network is synchronous, the output set will include all honest values, so the Honest Core property is also ensured. □

Achieving CC. We now provide the formal code of our protocol achieving CC.

Protocol Π_{CC}

Code for party P with input v_{in}

1: Join Π_{ACS} with input v_{in}. Upon obtaining output $\mathcal{M}$:

[5] Note that this implies t_s-Common Core, so we do not ask for it separately.

[6] Intuitively, this is because in Π_{GTHR} parties distribute their values via Reliable Broadcast.

2: $k := \left|\mathcal{M}\right| - (n - t_s)$; $S := safe_{\max(k,t_a)}(\mathcal{M})$
3: Choose $v_{out} \in S$ according to a predetermined, publicly available, deterministic rule.
4: Output v_{out} and terminate.

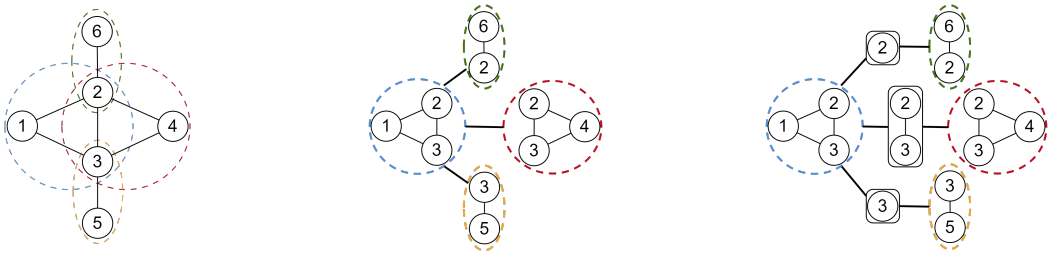
Proof of Theorem 11. Since parties distribute their values via Π_{ACS}, parties obtain the same set $\mathcal{M}$ of $n - t_s + k$ value-sender pairs, where $0 \leq k \leq t_s$. Then, regardless of whether the network is synchronous or asynchronous, $\mathcal{M}$ contains at most $\max(k, t_a)$ values from dishonest senders. Note that this holds when the network is synchronous due to Π_{ACS}'s Honest Core property. Hence, there is a subset $M_H \subseteq \mathcal{M}$ of size $\left|\mathcal{M}\right| - \max(k, t_a)$ only containing values from honest senders. By definition, $M_H \in restrict_{\max(k,t_a)}(\mathcal{M})$, so $S \subseteq \langle M_H \rangle$, meaning that the safe area S obtained by the honest parties is included in the honest inputs' convex hull. Lemma 14 ensures that S is non-empty, so parties agree on the same value v_{out} in the honest inputs' convex hull. Hence, Convex Validity, Exact Agreement and (Probabilistic) Termination hold. □

5 Approximate Agreement on Chordal Graphs

We investigate the previously known deterministic protocol that efficiently achieves chordal graph AA with monophonic path convexity [27, Section 4.2], finding that it is sadly incorrect. Recall that chordal graphs are precisely those for which monophonic path convexity induces a convex geometry. In this section, we will introduce a new protocol to solve the problem correctly, designed for the best-of-both-worlds setting.

The protocol of [27, Section 4.2] focuses strictly on the asynchronous setting, hence, when describing it, for brevity we assume a single number $t = t_a$. Roughly speaking, this proceeds in iterations. In each iteration, parties distribute their values via a weaker variant of GTHR, say Π. Instead of ensuring t-Common Core, Π ensures that the honest parties' output sets have *pair-wise* intersections of size $n - t$. Parties then compute safe areas and select some new value from their safe area as their new value. In order to ensure fast convergence, these new values are to be chosen carefully. This is done by relying on a special kind of tree decomposition admitted by chordal graphs, namely clique trees, to be introduced below. In particular, in tandem with the main algorithm, parties additionally run a tree AA protocol on the tree decomposition of the graph, using it to guide the main algorithm. Hence, at each step, each party computes both a "normal" and a "tree" safe area. It is then proven that each vertex in the tree safe area corresponds to at least one vertex in the graph safe area. Then, if the new value is to be taken from the center of the tree safe area, convergence can be ensured by an argument showing that the diameter of the tree safe area is roughly halved at each iteration. As we will show, it might actually be that no vertices in the center of the tree safe area appear in the graph safe area, preventing the algorithm from proceeding further. We now make the previous more exact by introducing the algorithm of [27, Section 4.2]. To do so, we first need to introduce clique trees.

Chordal graphs can be equivalently characterized as graphs admitting a *clique tree*. A clique tree $T = (V(T), E(T))$ for a graph $G = (V(G), E(G))$ is a tree whose vertices are subsets of V; i.e. $V(T) \subseteq 2^{V(G)}$. Every vertex of T has to induce a clique in G. Moreover, the following requirements have to be satisfied: (i) for all $v \in V(G)$ there is $b \in V(T)$ such that $v \in b$; (ii) for all $(u, v) \in E(G)$ there is $b \in V(T)$ such that $\{u, v\} \subseteq b$ and (iii) if $a, b \in V(T)$ and $v \in a \cap b$, then $v \in c$ for all $c \in V(T)$ residing on the unique $a - b$ path in T. Usually, one also requires that the cliques induced by vertices of T are maximal cliques. Note that, unlike general graphs, chordal graphs have a number of maximal cliques that is at most linear in the number of vertices in G, so they admit a clique tree with at most this many vertices. To illustrate, consider the chordal graph G in Figure 3a. The four maximal cliques are circled with dashed lines of different colors. One of the clique trees of G is given in Figure 3b. Namely, this is a



(a) Chordal graph G. (b) Possible clique tree for G. (c) Resulting expanded clique tree.

Figure 3: Chordal graph on which protocol $\Pi_{\mathsf{IncorrectChordal}}$ fails.

four-node star graph with the center in $\{1,2,3\}$ and leaves $\{2,6\}, \{2,3,4\}$ and $\{3,5\}$, in order from top to bottom.

Fix an arbitrary clique tree T of the input graph G. The algorithm will operate on what the authors call the "expanded clique tree": take T and subdivide each edge $(a,b) \in E(T)$ to get two edges $a - x$ and $x - b$, where $x = a \cap b$. Note that the expanded clique tree is also a clique tree for G, although the newly added vertices might no longer correspond to maximal cliques. This construction is exemplified for the clique tree in Figure 3b and Figure 3c.

We are now ready to give the protocol $\Pi_{\mathsf{IncorrectChordal}}$ from [27, Section 4.2], presented below. Note that it is only concerned with the asynchronous case, so only a single bound $t = t_s = t_a$ is used here instead of separate bounds t_s and t_a. Moreover, since the algorithm requires convex hulls both in graph G and in tree T, for $S \subseteq V(G)$ we write $\langle S \rangle_G$ for the hull of S in G, and for $S \subseteq V(T)$ we write $\langle S \rangle_T$ for the hull of S in T. We define $safe_k^G$ and $safe_k^T$ analogously. We note that now the values in the pairs of the message sets $\mathcal{M}$ are pairs of vertices (v, b). We, therefore, expand the definition of $\langle \mathcal{M} \rangle$ to such pairs: $safe_k^G$ refers to the values v in these pairs, while $safe_k^T$ to the values b.

Protocol $\Pi_{\mathsf{IncorrectChordal}}$

Code for party P with input $v_0 \in V(G)$

1: Select $b_0 \in V(T)$ arbitrarily such that $v_0 \in b_0$
2: **for** $\mathsf{it} = 1 \ldots \mathsf{max\_it} := \lceil \log_2 diam(T) \rceil + 2$ **do**
3: Join Π with input $(v_{\mathsf{it}-1}, b_{\mathsf{it}-1})$. Upon obtaining output $\mathcal{M}$ in Π:
4: $S_T := safe_t^T(\mathcal{M})$; $b_{\mathsf{it}} = center(S)$; $S_G := safe_t^G(\mathcal{M})$;
5: Select $v_{\mathsf{it}} \in S_G \cap b_{\mathsf{it}}$ arbitrarily.
6: **end for**
7: Output $v_{\mathsf{max\_it}}$ and terminate.

We next show an example where $\Pi_{\mathsf{IncorrectChordal}}$ does not execute correctly. In particular, it will be that for some honest party $S_G \cap b_{\mathsf{it}} = \varnothing$, implying that the party can not proceed further. To construct this, consider the graph G in Figure 3a and assume that its chosen clique tree is the one in Figure 3b. The expanded clique tree is then the one in Figure 3c. We assume $t = 3$ and that there are $n = 13 > \omega \cdot t = 4 \cdot t = 12$ parties. In our scenario, the $t = 3$ corrupted parties crash before taking part in the protocol. The other ten (honest) parties have inputs as follows: three parties have input 5, three parties have input 6, and four parties have input 4. For our input values $4, 5, 6$, there are unique nodes in the clique tree containing them. In particular, parties holding 4 will set $b_0 := \{2,3,4\}$ at the beginning of the protocol. Likewise, parties holding 5 will set $b_0 = \{3,5\}$ and parties holding 6 will set $b_0 = \{2,6\}$. Now, consider what subsequently happens in the protocol during the first iteration for an arbitrary party P holding input 4. Because the Byzantine parties have crashed, the set of messages $\mathcal{M}$ received by P is uniquely determined. In particular, in terms of the (v, b) payloads, $\mathcal{M}$ contains four pairs $(4, \{2,3,4\})$, three pairs $(5, \{3,5\})$, and three pairs $(6, \{2,6\})$. We then obtain $S_T = \langle \{\{3,5\}, \{2,3,4\}\} \rangle_T \cap$

$\langle\{\{2,6\},\{2,3,4\}\}\rangle_T = \{\{1,2,3\},\{2,3\},\{2,3,4\}\}$. Hence, $center(S_T) = \{2,3\}$, so party P sets $b_{\mathsf{it}} := \{2,3\}$. Similarly, let us compute $S_G = \langle\{5,4\}\rangle_G \cap \langle\{6,4\}\rangle_G = \{5,3,4\} \cap \{6,2,4\} = \{4\}$. Therefore, party P can not select $v_{\mathsf{it}} \in S_G \cap b_{\mathsf{it}} = \{4\} \cap \{2,3\} = \varnothing$.

It is now instructive to also identify the error in the original proof that $S_G \cap b_{\mathsf{it}} \neq \varnothing$, namely [27, Lemma 11]. The core of the proof hinges on an argument showing that there is some vertex $u \in b_{\mathsf{it}}$ such that there are at least $t+1$ pairs of parties from which P has received values at iteration it whose two values $v_{\mathsf{it}-1}$ have an induced path in G between them that passes through u. The proof of this fact is correct [27, Lemma 10]. However, it is then claimed that, because there are at most t corrupted parties, for at least one such pair of parties at least one of them is not corrupted. This is false in general: consider for simplicity $t = 10$ and parties $P_1, \ldots, P_{10}$. There are $\binom{10}{2} = 45 \geq t+1 = 11$ pairs of parties, yet 10 corruptions are enough to corrupt both parties in each pair. For this fact to be true, one would need to replace $t+1$ by $\binom{t}{2}+1$, for which [27, Lemma 10] seems unlikely to hold. On a secondary note, towards the end of the proof, two induced (or even shortest) paths in G, say one from say a to b, and one from b to c are implicitly claimed to yield an induced path from a to c that passes through b, which is not the case in general, e.g., in our graph G no induced path from 5 to 6 passes through 4.

