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 abstract 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 network-agnostic model. These are a deterministic primitive allowing parties to obtain intersecting views (*Gather*), and a randomized primitive leading to identical views (*Agreement on a Core-Set*). Our primitives provide stronger guarantees than previous counterparts, making them of independent interest.

Keywords: convex consensus, network-agnostic protocols, agreement on a core-set.

Acknowledgements: We thank Julian Loss and the anonymous reviewers for their useful suggestions.

1 Introduction

Arranging a meeting place for a group of n people in a city is a common problem, as determining a location that is convenient and accessible for everyone can be challenging. While locations can be determined by their geographic coordinates, we need to prevent agreement on the coordinates of a restricted area, e.g., some private property. Hence, it may be more realistic to represent the city as a graph, with streets modeled as edges and intersections as vertices. Participants are initially in different locations (i.e., vertices), and they want to agree on a vertex for the meeting point using pair-wise communication channels. Finding such a meeting point, while also considering that some of the participants may choose not to 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 the set of locations) equipped with a convexity notion $\mathcal{C}$ (roughly meant to formalize which 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

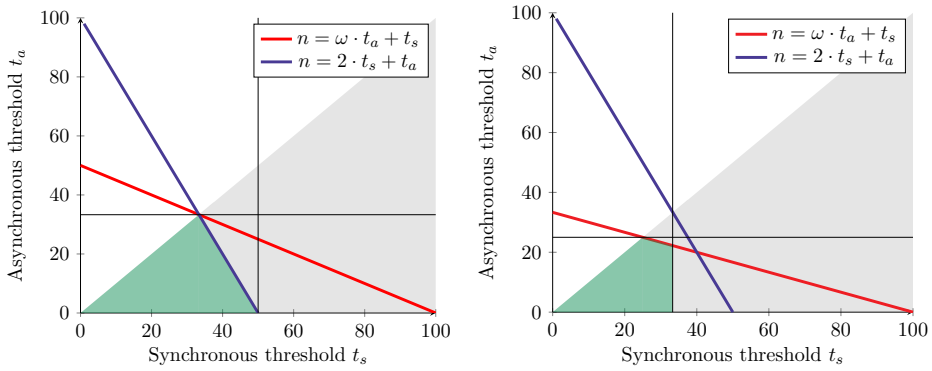
monophonic convexity, defined over minimal/chordless paths. For a given convexity notion, CC is concerned with enabling parties to agree on a value in the convex hull of their inputs. This should be achieved even if up to t of the parties are corrupted (byzantine) and may exhibit malicious behavior.

A natural question to ask is *"For which values of t can CC be achieved?"*. Prior work provides an almost complete answer for *the synchronous model*, i.e., where the parties' clocks are synchronized and messages get delivered within a known amount of time Δ. In this model, the solvability of CC depends on the structure of the input space: concretely, on the space's Helly number ω (e.g., $D+1$ for $\mathbb{R}^D$ with straight-line convexity). CC can be solved in the synchronous model if $t < n/\omega$, and this condition is also necessary for convex geometries (a restricted class of convexity spaces) and for $\mathbb{R}^D$ with straight-line convexity [37, 41].

One may argue that the synchronous model's assumptions are too strong: in practice, the maximum delay Δ will often be violated during times of increased network load or outages. A well-established alternative is the *asynchronous model*. This only assumes that parties' messages get delivered eventually, leading to protocols that are highly robust to adverse network conditions. The solvability of CC in this model has only been partly characterized so far: for convex geometries and $\mathbb{R}^D$ with straight-line convexity, the condition $t < n/(\omega+1)$ is necessary [37,41]. This is, however, only known to be sufficient in the asynchronous model for a relaxed version of CC which allows parties to agree up to some error (Approximate Agreement, AA), and only on particular input spaces. We highlight that the asynchronous model comes with an intrinsic limitation: even if the condition $t < n/(\omega+1)$ were to be proven sufficient for achieving asynchronous CC in all convexity spaces, there is a gap between this threshold and the $t < n/\omega$ threshold sufficing for synchronous networks. That is, an asynchronous CC protocol (which would have to be randomized, due to [21]) would achieve its guarantees regardless of the network conditions, but at the expense of tolerating a lower number of corruptions in comparison to synchronous alternatives.