A Corect Hybrid Protocol for AA on Chordal Graphs. In this section, we give a correct deterministic protocol achieving Monophonic Hull validity, Termination, and Agreement within Graph Distance 1 for chordal graphs. Unlike the protocol $\Pi_{\mathsf{IncorrectChordal}}$ presented in the previous section, our protocol does not directly rely on the clique tree decomposition of the chordal graph $G = (V, E)$, hence simplifying notation when it comes to convex hulls, as all convex hulls are now on G. Our protocol follows the common outline of AA protocols, proceeding in iterations. In every iteration, parties distribute their current values, and based on the values received obtain a new value for the next iteration, while ensuring that the new values "get closer" and stay within the convex hull of honest inputs. More concretely, once P obtains a set of $n - t_s + k$ value-sender pairs $\mathcal{M}$, it computes its safe area as $S := safe_{\max(k,t_a)}(\mathcal{M})$, as described for our CC protocol. To ensure that S is non-empty and included in the convex hull of the values proposed by honest parties, the underlying communication primitive needs to ensure that the sets $\mathcal{M}$ are large enough, and contain sufficient honest values. Then, once the safe area is obtained, P may compute its new value. Since G is chordal, assume $\succ$ is a perfect elimination order over the vertices of G. Namely, for $u, v \in V$ write $u \succ v$ if u comes after v in the elimination order. Then, given a set of vertices $S \subseteq V$, write $\max_{\succ} S$, or simply $\max S$, for the vertex in S that comes last in the elimination order, and define $\min S$ similarly. If S induces a clique in G, party P will pick $\max S$ as its new value. Otherwise, P will pick its new value as an arbitrary vertex in S that is not an extremal point. We will show that this update rule guarantees that agreement within distance $d = 1$ is achieved after a sufficient number of iterations, under the assumption that the safe areas obtained by all honest parties intersect. To fulfill this assumption, we make use of the protocol Π_{GTHR} of Theorem 19. This is shown with the help of the lemma below, which is proven in Appendix D.

Lemma 20. *Let $(\mathcal{M}_i)_{i=1}^K$ be sets of value-party pairs such that $k_i := |\mathcal{M}_i| - (n - t_s) \geq 0$. If $|\bigcup_{i=1}^K \mathcal{M}_i| \leq n$ and $|\bigcap_{i=1}^K \mathcal{M}_i| \geq n - t_s$ hold, then $\bigcap_{i=1}^K safe_{\max(k_i,t_a)}(\mathcal{M}_i) \neq \varnothing$.*

Our protocol Π_{Chordal} is given below:

Protocol Π_{Chordal}

Code for party P with input $v_0 \in V$

1: **for** $\mathsf{it} = 1 \ldots \mathsf{max\_it} := |V| - 1$ **do**
2: Join Π_{GTHR} with input $v_{\mathsf{it}-1}$
3: Upon obtaining output $\mathcal{M}$ in Π_{GTHR}:
4: $k := |\mathcal{M}| - (n - t_s)$; $S := safe_{\max(k,t_a)}(\mathcal{M})$.
5: If $S = ex(S)$, $v_{\mathsf{it}} := \max S$ // S induces a clique in G.

6: Otherwise, select $v_{\mathsf{it}} \in S \setminus ex(S)$ arbitrarily.
7: **end for**
8: Output $v_{\mathsf{max\_it}}$ and terminate.

We may now analyze Π_{Chordal}. We provide the result below, and we include the formal analysis in Section E of the Appendix.

Theorem 21. *Consider a chordal graph G with maximum clique size ω. Given n, t_s, t_a such that $t_s \geq t_a$ and $n > \omega \cdot t_s + t_a$, Π_{Chordal} is a (t_s, t_a)-resilient deterministic protocol achieving Monophonic Convex Validity, Termination and Agreement within Graph Distance 1.*

We use H_0 to denote the convex hull of the honest inputs, and H_{it} to denote the convex hull of the honest vertices v_{it} obtained in iteration $\mathsf{it} \geq 1$. First, the Validity condition is guaranteed, as a direct consequence of Lemma 14 along with the properties of Π_{GTHR}.

To show that Agreement within Graph Distance 1 is also guaranteed, we will show that, unless the values held by honest parties at a certain iteration already form a clique; i.e., $H_{\mathsf{it}-1}$ induces a clique; then the convex hull in the next iteration H_{it} is a strict subset of $H_{\mathsf{it}-1}$.

Lemma 22. *Assume $1 \leq \mathsf{it} \leq \mathsf{max\_it}$ and that $H_{\mathsf{it}-1}$ does not form a clique in G. Then, $H_{\mathsf{it}} \subsetneq H_{\mathsf{it}-1}$.*

Proof Sketch. Since G has $|V|$ vertices, this implies that the hull can only decrease at most $|V| - 1$ times before agreement is reached, justifying the choice for $\mathsf{max\_it}$. Hence, it remains to prove that the hull indeed strictly decreases with each iteration until agreement is reached. The proof here consists of a few ideas, including a number of helper lemmas. In essence, the first lemma says that simplicial vertices in the subgraph induced by a convex set are extremal for that set. The second lemma says that nodes that are not extremal for a convex set can not be extremal for a convex superset of it. The third is the classical result of Dirac [14] that a chordal graph has at least two simplicial vertices, provided it is not a clique.

To show that the honest hull decreases, write $s = \min H_{\mathsf{it}-1}$ and consider a node y that appears in the safe area of all honest parties. Note that such a node is guaranteed to exist by the properties of Π_{GTHR}. Note that s is simplicial in the subgraph induced by $H_{\mathsf{it}-1}$, from which it is extreme for $H_{\mathsf{it}-1}$, so if no party selects it as their next value, that would decrease the hull. If the common value satisfies $y \neq s$, then all honest parties whose safe area is a clique do not pick s since $y \succ s$, and all of those for which it is not a clique will have s as an extreme point since it is an extreme point of the honest hull as a whole, and hence will also not pick it. The case $y = s$ is more interesting since in that case, it could be that some party only has s in their safe area, so s does not get removed on this iteration. However, if this was the case, use the assumption that the honest hull is not a clique together with the result of Dirac to get another simplicial vertex $a \in H_{\mathsf{it}-1}$. It can be shown that this vertex can be chosen so that the edge $s - a$ is not in the graph. This time, instead of s, node a will be guaranteed to be removed from the hull. This is because any party who has a in their safe area also has s, so also has a length-two path from a to s in their safe area, meaning the safe area is not a clique, from which similarly to the above it follows that a will not be chosen because it is simplicial and hence extremal. □

Impossibility of Deterministic Graph Approximate Agreement. In the previous section, we have seen that chordal graphs admit a deterministic protocol achieving monophonic path convexity, termination, and agreement within graph distance 1, as long as $n > \omega \cdot t_s + t_a$, and the number of corruptions is bounded by t_s if the network is synchronous and t_a when the network is asynchronous. One might wonder whether something similar holds for general graphs, perhaps by increasing the bound on the allowed distance between any two honest outputs from 1 to some function in the size of the graph. We hereby show that, at least for geodesic convexity, if the network can be asynchronous, this function would have to be linear in the number of vertices of

the graph even to tolerate as low as two crash failures. This is shown in the following, proven by reduction from 2-Set Agreement in Appendix E.

Theorem 23. *Assume $n \geq 3$ and that the network is asynchronous, then for any $d \geq 0$ there is a graph G_d with $\Theta(d)$ vertices and edges such that no deterministic n-party protocol resilient against two crashes satisfies Geodesic Convex-Hull Validity, Termination, and Agreement within Graph Distance d.*

6 Conclusions

We have presented multiple feasibility and impossibility results in the realm of network-agnostic agreement problems in convex spaces when Byzantine corruptions are also involved.

We have seen that for any convexity space with Helly number ω achieving convex validity, (probabilistic) termination and agreement on at most $\min(n, \omega) - 1$ values requires $n > \omega \cdot t$ in synchronous networks and $n > (\omega + 1) \cdot t$ in asynchronous ones. In the best-of-both-worlds model, we have shown that $n > \max(\omega \cdot t_s, \omega \cdot t_a + t_s, 2 \cdot t_s + t_a)$ is necessary and sufficient for achieving CC. To this end, we provided a protocol Π_{CC} achieving CC when this condition holds by making use of randomization, which can be seen to be necessary due to the celebrated result of Fischer, Lynch, and Paterson [18].

In the process, we proposed two communication primitives for the best-of-both-worlds model which we believe to be of independent interest. These are variants of ACS and GTHR, which allow each party to distribute its input so that parties obtain highly reliable and consistent views on the original inputs. These variants differ from their previous counterparts by ensuring that, if the network is synchronous, all honest parties' proposed values are included in the parties' views. These stronger definitions enabled us to provide the additional synchronous guarantees of CC in terms of resilience in the best-of-both-worlds model. With its stronger guarantees, our ACS protocol can simplify future works on network-agnostic secure Multi-Party Computation, where ACS protocols are often employed during the input-sharing part of the protocol (for instance, [9] uses a less general form of ACS).

We have also focused on deterministic variants of CC; namely on AA. Here, we identified an error in a previously known AA protocol for chordal graphs and provided a different protocol for the problem, which is additionally tailored to the best-of-both-worlds model. This protocol constitutes an application of our best-of-both-worlds variant of GTHR, and achieves AA under a stronger resilience assumption of $n > \omega \cdot t_s + t_a$, where ω denotes the size of the largest clique in the input graph (equaling the Helly number). While we could only prove this resilience bound to be point-wise tight, this is indeed a natural extension of the requirements of previous AA protocols and an outstanding open problem, directly related to the corresponding question for $\mathbb{R}^D$ left open in [20]. Finally, we considered relaxing the AA agreement condition for graphs from "outputs should be at most distance 1 apart" to "outputs should be at most distance d apart." Sadly, even for d a constant fraction of the number of nodes, we found that there exist simple graphs where no deterministic asynchronous AA protocol can be resilient even to at most two parties crashing.

References

[1] Ittai Abraham, Yonatan Amit, and Danny Dolev. Optimal resilience asynchronous approximate agreement. In Teruo Higashino, editor, *Principles of Distributed Systems*, pages 229–239, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[2] Ittai Abraham, Philipp Jovanovic, Mary Maller, Sarah Meiklejohn, Gilad Stern, and Alin Tomescu. Reaching consensus for asynchronous distributed key generation. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, PODC'21, page 363–373, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3465084.3467914.

[3] Dan Alistarh, Faith Ellen, and Joel Rybicki. Wait-free approximate agreement on graphs. In Tomasz Jurdziński and Stefan Schmid, editors, *Structural Information and Communication Complexity*, pages 87–105, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-79527-6_6.

[4] Ananya Appan, Anirudh Chandramouli, and Ashish Choudhury. Perfectly-secure synchronous mpc with asynchronous fallback guarantees. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing*, PODC'22, page 92–102, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519270.3538417.

[5] Michael Ben-Or, Ran Canetti, and Oded Goldreich. Asynchronous secure computation. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 52–61, 1993.

[6] Michael Ben-Or, Boaz Kelmer, and Tal Rabin. Asynchronous secure computations with optimal resilience. In *Proceedings of the thirteenth annual ACM symposium on Principles of distributed computing*, pages 183–192, 1994.

[7] Erica Blum, Jonathan Katz, and Julian Loss. Synchronous consensus with optimal asynchronous fallback guarantees. In *Theory of Cryptography Conference*, pages 131–150. Springer, 2019.

[8] Erica Blum, Jonathan Katz, and Julian Loss. Tardigrade: An atomic broadcast protocol for arbitrary network conditions. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part II*, volume 13091 of *LNCS*, pages 547–572. Springer, Heidelberg, December 2021. doi:10.1007/978-3-030-92075-3_19.

[9] Erica Blum, Chen-Da Liu Zhang, and Julian Loss. Always have a backup plan: Fully secure synchronous mpc with asynchronous fallback. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, volume 12171 of *LNCS*, pages 707–731. Springer, 8 2020.

[10] Erica Blum, Chen-Da Liu-Zhang, and Julian Loss. Always have a backup plan: Fully secure synchronous mpc with asynchronous fallback. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 707–731, Cham, 2020. Springer International Publishing.

[11] Gabriel Bracha. Asynchronous byzantine agreement protocols. *Information and Computation*, 75(2):130–143, 1987.

[12] Ran Canetti and Tal Rabin. Fast asynchronous byzantine agreement with optimal resilience. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, page 42–51, New York, NY, USA, 1993. Association for Computing Machinery. doi:10.1145/167088.167105.

[13] Giovanni Deligios, Martin Hirt, and Chen-Da Liu-Zhang. Round-efficient byzantine agreement and multi-party computation with asynchronous fallback. In *Theory of Cryptography Conference*, pages 623–653. Springer, 2021.

[14] Gabriel Andrew Dirac. On rigid circuit graphs. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 25:71–76, 1961.

[15] Danny Dolev, Nancy A. Lynch, Shlomit S. Pinter, Eugene W. Stark, and William E. Weihl. Reaching approximate agreement in the presence of faults. *J. ACM*, 33(3):499–516, May 1986. doi:10.1145/5925.5931.

[16] Danny Dolev and H. Raymond Strong. Authenticated algorithms for byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983.

[17] Martin Farber and Robert E. Jamison. Convexity in graphs and hypergraphs. *SIAM Journal on Algebraic Discrete Methods*, 7(3):433–444, 1986. doi:10.1137/0607049.

[18] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

[19] Diana Ghinea, Chen-Da Liu-Zhang, and Roger Wattenhofer. Optimal synchronous approximate agreement with asynchronous fallback. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing*, PODC'22, page 70–80, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519270.3538442.

[20] Diana Ghinea, Chen-Da Liu-Zhang, and Roger Wattenhofer. Multidimensional Approximate Agreement with Asynchronous Fallback. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), Orlando, Florida, USA*, June 2023.

[21] Maurice Herlihy, Dmitry Kozlov, and Sergio Rajsbaum. *Distributed Computing Through Combinatorial Topology*. Morgan Kaufmann, Boston, 2014.

[22] Jérémy Ledent. Brief announcement: Variants of approximate agreement on graphs and simplicial complexes. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, PODC'21, page 427–430, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3465084.3467946.

[23] Hammurabi Mendes and Maurice Herlihy. Multidimensional approximate agreement in byzantine asynchronous systems. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 391–400. ACM Press, June 2013. doi:10.1145/2488608.2488657.

[24] Hammurabi Mendes, Maurice Herlihy, Nitin Vaidya, and Vijay K Garg. Multidimensional agreement in byzantine systems. *Distributed Computing*, 28(6):423–441, 2015.

[25] Atsuki Momose and Ling Ren. Multi-threshold byzantine fault tolerance. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 1686–1699, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3460120.3484554.

[26] Achour Mostéfaoui, Hamouma Moumen, and Michel Raynal. Signature-free asynchronous byzantine consensus with t¡ n/3 and o (n2) messages. In *Proceedings of the 2014 ACM symposium on Principles of distributed computing*, pages 2–9, 2014.

[27] Thomas Nowak and Joel Rybicki. Byzantine Approximate Agreement on Graphs. In Jukka Suomela, editor, *33rd International Symposium on Distributed Computing (DISC*

2019), volume 146 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:17, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2019/11336`, `doi:10.4230/LIPIcs.DISC.2019.29`.

[28] Gerard Sierksma. Caratheodory and helly-numbers of convex-product-structures. *Pacific Journal of Mathematics*, 61:275–282, 1975.

[29] Nitin H. Vaidya and Vijay K. Garg. Byzantine vector consensus in complete graphs. In Panagiota Fatourou and Gadi Taubenfeld, editors, *32nd ACM PODC*, pages 65–73. ACM, July 2013. `doi:10.1145/2484239.2484256`.

A Preliminary Proofs

Lemma 1. *Let $G = (V, E)$ be a graph and $S \subseteq V$ be a subset of its vertices. Then, under both geodesic or monophonic convexity S is a free set if and only if S induces a clique in G.*

Proof. First, note that for all S inducing a clique, $S = \langle S \rangle$, as all vertices in S are linked by edges. Moreover, if S induces a clique, then $S = ex(S)$ because for any $s \in S$ it holds that $S \setminus \{s\}$ induces a clique and hence $s \notin S \setminus \{s\} = \langle S \setminus \{s\} \rangle$. This proves the "if" direction. For the "only if" direction, assume S does not induce a clique and let $a, b \in S$ be two vertices not joined by an edge. Consider a shortest/induced path P from a to b in G. Since a and b are not adjacent, P consists of at least three nodes, so take $v \notin \{a, b\}$ to be on path P. By definition, this means that $v \in \langle S \rangle$ Moreover, $v \in \langle S \setminus \{v\} \rangle$ because of path P, from which $v \notin ex(S)$. Therefore, $v \in \langle S \rangle \setminus ex(S)$, so by definition S is not free. □

B Impossibility Results Using Adversarial Families

Lemma 3. *Let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be an m-adversarial family for convexity space $\mathcal{C}$. Assume $n \geq m$ and that, moreover, $n \leq m \cdot t$ if the network is synchronous and $n \leq (m+1) \cdot t$ if the network is asynchronous. Then, any (deterministic or randomized) n-party protocol satisfying Convex-Hull Validity and (Probabilistic) Termination will have a terminating execution where there are honest parties $P_1, \ldots, P_m$ such that the output v^i_{out} of party P_i satisfies $v^i_{out} \in A_i$.*

Proof. Write $A = \cup \mathcal{A}$ and consider a protocol Π satisfying convex-hull validity and termination. Partition the n parties into m groups $G_1, G_2, \ldots, G_m$ such that $1 \leq |G_i| \leq t$ for all i and consider an instance of Π where each party in G_i has as input some arbitrary value $a_i \in A_i$. Consider $m+1$ scenarios. In scenario $s \in [m]$, the adversary corrupts precisely the parties in G_s, while in scenario $m+1$, the adversary corrupts no parties. In all scenarios, the adversary ensures no corrupted parties ever deviate from the protocol and does not manipulate the scheduler. By construction, observe that any execution of the protocol that is consistent with any of the scenarios is consistent with all scenarios. Since Π satisfies termination, consider an arbitrary execution E of the protocol consistent with scenario $m+1$. Note that this implies that all parties obtain outputs and that execution E is consistent with the other scenarios as well. For execution E, consider an arbitrary i and a party $P \in G_i$ whose output is v^P_{out}. We will show that $v^P_{out} \in A_i$. Assume otherwise, then, since $A_i = \cap_{\ell \neq i} \langle A \setminus A_\ell \rangle$, there exists $k \neq i$ such that $v^P_{out} \notin \langle A \setminus A_k \rangle$. In scenario k the set of corrupted parties is G_k, so the convex hull of the honest inputs is a subset of $\langle A \setminus A_k \rangle$. In this scenario party P is not corrupted and has output $v^P_{out} \notin \langle A \setminus A_k \rangle$, contradicting convex-hull validity. Therefore, we get that $v^P_{out} \in A_i$ as claimed, and hence the conclusion.

For the asynchronous case, the proof is similar in spirit. This time, partition the n parties into $m+1$ groups $G_1, G_2, \ldots, G_{m+1}$ such that $1 \leq |G_i| \leq t$ for $i \in [m]$ and $0 \leq |G_{m+1}| \leq t$. Consider an instance of Π where each party in G_i has input some arbitrary value $a_i \in A_i$, except parties in G_{m+1}, which can have arbitrary inputs. Consider again $m+1$ scenarios. In scenario $s \in [m+1]$, the adversary corrupts precisely the parties in G_s. For $s = m+1$, the adversary makes the corrupted parties crash immediately and does not manipulate the scheduler. For $s \in [m]$, the adversary ensures no corrupted party deviates from the protocol, but this time delays messages sent from parties in G_{m+1} until all other parties have obtained outputs. Note that this could lead to messages getting delayed indefinitely if some honest party does not obtain output (e.g., if the protocol is randomized), which would not be within the power of the adversary, but this will not be the case for the executions we consider. Because Π satisfies termination, consider an arbitrary execution E of the protocol consistent with scenario $m+1$ in which all honest parties obtain outputs. Note that any such execution is also consistent with the other scenarios. The rest of the proof is analogous, showing with the same argument that for execution E we have that $v^P_{out} \in A_i$ for any $P \in G_i$, where $i \in [m]$. □

Lemma 4. *Assume a convexity space $\mathcal{C}$ admitting a 2-adversarial family $\mathcal{A} = \{A_1, A_2\}$. Assume $2 \leq n \leq 2 \cdot t_s + t_a$. Let Π denote an arbitrary (deterministic or randomized) protocol achieving Convex-Hull Validity and (Probabilistic) Termination for at most t_s corruptions when the network is synchronous and at most t_a corruptions when it is asynchronous. Then, Π has a terminating execution where the outputs v^1_{out} and v^2_{out} of two honest parties satisfy $v^1_{out} \in A_1$ and $v^2_{out} \in A_2$.*

Proof. Write $A = \bigcup \mathcal{A}$. We partition the n parties into three groups G_1, G_2 and G_a, such that $1 \leq |G_1|, |G_2| \leq t_s$ and $0 \leq |G_a| \leq t_a$. For $1 \leq i \leq 2$, assume that each party in G_i has as input some arbitrary value $a_i \in A_i$. We consider three scenarios.

In the first scenario, we assume that the network is synchronous, hence at most t_s parties may be corrupted. The adversary therefore corrupts the parties in G_2, causing them to not send any messages. The parties in G_1 and G_a are honest and hold as input some arbitrary value $a_1 \in A_1$. Then, Convex-Hull Validity ensures that, in any terminating execution consistent with the scenario, parties in G_1 obtain outputs in A_1.

Similarly, in the second scenario, we assume that the network is synchronous, but this time the adversary corrupts the parties in G_1, causing them to not send any messages. The parties in G_2 and G_a are honest and hold as input some arbitrary value $a_2 \in A_2$. Then, Convex-Hull Validity ensures that, in any terminating execution consistent with the scenario, parties in G_2 obtain outputs in A_2.

In the third scenario, we assume that the network is asynchronous, hence at most t_a parties may be corrupted. The adversary therefore corrupts the parties in G_a. Intuitively, the adversary will make use of the parties in G_a and of the message delivery scheduler to cause honest parties' views to be indistinguishable from their views in the previous two scenarios. We assume that the honest parties' clocks are still synchronized; however, the adversarial scheduler will block the communication between the two groups of honest parties G_1 and G_2. The messages sent within $G_1 \cup G_a$ or within $G_2 \cup G_a$ will be delivered with delay at most Δ, as if the network were synchronous. Then, we make a virtual copy of each party in G_a, obtaining two virtual sets of corrupted parties: G^1_a and G^2_a. The virtual copies in G^1_a run Π correctly with the same inputs $a_1 \in A_1$ as in the first scenario towards the parties in G_1. Similarly, the virtual copies in G^2_a run Π correctly with the same inputs $a_2 \in A_2$ as in the second scenario towards the parties in G_2. This ensures that parties in G_1 and G_2 have the same view as in the first and second scenario respectively. Since Π achieves Termination, there is a terminating execution consistent with this scenario, hence also consistent with the first two scenarios. Then, as argued previously, in any such execution, parties G_1 and G_2 obtain outputs $v^1_{\mathsf{out}} \in A_1$ and resp. $v^2_{\mathsf{out}} \in A_2$, completing the proof. □

Lemma 5. *Let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be an m-adversarial family for convexity space $\mathcal{C}$. Assume that $m \leq n \leq m \cdot t_a + t_s$. Then, any (deterministic or randomized) n-party protocol satisfying Convex-Hull Validity and (Probabilistic) Termination for at most t_s corruptions when the network is synchronous and at most t_a corruptions when the network is asynchronous will have a terminating execution where there are honest parties $P_1, \ldots, P_m$ such that the output v^i_{out} of party P_i satisfies $v^i_{out} \in A_i$.*

Proof. Consider a protocol Π satisfying convex-hull validity and termination. Partition the n parties into $m + 1$ groups $G_1, G_2, \ldots, G_{m+1}$ such that $1 \leq |G_i| \leq t_a$ for all $1 \leq i \leq m$, and $0 \leq \left|G_{m+1}\right| \leq t_s$. The rest of the proof is identical to the proof for the asynchronous setting in Lemma 3. □

Theorem 9. *Consider a convexity space $\mathcal{C}$ with Helly number $\omega \geq 2$. Assume $2 \leq n \leq 2 \cdot t_s + t_a$ or $2 \leq n \leq \omega \cdot t_a + t_s$. Then, there is no (deterministic or randomized) n-party protocol satisfying Convex-Hull Validity, (Probabilistic) Termination and Exact Agreement can simultaneously tolerate at most t_s corruptions when the network is synchronous and at most t_a corruptions*

when the network is asynchronous. For the case $2 \leq n \leq \omega \cdot t_a + t_s$*, the same holds even for the weaker condition of agreeing on at most* $\min(n, \omega) - 1$ *values.*

Proof. For the case $2 \leq n \leq 2 \cdot t_s + t_a$, by Lemma 6, there is an ω-adversarial family for $\mathcal{C}$. Since $2 \leq \omega$, using Lemma 7, let $\mathcal{A} = \{A_1, A_2\}$ be a 2-adversarial family for $\mathcal{C}$. Consider a protocol Π satisfying convex-hull validity and termination. By Lemma 4, there is a terminating execution of Π where the set of honest outputs contains $\{a_1, a_2\}$ where $a_1 \in A_1$ and $a_2 \in A_2$. Since A_1 and A_2 are disjoint, $a_1 \neq a_2$, from which the conclusion follows.