This is where a third model steps in: *the network-agnostic model*, introduced by Blum, Katz, and Loss [9], which aims to combine the advantages of both established models. This model has gained significant popularity in recent years, and has covered problems such as Byzantine Agreement [9, 15], AA on real and multidimensional values [23, 24], State-Machine Replication [10] and Multi-Party Computation [6,11,15]. Concretely, given two thresholds $t_a \leq t_s$, a network-agnostic protocol should tolerate t_s corruptions if the network is synchronous and t_a if it is asynchronous, without knowing which of the two happens to be the case. We add that such a network-agnostic protocol with $t_a = 0$ still provides superior guarantees in comparison to a synchronous counterpart: it maintains its properties even if the synchrony assumptions fail provided that no party is corrupted.

Our work will primarily investigate the necessary and sufficient conditions for achieving CC for arbitrary convexity spaces in the *network-agnostic model*. We provide a complete characterization, showing that the condition $n > \max(\omega \cdot t_s, \omega \cdot t_a + t_s, 2 \cdot t_s + t_a)$ is necessary and sufficient in any convexity space with $\omega > 1$. This condition, illustrated in Figure 1, allows for a trade-off between the two expected optimal resilience bounds of the pure models (which we additionally prove to be tight leading up to our main result). We add that $\omega = 1$ refers to convexity spaces where some $v \in V$ is contained in all non-empty convex sets, allowing for trivial protocols (parties may simply output v).



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

Figure 1: Our results on the 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 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 $\omega = 3$.

1.1 Our contributions

Impossibility results. We first prove that the condition $n > \max(\omega \cdot t_s, \omega \cdot t_a + t_s, 2 \cdot t_s + t_a)$ is necessary. We additionally generalize the aforementioned lower bounds from convex geometries and $\mathbb{R}^D$ to all convexity spaces, so that $t < n/\omega$ is required for the synchronous case and $t < n/(\omega + 1)$ for the asynchronous one. For these proofs, we define *adversarial families*, which will allow us to derive general scenario-based arguments [36].

Feasibility results. Afterwards, we show that the condition $n > \max(\omega \cdot t_s, \omega \cdot t_a + t_s, 2 \cdot t_s + t_a)$ is also sufficient: we give a protocol achieving CC when this condition holds. Together with our impossibility results, this completes the landscape of feasibility for the purely synchronous, purely asynchronous, and network-agnostic models. Our protocol assumes cryptographic setup, namely digital signatures. Note that this is necessary to tolerate $t_s < n/\omega$ corruptions in the synchronous model for $\omega = 2$, a fact which can be easily inferred from various impossibility results (e.g., [27]). When $\omega \geq 3$, however, signatures are no longer required (since $t_s < n/3$ holds in that case), and we will briefly explain how they can be removed. We also note that our protocol is randomized (which is needed [21]), but randomization is restricted to Byzantine Agreement subprotocols [15]. The use of signatures is similarly constrained to Reliable Broadcast and Byzantine Agreement subprotocols [15, 34].

Network-agnostic communication primitives. The core of our CC protocol is a novel network-agnostic implementation of Agreement on a Core-Set (ACS) [8], which may be of independent interest. In essence, ACS allows parties to distribute their inputs and obtain identical views. Our ACS protocol provides stronger guarantees than previous network-agnostic variants [4,10,11]. These stronger guarantees will be crucial for achieving CC: when the network is synchronous, we enable the parties to obtain a common view that includes all the honest parties' inputs. Obtaining these properties for ACS when

$n > 2 \cdot t_s + t_a$ requires us to diverge from the outline of previous ACS constructions and to instead provide a novel implementation relying on *Gather* (GTHR) [2, 13], a second primitive that we adapt to the network-agnostic model. Roughly, GTHR enables parties to obtain intersecting views.