For the case $2 \leq n \leq \omega \cdot t_a + t_s$, write $m = \min(n, \omega)$. Similarly, by Lemma 6, there is an ω-adversarial family for $\mathcal{C}$. Since $m \leq \omega$, using Lemma 7, let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be an m-adversarial family for $\mathcal{C}$. Consider a protocol Π satisfying convex-hull validity and termination. By Lemma 5, there is a terminating execution of Π where the set of honest outputs contains $\{a_1, \ldots, a_m\}$ where $a_i \in A_i$. As sets in $\mathcal{A}$ are pairwise disjoint, this set has cardinality m, implying the conclusion. □

Theorem 10. *Consider* $\mathbb{R}^D$ *with straight-line convexity and let* $d > 0$ *be arbitrary. Assume* $n \leq (D+1) \cdot t$ *if the network is synchronous and* $n \leq (D+2) \cdot t$ *if the network is asynchronous. Then, there is no (deterministic or randomized) n-party protocol satisfying Convex-Hull Validity and Termination such that no two honest outputs are more than Euclidean distance d apart.*

Proof. It suffices to consider the case $n \geq D + 1$, as otherwise the inputs would be contained in an $(n-1)$-dimensional subspace of $\mathbb{R}^D$, which is equivalent to assuming they are points in $\mathbb{R}^{n-1}$, so the result could then be invoked for $\mathbb{R}^{n-1}$ with $n \geq (n-1) + 1$. Consider the origin point 0 of $\mathbb{R}^D$, as well as the unit vectors $e_1, \ldots, e_D$, and define the family of disjoint convex sets $\mathcal{A} = \{A_0, \ldots, A_D\}$, where $A_0 = \{0\}$ and $A_i = \{(2d)e_i\}$ for $i \in [D]$. One can check that $\mathcal{A}$ is a $(D+1)$-adversarial family because the intersection of any D faces of a D-simplex is the point common to all of them. Hence, by Lemma 3 any protocol Π satisfying convex-hull validity and termination has a terminating execution where $\{0, (2d)e_1, \ldots, (2d)e_D\}$ is a subset of the honest outputs. The distance between 0 and $(2d)e_1$ is $2d > d$, implying the conclusion. □

B.1 Comparison with [27, Theorems 10 and 11]

In this section, we compare our impossibility results with the related [27, Theorems 10 and 11]. We find that our results generalize the aforementioned, with the exception of the first part of [27, Theorems 11], to which our findings are orthogonal. However, we exhibit what we believe to be an error in the original proof of this part, rendering the result false in general.

Theorem 24 ([27, Theorem 10])**.** *Let* $\mathcal{C}$ *be a convex geometry with Helly number* ω*. If the network is synchronous and* $n \leq \omega t$*, then no n-party protocol satisfies convex-hull validity, termination and exact agreement.*

Contrasting this with Theorem 8, for the synchronous case our results generalize the previous by removing the strong requirement on $\mathcal{C}$ to be a convex geometry and by adding the fact that even agreement on at most $\min(n, \omega) - 1$ values is not possible. Next, for use in the following, call a (not necessarily convex) subset $\mathcal{I} \subseteq V$ *irredundant* if there is a point $p \in \langle I \rangle$ such that the hull of no proper subset of $\mathcal{I}$ contains p. The Carathéodory number c of $\mathcal{C}$ is then the size of the largest such irredundant set $\mathcal{I}$.

Theorem 25 ([27, Theorem 11])**.** *Let* $\mathcal{C}$ *be a convexity space with Helly number* ω *and Carathéodory number c. Assume the network is asynchronous and consider a protocol satisfying convex-hull validity and termination, then:*

1. *If* $n \leq (c+1)t$ *there is an execution where the honest outputs do not form a free set in* $\mathcal{C}$*.*

2. *If $n \leq (\omega+1)t$ and $\mathcal{C}$ is a convex geometry there is an execution where the set of honest outputs either has size at least ω or is not a free set in $\mathcal{C}$.*

Contrasting with Theorem 8, for the asynchronous case our results generalize Part 2 of the above by once again removing the requirement on $\mathcal{C}$ to be a convex geometry and also by no longer requiring the clause "or is not a free set in $\mathcal{C}$." Our result also replaces ω by $\min(n, \omega)$, which we believe is also implicitly meant in the original result, as when $n < \omega$ the condition becomes vacuous, and a protocol where parties just output their own inputs satisfies convex-hull validity and termination in some convex geometries.

Part 1 of Theorem 25, on the other hand, is orthogonal to our results. In our attempt to use adversarial families to potentially also recover Part 1, we have discovered what we believe to be an error in the proof of this part, making the result false in general. Namely, the proof of Part 1 hinges on the following technical lemma:

Lemma 26 ([27, Lemma 15])**.** *Let $\mathcal{C}$ be a convexity space and A be an irredundant set such that $|A| > 1$. Then for any $a \in A$ and $y \in \langle A \rangle \setminus A$ there exists $b \in A \setminus \{a\}$ such that $y \notin \langle A \setminus \{b\} \rangle$.*

Note that we have added the condition "A is irredundant" missing from the original statement.[7] The error in the proof is towards the end where, using the original notation, it is stated that $y \notin \partial A = \langle A \rangle \setminus B \subseteq \langle A \rangle \setminus A$ implies that $y \notin \langle A \rangle \setminus A$, contradicting the hypothesis. However, in general, if some sets satisfy $S_1 \subseteq S_2$ and $y \notin S_1$ it does not follow that $y \notin S_2$. We next construct a convexity space where the lemma in fact fails for all irredundant sets A and all $a \in A$. First, introduce some auxiliary notation: given two convexity spaces $\mathcal{C}_1$ and $\mathcal{C}_2$ defined on universes V_1 and V_2 respectively, define $\mathcal{C}_1 \oplus \mathcal{C}_2$ to be the convexity space on universe $V_1 \times V_2$ such that $\mathcal{C}_1 \oplus \mathcal{C}_2 = \{C_1 \times C_2 \mid C_1 \in \mathcal{C}_1, C_2 \in \mathcal{C}_2\}$. For the construction, start with an arbitrary convexity space $\mathcal{C}$ on universe V and consider the convexity space $\mathcal{C}' = \mathcal{C} \oplus \{\varnothing, \{0,1\}\}$. To build intuition for $\mathcal{C}'$, notice that $\langle \{(v,i)\} \rangle = \{(v,0),(v,1)\}$ for any $v \in V$ and $i \in \{0,1\}$. Assume A is an irredundant set for $\mathcal{C}'$. Note that for no $v \in V$ does A contain both points $(v,0)$ and $(v,1)$, as otherwise it would be that $\langle A \rangle = \langle A \setminus \{(v,1)\} \rangle$, so A would not be irredundant. Consider any $a = (v,i) \in A$ and take $y = (v, 1-i) \in \langle A \rangle \setminus A$, then for any $b \in A \setminus \{a\}$ it holds that $a \in A \setminus \{b\}$, from which $\langle \{a\} \rangle = \{(v,0),(v,1)\} \subseteq \langle A \setminus \{b\} \rangle$, so $y \in \langle A \setminus \{b\} \rangle$, contradicting the statement of the lemma. Hence, we have constructed a space for which the lemma fails for any irredundant set A and any $a \in A$, indicating that any correct weakening of the lemma might sadly not be of much use in its current form.

We conclude by constructing a space whose Carathéodory number c is much larger than its Helly number ω, showing that our possibility results are not consistent with Part 2 of Theorem 25. To do so, we will use the fact [28, Theorems 2.1 and 3.2] that given convexity spaces $\mathcal{C}_1$ and $\mathcal{C}_2$ with Helly numbers ω_1, ω_2 and Carathéodory numbers c_1, c_2 the space $\mathcal{C}_1 \oplus \mathcal{C}_2$ has Helly number $\omega = \max\{\omega_1, \omega_2\}$ and Carathéodory number c satisfying $c_1 + c_2 - 2 \leq c \leq c_1 + c_2$. Consider the space $\mathcal{C} = \mathbb{R}^2$ with straight-line convexity, whose Helly and Carathéodory numbers are both 3. Then, the space $\mathcal{C}_k = \bigoplus_{\ell=1}^{k} \mathcal{C}$ has Helly number $\omega_k = 3$ and Carathéodory number $c_k \geq 3k - 2(k-1) = k + 2$. For this space, our possibility results imply that, for instance when the network is synchronous, convex consensus be solved assuming $n > 2t$, while Part 2 of Theorem 25 would imply that it can not be solved for $n \leq (k+2)t$.

C Communication Primitives

In this section, we include the formal definitions of the communication primitives used in our algorithms, along with formal proofs.

[7] The proof of the lemma notes that if A is not irredundant the claim becomes vacuous, however, one can actually consider $\mathbb{R}^2$ with straight-line convexity, $A = \{(\pm 1, \pm 1)\}$ and $y = (0.5, 0.5)$, in which case $y \in \langle A \setminus \{a\} \rangle$ for any $a \in A$. This issue is however only minor since the lemma is only invoked in the proof of the subsequent [27, Lemma 16], where A is assumed to be irredundant.

C.1 Reliable Broadcast

We recall the formal definition of rBC. We use the definition of [19], which makes the termination time explicit when the network is synchronous.

Definition 27. *Let Π be a protocol where a designated party S (called the sender) holds a value v_S, and every party P may output a value v_P.*

- *Validity: If S is honest, and an honest party outputs v_P, then $v_P = v_S$.*
- *Consistency: If P and P' are honest and output v_P and resp. $v_{P'}$, then $v_P = v_{P'}$.*
- *c-Honest Termination: If S is honest, parties obtain outputs eventually. In addition, if the network is synchronous and the parties start executing the protocol at the same time τ, every honest party obtains output by time $\tau + c \cdot \Delta$.*
- *c'-Conditional Termination: If an honest party P obtains output at time τ, then all honest parties obtain outputs eventually. In addition, if the network is synchronous and the honest parties start executing the protocol at the same time, then all honest parties obtain output by time $\tau + c' \cdot \Delta$.*

We say that Π is a (t_s, t_a, c, c')-resilient Reliable Broadcast protocol if it achieves Validity, Consistency, c-Honest Termination, and c'-Conditional Termination even when t_s of the parties involved are corrupted if it runs in a synchronous network, and even when t_a of the parties involved are corrupted otherwise.

Our protocols will make use of two rBC protocols. The first one is Bracha's protocol [11], which does not assume a public key infrastructure. The theorem below follows from the analysis of [20].

Theorem 28 (Bracha [11])**.** *Assume that $n > 3t$. Then, there is an n-party protocol achieving $(t, t, c_{rBC}, c'_{rBC})$-resilient Reliable Broadcast, where $c_{rBC} := 3$ and $c'_{rBC} := 2$.*

The second protocol is that of Momose and Ren [25]. The theorem below follows from the analysis of [19].