Prior works corrections. We need to note that our findings seemingly contradict an impossibility result of [37] for general convexity spaces, which depends on the Carathéodory number of the convexity space and not on its Helly number (in general, there is no relationship between the two). Upon closer inspection, we exhibit an error in the proof of [37], meaning that the correct bound is in terms of ω, and not the Carathéodory number.

As a secondary contribution, we identify a core issue in the asynchronous AA protocol for chordal graphs with monophonic convexity of [37]. We describe this issue in detail in Section 6, and we provide an alternative solution in the network-agnostic model. The alternative protocol is obtained by adapting the AA protocol on cycle-free (chordal) semilattices (i.e., a particular case of chordal graphs) of the same paper [37], while incorporating insights from the protocol of [5] achieving *wait-free* AA on chordal graphs. Our AA protocol relies on our GTHR variant and provides network-agnostic resilience guarantees, provided that $n > \omega \cdot t_s + t_a$ where ω denotes the size of the largest clique in the input graph (which is the Helly number in this case [19, 25]).

1.2 Related work

CC and AA in the pure synchronous and pure asynchronous models. To the best of our knowledge, the problem of agreeing on a value in the honest inputs' convex hull was introduced for AA on $\mathbb{R}$ [17] (where $\omega = 2$). When considering this problem in the synchronous model, $t < n/3$ is tight when no cryptographic setup is allowed [17], and $t < n/2 = n/\omega$ is tight with cryptographic setup [23, 29]. In the asynchronous model, the optimal resilience threshold for AA is $t < n/3$ and was proven in [1]. The more general setting of $\mathbb{R}^D$ with straight-line convexity (where $\omega = D + 1$) was first considered in [31, 41] (see also the journal version [32]). Here, the bound $t < n/(D+2)$ is necessary and sufficient for asynchronous AA. Along with the multidimensional variant of AA, Vaidya and Garg [41] have introduced the CC problem on $\mathbb{R}^D$ and showed that the condition $t < n/(D + 1)$ is tight for achieving CC in the synchronous model. Tseng and Vaidya [40] later presented an asynchronous variant of CC resilient against crash failures with incorrect inputs, also on $\mathbb{R}^D$, where parties agree on a polytope in the convex hull of the honest parties' inputs as opposed to a single value.

We also note the works of [22, 36], which focus of achieving agreement *on an honest input*. This is a particular case of CC on a space with universe V where a subset's convex hull is the subset itself. Their necessary and sufficient conditions match the more general variants, as the Helly number of this convexity space is $\omega = \big|V\big|$: $t < n/\big|V\big|$ in the synchronous model [36], and $t < n/(\big|V\big| + 1)$ in the asynchronous one [22].

Nowak and Rybicki [37] generalized the problems of CC and AA to abstract convexity spaces. We partially answers an open question raised in [37] on whether there exists an input convexity space for which the optimal resilience threshold for AA depends on the Carathéodory number and not on the Helly number. Our tight conditions for CC imply that, at least for randomized protocols, the asynchronous resilience threshold is actually independent of the Carathéodory number and only depends on the Helly number instead. In addition, we identify a core issue in the deterministic algorithm of [37] for asynchronous AA on chordal graphs. A related line of work considers graph AA in the wait-free model

(where $t < n$ of the parties involved may crash) and its variants, primarily focusing on characterizing the families of graphs on which wait-free AA can be achieved [3,5,14,28,30].

CC and AA in the network-agnostic model. The problem of AA on real values has also been considered in the network-agnostic model in [23], where the condition $n > 2 \cdot t_s + t_a$ has been proven necessary and sufficient. For the $\mathbb{R}^D$ variant of AA, the condition $n > (D+1) \cdot t_s + t_a$ has been proven to be sufficient [24], but whether this condition is also necessary is still an open problem. Our work will build upon and extend some of the techniques of [23,24]. Concretely, our Gather protocol is obtained by making an adjustment to the network-agnostic Overlap All-to-All Broadcast primitive of [23, 24], while incorporating insights from asynchronous Gather protocols [2, 13]. In addition, we rely on similar insights on deriving *safe areas* in the honest parties' convex hulls to [24], and also of prior works in the asynchronous model such as [1, 31, 37, 41]. Previously known techniques and insights will be noted precisely in the following sections. As a summary, our paper will diverge from [23, 24] since: (i) we do not assume a particular input space: we consider abstract convexity spaces, and this requires us to provide generalized variants of lower bounds and safe area calculations; (ii) we focus on CC as opposed to AA, and achieving exact agreement requires us to design a stronger communication primitive: ACS. We also need to note that our feasibility result defines the first protocol in the network-agnostic model achieving an optimal resilience trade-off with a non-linear boundary (see Figure 1b). This gives a hint (but not yet an answer) on the question regarding necessary conditions in multidimensional AA left open in [24].