Theorem 29 (Momose and Ren [25])**.** *Assume that $n > 2 \cdot t_s + t_a$ and $t_s \geq t_a$. Then, there is an n-party protocol achieving $(t_s, t_a, c_{rBC}, c'_{rBC})$-resilient Reliable Broadcast (assuming PKI), where $c_{rBC} := 3$ and $c'_{rBC} := 1$.*

C.2 Agreement on a Core-Set When $t_s, t_a < n/3$

We present a protocol achieving our best-of-both-worlds definition of ACS that assumes $t_s, t_a < n/3$. We note that this protocol does not require a public key infrastructure.

As previously mentioned, to achieve this definition of ACS, we follow the outline of [6]. In this protocol, parties first distribute their values via an rBC protocol Π_{rBC}. In this case, Bracha's protocol, noted in Theorem 28, is sufficient. A second building block that we assume is the asynchronous (BA) protocol Π_{BA} of Mostefaoui et al [26], noted in the theorem below.

Theorem 30. *Given $t_s, t_a \leq n/3$, there is a (t_s, t_a)-resilient BA protocol Π_{BA} that does not assume PKI.*

In the asynchronous ACS protocol of [6], n parallel invocations of Π_{BA} are initiated – one for each party. When a party P receives a value v from P' via Π_{rBC}, it joins the Π_{BA} invocation that corresponds to P' with input 1. Once $n - t_a$ Π_{BA} invocations have resulted in output 1 (and therefore $n - t_a$ values are *worth waiting for*), P may join any remaining invocations with input 0. All the Π_{BA} invocations are guaranteed to terminate eventually, and once this is the case, P outputs the set of value-sender pairs corresponding to the Π_{BA} invocations terminating

in 1. This way, the protocol ensures that all honest parties obtain an identical view, regardless of the type of network.

The first modification that we make is to only wait for $n - t_s$ (instead of $n - t_a$) Π_{BA} invocations to terminate with output 1 before allowing parties to join the remaining invocations with input 0. This is needed as, if the network is synchronous, $n - t_s \leq n - t_a$ and therefore waiting for $n - t_a$ values may not be possible.

We still need to ensure that, if the network is synchronous, all honest parties are included in the output sets. We make use of a few non-standard properties of Π_{rBC}, proven in [20]. Namely, in the case of Bracha's protocol, if the network is synchronous and the sender is honest, all parties obtain output within $c_{\mathsf{rBC}} \cdot \Delta$ time for $c_{\mathsf{rBC}} = 3$. Therefore, to ensure that all honest values are included in the output set agreed upon, we force parties to wait until $c_{\mathsf{rBC}} \cdot \Delta$ time passed before joining any Π_{BA} invocation with input 0 (which does not change anything if the network is asynchronous).

We present the formal code of the protocol below.

Protocol Π_{ACS}

Code for party P with input v

1: $\tau_{\mathtt{start}} := \tau_{\mathtt{now}}$
2: Send v to every party via Π_{rBC}.
3: When receiving a value v from P' via Π_{rBC}:
4: If less than $n - t_s$ invocations of Π_{rBC} have terminated, or $\tau_{\mathtt{now}} \leq c_{\mathsf{rBC}} \cdot \Delta$:
5: Join the invocation of Π_{BA} for P' with input 1.
6: When $\tau_{\mathtt{now}} > c_{\mathsf{rBC}} \cdot \Delta$ and at least $n - t_s$ of the Π_{BA} invocations have terminated with input 1, join the remaining Π_{BA} invocations with input 0.
7: When all Π_{BA} invocations have terminated:
8: $\mathcal{P}$:= parties whose corresponding Π_{BA} invocations have terminated with output 1.
9: When all invocations of Π_{rBC} having senders in $\mathcal{P}$ have terminated:
10: $\mathcal{M}$:= the set of pairs (v', P'), where $P' \in \mathcal{P}$ and v' is the value P' sent via Π_{rBC}.
11: Output $\mathcal{M}$ and terminate.

We may now prove the following result.

Theorem 31. *There is a (t_s, t_a)-resilient ACS protocol for $t_s, t_a < n/3$.*

We separate the proof into the analysis of Π_{ACS} in the synchronous setting only, and then in the asynchronous setting only.

Lemma 32. *When running in a synchronous network where at most t_s of the parties involved are corrupted, Π_{ACS} achieves Validity, Consistency, Exact Agreement, t_s-Output Size, and Probabilistic Termination, and Honest Core.*

Proof. First, Validity and Consistency follow immediately from the properties of Π_{rBC}.

Since the network is synchronous, at least the $n - t_s$ honest invocations of Π_{rBC} terminate by time $c_{\mathsf{rBC}} \cdot \Delta$. Hence, at this time, all honest parties join the n invocations of Π_{BA}. Moreover, all honest parties join the invocations corresponding to honest parties with input 1.

Π_{BA}'s Weak Validity then ensures that at least the $n - t_s$ invocations corresponding to honest parties result in output 1. In addition, if the Π_{BA} invocation for some party P results in output 1, then at least one honest party has joined this invocation with input 1, and therefore has received a value from P via Π_{rBC}. Then, c'_{rBC}-Conditional Termination ensures that all honest parties receive this value.

Hence, Π_{BA} has allowed (by to Probabilistic Termination) parties to agree on the same set (by Exact Agreement) of at least $n - t_s$ parties $\mathcal{P}$, that contains all the honest parties, and each honest party eventually receives the values sent by all parties in $\mathcal{P}$. Hence, all parties output the same set $\mathcal{M}$ of at least $n - t_s$ values. Therefore, Exact Agreement, Probabilistic Termination, and Honest Core hold. □

Lemma 33. *When running in an asynchronous network where at most t_a of the parties involved are corrupted, Π_{ACS} achieves Validity, Consistency, Exact Agreement, t_s-Output Size, and Probabilistic Termination.*

Proof. Similarly, Validity and Consistency follow immediately from Π_{rBC}'s Validity and Consistency properties.

We first show that at least $n - t_s$ invocations of BA terminate with output 1. Note that no honest party joins Π_{BA} with input 0 until $n - t_s$ of the Π_{BA} invocations have terminated with output 1. c_{rBC}-Honest termination ensures that at least the Π_{rBC} invocations having honest senders terminate. Therefore, eventually $n - t_s$ of the Π_{BA} invocations indeed terminate with output 1.

Then, similarly to the proof for the synchronous case, if the Π_{BA} invocation for some party P results in output 1, Weak Validity ensures that at least an honest party has joined this invocation with input 1, and therefore has received a value from P. Then, Π_{rBC}'s Consistency and c'_{rBC}-Conditional Termination properties ensure that all parties eventually receive the same value from P.

Hence, Π_{BA} allows the parties to agree on the same set $\mathcal{P}$ of at least $n - t_s$ parties, and eventually parties receive the same values sent by the parties in set $\mathcal{P}$ and therefore may terminate. It follows that Exact Agreement, t_s-Output Size, and Probabilistic Termination. also hold. □

C.3 Gather

We may now present our construction realizing Theorem 19.

Theorem 19. *Let n, t_s, t_a be such that $n > 2 \cdot t_s + t_a$ and $t_a \leq t_s$. Then, there is a protocol Π_{GTHR} achieving (t_s, t_a)-resilient GTHR.*

Our protocol Π_{GTHR} will make use of an underlying rBC protocol Π_{rBC}. When instantiated with the protocol of Theorem 29, this will lead exactly to the construction proving Theorem 19. Making use of the rBC protocol of Theorem 28 instead will lead to a $(t_s, t_a, 8)$-resilient GTHR protocol that does not assume any cryptographic setup.

Theorem 34. *Let $t_s, t_a < n/3$. Then, there is a (t_s, t_a)-resilient GTHR protocol Π_{GTHR} that does not assume PKI.*

Π_{GTHR} follows the outline of the initialization subroutine used in the AA protocol of [20] (Section 5). That is, it heavily relies on the *witness technique* [1]. Parties distribute their inputs via Π_{rBC}, and add any value received and its sender to a set of value-party (or value-sender) pairs $\mathcal{M}$. Whenever such an input is received from Π_{rBC}, the sender is added to a set W_0, representing *level-zero witnesses*.

When at least $c_{\mathsf{rBC}} \cdot \Delta$ time has passed (meaning that, if the network is synchronous, every honest input was received), and when $\left|\mathcal{M}\right| \geq n - t_s$ (since at most t_s parties are corrupted), the parties reliably broadcast their set of level-zero witnesses W_0. Then, if P receives a set of level-zero witnesses W'_0 from P' such that all values sent by parties in W'_0 were also received by P (roughly, $\mathcal{M}_{P'} \subseteq \mathcal{M}$), P marks P' as a *level-one witness* by adding it to its set W_1.

Following the insights of [2], we obtain that, when $n - t_s$ honest parties hold W_1 sets of size $n - t_s$, then $t_s + 1$ honest parties have a common level-one witness $P^\star$ that is also an honest party. This will then enable us to achieve the t_s-Common Core property. Concretely, when party P gathers $n - t_s$ level-one witnesses, it sends its set W_1 to all the parties. When receiving a set of W'_1 from some party P'' such that $W'_1 \subseteq W_1$, P marks P' as a *level-two witness* by adding it to its set W_2. Once P collects $n - t_s$ level two-witnesses, it may output its set $\mathcal{M}$. This ensures that P has marked at least one of the parties that have announced $P^\star$ to be one of their level-one witnesses – and therefore P has marked $P^\star$ as a level-one witness as well. Hence, P has received the set $W_0^\star$ sent by $P^\star$, and therefore all the values sent by the parties in $W_0^\star$

have been included in P's set $\mathcal{M}$. This argument applies to every honest party, which ensures that the t_s-Common Core property holds.

We include the formal code of Π_{GTHR} below. Note that, in the protocol's code and proofs, we assume that $c_{\mathsf{rBC}} \geq c'_{\mathsf{rBC}} \geq 1$.

Protocol Π_{GTHR}

Code for party P with input v

1: $\tau_{\mathtt{start}} := \tau_{\mathtt{now}}$; $\mathcal{M} := \varnothing$; $W_0, W_1, W_2 := \varnothing$.
2: Send v to every party via Π_{rBC}.
3: Whenever receiving a value v' from P' via Π_{rBC}, add (v', P') to $\mathcal{M}$ and P' to W_0.
4: If $\tau_{\mathtt{now}} \geq \tau_{\mathtt{start}} + c_{\mathsf{rBC}} \cdot \Delta$ and $\left|W_0\right| \geq n - t_s$:
5: Send W_0 to all parties via Π_{rBC}.
6: Whenever receiving W'_0 from P' via Π_{rBC} such that $\left|W'_0\right| \geq n - t_s$:
7: When $W'_0 \subseteq W_0$, add P' to W_1.
8: When $\tau_{\mathtt{now}} \geq 2 \cdot c_{\mathsf{rBC}} \cdot \Delta$ and $\left|W_1\right| \geq n - t_s$:
9: Send W_1 to all parties.
10: Whenever receiving W'_1 from P' such that $\left|W'_1\right| \geq n - t_s$:
11: When $W'_1 \subseteq W_1$, add P' to W_2.
12: When $\tau_{\mathtt{now}} \geq (2 \cdot c_{\mathsf{rBC}} + c'_{\mathsf{rBC}}) \cdot \Delta$ and $\left|W_2\right| \geq n - t_s$:
13: Output $\mathcal{M}$.

We now proceed to analyze Π_{GTHR}, first assuming that the network is synchronous, and then that the network is asynchronous. Theorem 19 follows immediately from Lemmas and 43 below.

Synchronous Network. In the following, we assume that the network is synchronous, all parties join the protocol at the same time $\tau_{\mathtt{start}}$, and at most t_s of the parties involved are corrupted.

Lemma 35. *Let P be an honest party. By time $\tau_{\mathtt{start}} + c_{rBC} \cdot \Delta$, P holds a set W_0 containing all honest parties.*

Proof. Follows immediately from Π_{rBC}'s c_{rBC}-Honest Termination. □

Lemma 36. *Let P and P' denote two honest parties. By time $\tau_{\mathtt{start}} + 2c_{rBC} \cdot \Delta$, P has added P' to its set W_1.*

Proof. According to Lemma 35, at time $\tau_{\mathtt{start}} + c_{\mathsf{rBC}} \cdot \Delta$, P' has sent its set W'_0 to all parties via Π_{rBC}. Hence, P receives W'_0 by time $\tau_{\mathtt{start}} + 2 \cdot c_{\mathsf{rBC}} \cdot \Delta$ due to Validity and c_{rBC}-Honest Termination.

The set W'_0 set contains at least $n - t_s$ parties from whom P' has received values via Π_{rBC} by time $\tau_{\mathtt{start}} + c_{\mathsf{rBC}} \cdot \Delta$. Then, the Consistency and c'_{rBC}-Conditional Termination properties ensure that P has received these values as well by time $\tau_{\mathtt{start}} + (c_{\mathsf{rBC}} + c'_{\mathsf{rBC}}) \cdot \Delta \leq \tau_{\mathtt{start}} + 2c_{\mathsf{rBC}} \cdot \Delta$ (since $c'_{\mathsf{rBC}} \leq c_{\mathsf{rBC}}$). This implies that, when P receives the set W'_0 from P', the condition $W'_0 \subseteq W_0$ holds. Therefore, P adds P' to its set W_1 at time $\tau_{\mathtt{start}} + 2c_{\mathsf{rBC}} \cdot \Delta$. □

Lemma 37. *Let P and P' denote two honest parties. By time $\tau_{\mathtt{start}} + (2c_{rBC} + c'_{rBC}) \cdot \Delta$, P has added P' to its set W_2.*

Proof. Lemma 36 ensures that, at time $\tau_{\mathtt{start}} + 2 \cdot c_{\mathsf{rBC}} \cdot \Delta$, P' holds a set W'_1 of size at least $n - t_s$, and therefore sends it to all the parties. These messages are received within one communication round, and, since $c'_{\mathsf{rBC}} \geq 1$, it follows that P has received this set at time $\tau_{\mathtt{start}} + (2c_{\mathsf{rBC}} + 1) \cdot \Delta \leq \tau_{\mathtt{start}} + (2c_{\mathsf{rBC}} + c'_{\mathsf{rBC}}) \cdot \Delta$.

Note that, if P' has added a party P'' in its set W'_1 by time $\tau_{\mathtt{start}} + 2 \cdot c_{\mathsf{rBC}} \cdot \Delta$, P receives all the necessary messages to add P'' to its set W_1 as well. Moreover, these messages are received within $c'_{\mathsf{rBC}} \cdot \Delta$ additional time, due to c'_{rBC}-Conditional Termination and Consistency. Therefore, by time $\tau_{\mathtt{start}} + (2c_{\mathsf{rBC}} + c'_{\mathsf{rBC}}) \cdot \Delta$, $W'_1 \subseteq W_1$ holds and hence P adds P' to W_2. □

Lemma 38. Π_{GTHR} *satisfies Synchronized Termination Honest Common Core, Validity, and Consistency when running in a synchronous network where at most* t_s *of the parties involved are corrupted.*

Proof. Validity and Consistency follow immediately from the properties of Π_{rBC}. Then, the Honest Common Core property follows from Lemma 35. Finally, Lemma 37 ensures that all honest parties output at time $(2c_{\mathsf{rBC}} + c'_{\mathsf{rBC}}) \cdot \Delta$, hence Synchronized Termination also holds. □

Asynchronous Network. In the following, we assume that the network is asynchronous, and at most t_a of the parties involved are corrupted.

Lemma 39. *For every honest party* P*, it eventually holds that* $\left|W_1\right| \geq n - t_s$.

Proof. We first note that $\left|W_0\right| \geq n - t_s$ eventually holds: this follows from Π_{rBC}'s Validity and c_{rBC}-Honest Termination properties, which ensures that every honest value gets delivered.

Hence, every honest party eventually sends W_0 to via Π_{rBC}. These sets are also eventually delivered. Then, Π_{rBC} ensures that every value included by an honest party in its set $\mathcal{M}$ is also received by all other parties due to Consistency and c'_{rBC}-Conditional Termination, and therefore at least $n - t_s$ parties are marked as level one witnesses eventually. □

Lemma 40. *For every honest party* P*, it eventually holds that* $\left|W_2\right| \geq n - t_s$.

Proof. According to Lemma 39, every honest party sends its set W_1 eventually via Π_{rBC}. Then, these sets are eventually received by Validity and c_{rBC}-Honest Termination. In addition, if $P'' \in W_1$ for some honest party P', then eventually every honest party receives the same set W''_0 that P' has received from P' due to Π_{rBC}'s Consistency and c'_{rBC}-Conditional Termination, and therefore may add P'' to their sets W_1. Hence, every honest party may add P' to its set W_2. Therefore, eventually $\left|W_2\right| \geq n - t_s$ for every honest party. □

Lemma 41. *Let* H *denote the first* $n - t_s$ *parties for whom* $\left|W_1\right| \geq n - t_s$ *holds. Then, there is an honest party* $P^\star$ *such that* $t_s + 1$ *parties in* H *have included* $P^\star$ *in their sets* W_1.

Proof. We prove this result with the help of a counting argument. We first provide a lower bound on the number of honest sets W_0 received in total by the parties in H. Since each of the $n - t_s$ in H has received $n - t_s$ such sets, and at least $n - t_s - t_a$ out of these sets were sent by honest parties, parties in H have received $(n - t_s - t_a) \cdot (n - t_s)$ such sets.

Then, assume that there is no such party $P^\star$, that was included in the W_1 sets of at least $t_s + 1$ parties in H. This implies that every honest party has been included in the W_1 sets of at most t_s parties in H. Therefore, the parties in H have received in total at most $t_s \cdot (n - t_s)$ sets W_0 from honest parties. This leads to a contradiction, as $n > 2 \cdot t_s + t_a$ implies that $(t_s + 1) \cdot (n - t_s) < (n - t_s - t_a) \cdot (n - t_s)$. □

Lemma 42. *Assume* $\left|W_2\right| \geq n - t_s$ *holds for all honest parties. Then, there is set of* $n - t$ *common value-sender pairs in their sets* $\mathcal{M}$.

Proof. Lemma 41 ensures that there is an honest party $P^\star$ included in the W_1 sets of at least $t_s + 1$ honest parties. Since parties wait until $\left|W_2\right| \geq n - t_s$ holds, they add at least one of these $t_s + 1$ parties to their sets W_2, and therefore are forced to wait until they add $P^\star$ to their set W_1. That is, until the set $W_0^\star$ sent by $P^\star$ is received, and $W_0^\star \subseteq W_0$ holds. This will eventually be the case, due to Π_{rBC}'s properties. Then, the value-sender pairs corresponding to the at least $n - t_s$ parties in $W_0^\star$ are included in the each of the honest parties' output sets $\mathcal{M}$. □

Lemma 43. Π_{GTHR} *satisfies Termination,* t_s*-Common Core, Validity, and Consistency when running in an asynchronous network where at most* t_a *of the parties involved are corrupted.*

Proof. Validity and Consistency follow immediately from the properties of Π_{rBC}. Then, Lemma 40 ensures Termination, and, together with Lemma 42, ensures t_s-Common Core. □

C.4 Core-Set Agreement When $n > 2 \cdot t_s + t_a$

We may now present our best-of-both-worlds protocol Π_{ACS} enabling us to achieve CC with optimal resilience.

As described in the main body of the paper, our approach for designing this protocol relies on GTHR. Parties first distribute their inputs via the Π_{GTHR} protocol realizing Theorem 19, described in Section C.3. When receiving output $\mathcal{M}_{\mathsf{GTHR}}$, party P join an invocation of the Π_{BA} protocol described in Theorem 17 for each party P': with input 1 if its set $\mathcal{M}_{\mathsf{GTHR}}$ contains some value from P' and 0 otherwise.

Then, parties output the set value-sender pairs of senders for whom the Π_{BA} invocations have resulted in output 1. Note that, although this implies that at least one honest party had input 1 in this invocation, this does not imply that every honest party has received the corresponding value via Π_{GTHR}. In fact, we make use of an additional property that the implementation of GTHR presented in Section C.3 provides, if the protocol is allowed to continue running for sufficient amount of time even after parties obtain output (regardless of whether the network is synchronous or asynchronous).

Lemma 44. *Let P and P' denote two honest parties, and let $\mathcal{M}$ and $\mathcal{M}'$ denote their message sets in Π_{GTHR}. If $(v, P'') \in \mathcal{M}$, then, eventually $(v, P'') \in \mathcal{M}'$ as well.*

Proof. Follows from Π_{rBC}'s Consistency and c'_{rBC}-Conditional Termination properties: once P receives a value via Π_{rBC}, P' receives it as well. □

We provide the code of our Π_{ACS} protocol below.

Protocol Π_{ACS}

Code for party P with input v

1: Join Π_{GTHR} with input v.
2: When receiving output $\mathcal{M}_{\mathsf{GTHR}}$ from Π_{GTHR}:
3: Join an invocation of Π_{BA} for each party P': with input 1 if $(v', P') \in \mathcal{M}_{\mathsf{GTHR}}$ for some v', and 0 otherwise.
4: Keep running line 2 of Π_{GTHR}:
5: Whenever receiving v' from P' via Π_{rBC} initiated in Π_{GTHR}, add (v', P') to $\mathcal{M}_{\mathsf{GTHR}}$.
6: When obtaining outputs in all invocations of Π_{BA}:
7: $\mathcal{P}$:= the set of parties whose Π_{BA} invocations returned output 1
8: When $(v', P') \in \mathcal{M}_{\mathsf{GTHR}}$ for every $P' \in \mathcal{P}$:
9: $\mathcal{M}$:= the set of pairs $(v', P') \in \mathcal{M}_{\mathsf{GTHR}}$ with $P' \in \mathcal{P}$.
10: Output $\mathcal{M}$ and terminate.

Theorem 15. *Let n, t_s, t_a be such that $n > 2 \cdot t_s + t_a$ and $t_a \leq t_s$. Then, there is a protocol Π_{ACS} achieving (t_s, t_a)-resilient ACS.*

Proof. First, the Validity property follows immediately from the Validity properties of Π_{rBC} and Π_{GTHR}. Next, Π_{GTHR} ensures that all honest parties obtain sets $\mathcal{M}_{\mathsf{GTHR}}$ that intersect in at least $n - t_s$ values. In addition, if the network is synchronous, these sets contain all honest values, and are obtained simultaneously.

Therefore, if the network is synchronous, all parties join the Π_{BA} invocations simultaneously, hence all properties that Π_{BA} ensures when running in a synchronous network hold. It follows that parties obtain output 1 for every honest party, and hence all honest values are included in the output sets $\mathcal{M}$. Moreover, parties agree on the same bit for every corrupted party. If the output bit for some corrupted party is 1, then at least one honest party has included this corrupted party's value in its set $\mathcal{M}_{\mathsf{GTHR}}$. Lemma 44 ensures that all honest parties receive this value as well. Therefore, all honest parties output the same set $\mathcal{M}$.

If the network is asynchronous, since Π_{GTHR} ensures Termination, all honest parties eventually join the Π_{BA} invocations, and hence agree on a bit for each party due to Weak Validity. Π_{GTHR}'s Common Core property ensures that all honest parties input 1 in at least $n - t_s$ of the Π_{BA} invocations, and therefore output 1 in these invocations. For each invocation returning 1, the Weak Validity property ensures that at least one honest party P has input 1, meaning that P has received the corresponding value via Π_{GTHR}. Lemma 44 then ensures that all parties eventually receive this value, and therefore all parties output the same set $\mathcal{M}$ of at least $n - t_s$ values. □

D Properties of The Safe Area

In this section, we prove a few useful properties of the honest parties' safe areas. In some of our proofs, we make use of the following version of the Pigeonhole principle.

Lemma 45 (Pigeonhole Principle)**.** *Let S be a finite set and consider m subsets $S_1, \ldots, S_m$ of S. If $\sum_{i=1}^{m} |S_i| > (m-1) \cdot |S|$, then $\bigcap_{i \in [m]} S_i \neq \varnothing$.*

Proof. For each $s \in S$, write $X_s = \{i \in [m] : s \in S_i\}$. Observe that $\sum_{s \in S} |X_s| = \sum_{i=1}^{m} |S_i| > (m-1) \cdot |S|$. If $|X_s| \leq m-1$ holds for all $s \in S$, then $\sum_{s \in S} |X_s| \leq (m-1) \cdot |S|$ would hold, which we know is not the case. Hence, for some $s \in S$ we have $|X_s| = m$, from which $s \in \bigcap_{i \in [m]} S_i$. □

We first prove the central lemma for the analysis of our CC protocol, ensuring honest parties compute non-empty safe areas. For this, $n > \max(\omega \cdot t_s, \omega \cdot t_s + t_a)$ is assumed, where recall that $t_a \leq t_s$ and ω is the Helly number of the convexity space.

Lemma 14. *Assume $n > \max(\omega \cdot t_s, \omega \cdot t_a + t_s)$, and that $\mathcal{M}$ is a set of $n - t_s + k$ value-party pairs, where $0 \leq k \leq t_s$. Then, $safe_{\max(k,t_a)}(\mathcal{M}) \neq \varnothing$.*

Proof. For brevity, write $t' = \max(k, t_a)$. Recall that $safe_{t'}(\mathcal{M}) := \bigcap_{M \in restrict_{t'}(\mathcal{M})} \langle M \rangle$, where $restrict_{t'}(\mathcal{M}) := \{M \subseteq \mathcal{M} : |M| = |\mathcal{M}| - t'\}$. To show that an intersection of convex sets is non-empty, it suffices to show that any ω of them intersect (by the definition of ω). Consider ω sets $M_1, \ldots, M_\omega \in restrict_{t'}(\mathcal{M})$. We show that $\cap_{i=1}^{\omega} M_i \neq \varnothing$ using Lemma 45, from which the required $\cap_{i=1}^{\omega} \langle M_i \rangle \neq \varnothing$ naturally follows. To apply the lemma, we need that $\sum_{i=1}^{\omega} |M_i| > (\omega - 1) \cdot |\mathcal{M}|$. Since $\sum_{i=1}^{\omega} |M_i| = \omega \cdot (|\mathcal{M}| - t')$, this amounts to showing that $\omega \cdot (|\mathcal{M}| - t') > (\omega - 1) \cdot |\mathcal{M}| \iff |\mathcal{M}| > \omega \cdot t' \iff n - t_s + k > \omega \cdot t'$. Distinguish two cases:

- If $k \geq t_a$, the latter becomes $n - t_s + k > \omega \cdot k \iff n - t_s > (\omega - 1) \cdot k$, which is true since $k \leq t_s$ and $n > \omega \cdot t_s$.
- Otherwise, $k < t_a$, and the latter becomes $n - t_s + k > \omega \cdot t_a$, which is true since $k \geq 0$ and $n > \omega \cdot t_a + t_s$.

□

The rest of the section builds towards proving Lemma 20, which is the central result ensuring the correctness of our AA protocol for chordal graphs. From this point on, our results make the stronger assumption $n > \omega \cdot t_s + t_a$. This will often not be stated explicitly in the statements of the lemmas to avoid unnecessary repetition.

Lemma 46. *Let $\mathcal{M}_1, \mathcal{M}_2$ denote two sets of value-sender pairs such that $|\mathcal{M}_1 \cup \mathcal{M}_2| \leq n$, and $|\mathcal{M}_1 \cap \mathcal{M}_2| \geq n - t_s$. Assume that $k_1 = |\mathcal{M}_1| - (n - t_s) \geq t_a$, and define $k_\cup = |\mathcal{M}_1 \cup \mathcal{M}_2| - (n - t_s)$. Then, $safe_{k_1}(\mathcal{M}_1) \supseteq safe_{k_\cup}(\mathcal{M}_1 \cup \mathcal{M}_2) \neq \varnothing$.*

Proof. Note that $t_a \leq k_\cup \leq t_s$. Lemma 14 then immediately implies that $safe_{k_\cup}(\mathcal{M}_1 \cup \mathcal{M}_2) \neq \varnothing$. Moreover, $restrict_{k_1}(\mathcal{M}_1) = \{M \subseteq \mathcal{M}_1 : |M| = n - t_s\} \subseteq \{M \subseteq \mathcal{M}_1 \cup \mathcal{M}_2 : |M| = n - t_s\} = restrict_{k_\cup}(\mathcal{M}_1 \cup \mathcal{M}_2)$. Hence, $safe_{k_1}(\mathcal{M}_1) \supseteq safe_{k_\cup}(\mathcal{M}_1 \cup \mathcal{M}_2)$. □

The following is a useful monotonicity property of safe areas.

Lemma 47. *Let m be a value-party pair and $\mathcal{M}$ a set of value-party pairs. Then, for any t we have $safe_t(\mathcal{M}) \subseteq safe_{t-1}(\mathcal{M})$ and $safe_t(\mathcal{M}) \subseteq safe_t(\mathcal{M} \cup \{m\})$.*

Proof. We reason equatorially:

$$\begin{aligned} safe_t(\mathcal{M}) &= \bigcap_{M \in restrict_t(\mathcal{M})} \langle M \rangle \subseteq \bigcap_{M \in restrict_t(\mathcal{M})} \left(\bigcap_{m' \in \mathcal{M} \setminus M} \langle M \cup \{m'\} \rangle \right) \\ &= \bigcap_{M \in restrict_{t-1}(\mathcal{M})} \langle M \rangle = safe_{t-1}(\mathcal{M}) \end{aligned}$$

$$\begin{aligned} safe_t(\mathcal{M} \cup \{m\}) &= safe_{t-1}(\mathcal{M}) \cap \left(\bigcap_{M \in restrict_t(\mathcal{M})} \langle M \cup \{m\} \rangle \right) \\ &\supseteq safe_t(\mathcal{M}) \cap \left(\bigcap_{M \in restrict_t(\mathcal{M})} \langle M \rangle \right) = safe_t(\mathcal{M}) \end{aligned}$$

□

Lemma 48. *Let $\mathcal{M}_1, \mathcal{M}_2$ be two sets of value-party pairs such that $\left|\mathcal{M}_1 \cup \mathcal{M}_2\right| \leq n$, and $\left|\mathcal{M}_1 \cap \mathcal{M}_2\right| \geq n - t_s$. Assume that $\left|\mathcal{M}_1\right| - (n - t_s) \leq t_a$. Then, $safe_{t_a}(\mathcal{M}_1) \supseteq safe_{t_a}(\mathcal{M}_1 \cap \mathcal{M}_2) \neq \varnothing$.*

Proof. Since $n - t_s \leq \left|\mathcal{M}_1 \cap \mathcal{M}_2\right| \leq \left|\mathcal{M}_1\right| \leq n - t_s + t_a$, Lemma 14 implies that $safe_{t_a}(\mathcal{M}_1 \cap \mathcal{M}_2) \neq \varnothing$. Moreover, Lemma 47 implies that $safe_{t_a}(\mathcal{M}_1 \cap \mathcal{M}_2) \subseteq safe_{t_a}(\mathcal{M}_1)$. □

Lemma 49. *Let $\mathcal{M}_1, \mathcal{M}_2$ be two sets of value-party pairs such that $\left|\mathcal{M}_1 \cup \mathcal{M}_2\right| \leq n$, and $\left|\mathcal{M}_1 \cap \mathcal{M}_2\right| \geq n - t_s$. Assume that $\left|\mathcal{M}_1\right| - (n - t_s) \leq t_a$ and $\left|\mathcal{M}_2\right| - (n - t_s) > t_a$, and define $k_\cup = \left|\mathcal{M}_1 \cup \mathcal{M}_2\right| - (n - t_s)$. Then, $safe_{t_a}(\mathcal{M}_1) \cap safe_{k_\cup}(\mathcal{M}_1 \cup \mathcal{M}_2) \neq \varnothing$.*

Proof. In order to prove this result, it will be useful for us to unroll the definition of the safe areas. The statement becomes the following:

$$\bigcap_{M \in restrict_{t_a}(\mathcal{M}_1)} \langle M \rangle \quad \cap \quad \bigcap_{M \in restrict_{k_\cup}(\mathcal{M}_1 \cup \mathcal{M}_2)} \langle M \rangle \neq \varnothing.$$

It suffices to prove that any ω terms of this intersection have a non-empty intersection. That is, for any $a, b \geq 0$ with $a + b = \omega$, every a elements $X_1, X_2, \ldots, X_a$ of $restrict_{t_a}(\mathcal{M}_1)$ and b elements $Y_1, Y_2, \ldots Y_b$ of $restrict_{k_\cup}(\mathcal{M}_1 \cup \mathcal{M}_2)$ have a non-empty intersection, implying the same holds about their convex hulls.

Note that the edge-cases $(a, b) \in \{(0, \omega), (\omega, 0)\}$ can be proven analogously to Lemma 14, which shows that safe areas are non-empty.

From this point on, we may assume that $a, b \geq 1$. For this case, we will show a slightly stronger claim: $\bigcap_{i=1}^{a} X_i \cap \bigcap_{i=1}^{b} (Y_i \cap \mathcal{M}_1) \neq \varnothing$. This way, to apply Lemma 45 it suffices to show that $\sum_{i=1}^{a} \left|X_i\right| + \sum_{i=1}^{b} \left|Y_i \cap \mathcal{M}_1\right| > (\omega - 1) \cdot \left|\mathcal{M}_1\right|$.

We first provide lower bounds for the sizes of sets X_i and $Y_i \cap \mathcal{M}_1$: each set X_i has size $\left|\mathcal{M}_1\right| - t_a$ and each set $Y_i \cap \mathcal{M}_1$ has size at least $\left|\mathcal{M}_1\right| - t_s = (n - t_s) - (t_s - k_1)$, where $k_1 = \left|\mathcal{M}_1\right| - (n - t_s)$. The latter is non-trivial to see: note that $Y_i \cap \mathcal{M}_1 = Y_i \setminus (\mathcal{M}_2 \setminus \mathcal{M}_1)$, from which $\left|Y_i \cap \mathcal{M}_1\right| \geq \left|Y_i\right| - \left|\mathcal{M}_2 \setminus \mathcal{M}_1\right| = (n - t_s) - \left|\mathcal{M}_2 \setminus \mathcal{M}_1\right|$. Moreover, $\left|\mathcal{M}_2 \setminus \mathcal{M}_1\right| \leq \left|\mathcal{M}_1 \cup \mathcal{M}_2\right| - \left|\mathcal{M}_1\right| \leq n - (n - t_s + k_1) = t_s - k_1$.

Since $t_a \leq t_s$, we obtain that $\left|\mathcal{M}_1\right| - t_s \leq \left|\mathcal{M}_1\right| - t_a$. Hence, $a \cdot (\left|\mathcal{M}_1\right| - t_a) + b \cdot (\left|\mathcal{M}_1\right| - t_s) \geq (\left|\mathcal{M}_1\right| - t_a) + (\omega - 1) \cdot (\left|\mathcal{M}_1\right| - t_s)$. We want to show that $(\left|\mathcal{M}_1\right| - t_a) + (\omega - 1) \cdot (\left|\mathcal{M}_1\right| - t_s) > (\omega - 1) \cdot \left|\mathcal{M}_1\right|$, which is the same as $\left|\mathcal{M}_1\right| > t_a + (\omega - 1) \cdot t_s$. This holds because $\left|\mathcal{M}_1\right| \geq n - t_s$ and $n > \omega \cdot t_s + t_a$.

Hence, Lemma 45 applies, so any ω of the relevant sets intersect, so their convex hulls also intersect. Then, by the definition of the Helly number ω all relevant sets intersect, proving our claim. □

Lemma 20. *Let $(\mathcal{M}_i)_{i=1}^K$ be sets of value-party pairs such that $k_i := \left|\mathcal{M}_i\right| - (n - t_s) \geq 0$. If $\left|\bigcup_{i=1}^K \mathcal{M}_i\right| \leq n$ and $\left|\bigcap_{i=1}^K \mathcal{M}_i\right| \geq n - t_s$ hold, then $\bigcap_{i=1}^K safe_{\max(k_i,t_a)}(\mathcal{M}_i) \neq \varnothing$.*

Proof. First, note that, for every $1 \leq i \leq K$ it holds that $0 \leq k_i \leq t_s$.

Write $\mathcal{M}_\cup := \bigcup_{i=1}^K \mathcal{M}_i$, and $\mathcal{M}_\cap := \bigcap_{i=1}^K \mathcal{M}_i$ and moreover define $k_\cup := \mathcal{M}_\cup - (n - t_s)$ and $k_\cap := \mathcal{M}_\cap - (n - t_s)$. Note that, according to Lemma 14, the safe areas of these two sets, namely $S_\cup := safe_{\max(k_\cup,t_a)}(\mathcal{M}_\cup)$ and $S_\cap = safe_{\max(k_\cap,t_a)}(\mathcal{M}_\cap)$ are non-empty. Then, Lemma 48 ensures that $S_\cap \neq \varnothing$ is included in the safe area of each set $\mathcal{M}_i$ with $k_i \leq t_a$. If there is no set $\mathcal{M}_i$ with $k_i > t_a$, our statement is proven. Similarly, Lemma 46 ensures that $S_\cup \neq \varnothing$ is included in the safe area of every set $\mathcal{M}_i$ with $k_i > t_a$. Hence, there is no $\mathcal{M}_i$ with $k_i \leq t_a$, our statement is proven.

If there is at least a set $\mathcal{M}_i$ with $k_i \leq t_a$ and a set $\mathcal{M}_j$ with $k_j > t_a$, it follows that that $k_\cap \leq t_a$, while $t_a < k_\cup \leq t_s$. Applying Lemma 49, we obtain that $\exists v \in S_\cap \cap S_\cup$. Then, since $S_\cap$ is included in the safe area of $\mathcal{M}_i$ with $k_i \leq t_a$, and $S_\cup$ is included in the safe area of any $\mathcal{M}_j$ with $k_j > t_a$, v belongs to all honest safe areas, which concludes the proof. □

E Approximate Agreement

We first establish that, when the network is synchronous, since Π_{GTHR} ensures Simultaneous Termination, its strong synchronous guarantees hold in every iteration. Then, regardless of whether the network is synchronous or asynchronous, the Termination property of Π_{Chordal} is trivially achieved. In the following, we focus on Validity and Agreement.

Lemma 50. *Let G be a chordal graph and A be a convex set of vertices in G. Let $a \in A$ be a simplicial vertex in the subgraph of G induced by A. Then, $a \in ex(A)$.*

Proof. Assume for a contradiction $a \in \langle A \setminus \{a\}\rangle$. Since in general $A \setminus \{a\} \subseteq \langle A \setminus \{a\}\rangle$, this means that $\langle A \setminus \{a\}\rangle = A$. Hence, considering the computation of $\langle A \setminus \{a\}\rangle$ by iterating the "take all nodes on induced paths" operator in Section 2.2.1, there is an induced path P between two vertices $b, c \in A \setminus \{a\}$ that passes through a. Because A is convex, all vertices in P are included in A. Because a is neither the beginning nor the end of P, it follows that a has two neighbors in $A \setminus \{a\}$ that are on P. However, since a is simplicial in A, those two neighbors are joined by an edge, contradicting the fact that P is an induced path. □

Lemma 51. *Let A and B be convex sets such that $A \subseteq B$, and consider $a \in A$. If $a \notin ex(A)$, then $a \notin ex(B)$. More succinctly, $A \setminus ex(A) \subseteq A \setminus ex(B)$.*

Proof. Write $\langle B \setminus \{a\}\rangle = \langle (B \setminus A) \cup (A \setminus \{a\})\rangle = \langle (B \setminus A) \cup \langle (A \setminus \{a\})\rangle\rangle = \langle (B \setminus A) \cup A\rangle = \langle B\rangle$, where we have used the standard property that $\langle X \cup Y\rangle = \langle X \cup \langle Y\rangle\rangle$. □

Lemma 52 (Dirac'1961, [14])**.** *Every chordal graph has a simplicial vertex. Every chordal graph that is not a clique has two non-adjacent simplicial vertices.*

Lemma 22. *Assume $1 \leq \mathit{it} \leq \mathit{max\_it}$ and that $H_{\mathit{it}-1}$ does not form a clique in G. Then, $H_{\mathit{it}} \subsetneq H_{\mathit{it}-1}$.*

Proof. Write $s = \min H_{\mathsf{it}-1}$. Note that s is simplicial in $H_{\mathsf{it}-1}$ by definition of ordering $\succ$, and hence it is also extremal by Lemma 50. Let $\mathcal{P}$ be the set of honest parties. Denote by S_{it}^P the safe area computed by honest party P in iteration it. Note that, by Lemma 14, $S_{\mathsf{it}}^P \subseteq H_{\mathsf{it}-1}$. The properties of Π_{GTHR} enable us to apply Lemma 20 and therefore obtain that the intersection of the honest parties' safe areas is non-empty, so consider an arbitrary $y \in \cap_{P \in \mathcal{P}} S_{\mathsf{it}}^P$. We now distinguish two cases:

1. If $y \neq s$, consider some honest party $P \in \mathcal{P}$ with computed safe area S^P_{it}. We will show that $v^P_{\mathsf{it}} \neq s$. Since s is extremal, this implies that $s \in H_{\mathsf{it}-1} \setminus H_{\mathsf{it}}$, and hence the conclusion. Consider two cases, corresponding to the two cases in the algorithm:
   (a) If $S^P_{\mathsf{it}} = ex(S^P_{\mathsf{it}})$, then party P sets $v^P_{\mathsf{it}} = \max S^P_{\mathsf{it}}$. However, as $y \in S^P_{\mathsf{it}}$ and $y \succ s$ it follows that $v^P_{\mathsf{it}} \succ s$, so $v^P_{\mathsf{it}} \neq s$, as claimed.
   (b) If $S^P_{\mathsf{it}} \neq ex(S^P_{\mathsf{it}})$, then party P picks $v^P_{\mathsf{it}} \in S^P_{\mathsf{it}} \setminus ex(S^P_{\mathsf{it}})$ arbitrarily. Since $S^P_{\mathsf{it}} \subseteq H_{\mathsf{it}-1}$, by Lemma 51, it follows that $v^P_{\mathsf{it}} \in S^P_{\mathsf{it}} \setminus ex(H_{\mathsf{it}-1})$. As $s \in ex(H_{\mathsf{it}-1})$, this means that $v^P_{\mathsf{it}} \neq s$, as claimed.
2. If $y = s$, then, since $H_{\mathsf{it}-1}$ does not induce a clique, by Lemma 52, the subgraph induced by $H_{\mathsf{it}-1}$ in G has two simplicial vertices $a, b \in H_{\mathsf{it}-1}$ which are not joined by an edge. For the proof, we will need two such vertices where one of them is s. Recall that s is known to be simplicial. If $s \notin \{a, b\}$ and the graph has both edges $s - a$ and $s - b$, then that would contradict the fact that s is simplicial. Hence, without loss of generality, assume that s and a are distinct simplicial vertices with no edge between them in the graph. We will now show that for any honest party $P \in \mathcal{P}$ it holds that $v^P_{\mathsf{it}} \neq a$. If $a \notin S^P_{\mathsf{it}}$, then this is clear, so assume $a \in S^P_{\mathsf{it}}$. As a result, we know that $\{s, a\} \subseteq S^P_{\mathsf{it}}$. Since the edge $s - a$ does not exist in G and its endpoints are contained in S^P_{it}, it follows that S^P_{it} does not induce a clique, meaning by Lemma 1 that S^P_{it} is not free, so $S^P_{\mathsf{it}} \neq ex(S^P_{\mathsf{it}})$. As a result, party P picks $v^P_{\mathsf{it}} \in S^P_{\mathsf{it}} \setminus ex(S^P_{\mathsf{it}})$ arbitrarily. Since $S^P_{\mathsf{it}} \subseteq H_{\mathsf{it}-1}$, by Lemma 51, it follows that $v^P_{\mathsf{it}} \in S^P_{\mathsf{it}} \setminus ex(H_{\mathsf{it}-1})$. Since a is a simplicial vertex in the subgraph induced by $H_{\mathsf{it}-1}$, it follows by Lemma 50 that $a \in ex(H_{\mathsf{it}-1})$, so $v^P_{\mathsf{it}} \neq a$, as claimed. To conclude, $a \in H_{\mathsf{it}-1} \setminus H_{\mathsf{it}}$, from which the conclusion follows.

□

Theorem 23. *Assume $n \geq 3$ and that the network is asynchronous, then for any $d \geq 0$ there is a graph G_d with $\Theta(d)$ vertices and edges such that no deterministic n-party protocol resilient against two crashes satisfies Geodesic Convex-Hull Validity, Termination, and Agreement within Graph Distance d.*

Proof. It suffices to prove this for odd d, as for even d one can just use the result for $d + 1$ to get the conclusion. Hence, it equivalently suffices to prove the result for $2d + 1$ for all $d \geq 0$. Consider a cycle graph G consisting of $6d + 3$ nodes. For graph G, assume Π is a deterministic n-party protocol achieving geodesic convex-hull validity, termination, and agreement within graph distance $2d + 1$ if the network is asynchronous and at most two parties might crash. Designate nine nodes $v_0, \ldots, v_8$ in, say, clockwise order along the cycle such that the distances $dist(v_i, v_{i+1})$ in order for $0 \leq i < 9$ are $d, 1, d, d, 1, d, d, 1, d$, where we assumed for convenience that v_9 denotes v_0. Construct a new protocol Π' which runs Π followed by each party P taking their output v^P_{out} from Π and computing their output for Π' as the node $v \in \{v_0, v_3, v_6\}$ which minimizes the distance $dist(v^P_{out}, v)$. Note that, because the distance between any two distinct nodes in $\{v_0, v_3, v_6\}$ is $2d + 1$, and hence odd, this node v is uniquely determined. Let us now investigate the guarantees achieved by protocol Π'. First, it can not be that three distinct honest parties output v_0, v_3 and respectively v_6, as one can check that this would require some outputs of Π to be more than distance $2d + 1$ apart, which can not be the case by assumption. Hence, Π' satisfies 2-Set Agreement. Moreover, note that Π' inherits termination from Π. Now, say we restrict protocol Π' to only allow parties to have inputs in $\{v_0, v_3, v_6\}$, making it a protocol with the three-valued input-output domain $\{v_0, v_3, v_6\}$, which we so far know satisfies termination and 2-Set Agreement. Because Π satisfies convex-hull validity, notice that if all honest parties have the same input, then they will also have the same output, equalling this input. Moreover, if the set of honest inputs consists of two values, say without loss of generality v_0 and v_3, then the set of honest outputs for Π will be contained in $\langle v_0, v_3 \rangle$, which is the path between v_0 and v_3

in G. Hence, the outputs for Π' computed by the honest parties will be contained in $\{v_0, v_3\}$. As a result, Π' satisfies Honest-Input Validity, meaning that the set of honest outputs is a subset of the set of honest inputs. Overall, Π' is an n-party deterministic protocol with a three-valued input-output domain that guarantees termination, 2-Set Agreement and Honest-Input Validity when the network is asynchronous and at most two parties may crash. Such a protocol is known not to exist for $n \geq 3$ by standard combinatorial topology arguments [21]. This is true for crash failures even if Honest-Input Validity is replaced with Any-Input Validity. Hence, Π can not exist either. □