ACS in the network-agnostic model. As previously mentioned, the term ACS has been present in network-agnostic literature, as a building block for State-Machine Replication [4,10] and Multi-Party Computation [11]. We highlight an important distinction between prior constructions and ours. First, prior ACS variants would only provide as output *a set of values*, while our construction provides a common view defined as *a set of value-party pairs*. Second, when running in the synchronous model, the ACS protocols of [4,10,11] only need to ensure that *pre-agreement* is maintained: if all honest parties hold input v, the output set is $\{v\}$. For our CC protocol, the following properties will be crucial: (i) parties agree on the output set if the network is synchronous and t_s of the parties are corrupted (even without pre-agreement); (ii) roughly, each value's multiplicity is reflected in the output set (hence why the output set consists of value-party pairs); and, most importantly, (iii) if the network is synchronous, all honest values are guaranteed to be included (with multiplicities) in the parties' common view. Our ACS implementation will hence focus on this stronger definition, requiring us to diverge from the outline of previous ACS constructions for $n > 2 \cdot t_s + t_a$.

The concurrent work of [26], addressing Atomic Broadcast, also proposes a network-agnostic ACS implementation. While their protocol also relies on Gather and appears to achieve agreement regardless of the type of network, our ACS protocol is strictly stronger, as the protocol proposed by [26] provides the parties with a single value as output, and this value may be proposed by a corrupted party. This would prevent CC, but is sufficient for achieving Atomic Broadcast as in [26], since parties' values are *justified*.

While our ACS definition is stronger than previous network-agnostic variants, the protocol of [4] remains the state-of-art in terms of efficiency. The ACS protocol of [4] achieves an expected communication complexity of $O(n^2 \cdot \ell + n^3 \cdot \kappa)$ bits, where κ is the security parameter, and parties' inputs are represented as ℓ-bit strings (assuming threshold signatures). Even with threshold signatures, our protocol would incur an expected communication complexity of $O(n^3 \cdot \ell + n^4 \cdot \kappa)$, where ℓ denotes the universe elements'

size in bits.

2 Preliminaries

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

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

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 arbitrary intersections: whenever $A, B \in \mathcal{C}$, it also holds that $A \cap B \in \mathcal{C}$ (and the infinite analogue). 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 definition's requirements; 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 closed intervals of the real line.

A central notion 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, which is indeed convex by closure under intersections. 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 infimum and the supremum 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 $S = ex(S)$.

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. This is a non-trivial requirement; e.g., $\mathbb{R}^D$ with straight-line convexity or box convexity is **not** a convex geometry. An example of a convex geometry is a chordal graph endowed with monophonic convexity. These notions are further discussed in Section 6, and we provide a detailed discussion in Appendix A.

The Helly Number ω of a Convexity Space. The following seminal result in convexity theory concerns $\mathbb{R}^D$ with straight-line convexity.

Theorem 1 (Helly's Theorem)**.** *Consider a finite collection of 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 finite collection of 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 now wonder: "What about box convexity?" In that case, $D+1$ can be replaced by 2. For instance, this means that any finite collection of 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 known as the Helly number ω of the convexity space.

More generally, the *Helly number* ω of a convexity space $\mathcal{C}$ is the smallest number h such that any finite collection of convex sets out of which any h intersect has a non-empty intersection. It is useful to think in terms of the contrapositive: any finite collection of convex sets that do not intersect has a subcollection consisting of (at most) h sets that do not intersect. As a result, ω is equivalently the size of the largest collection of convex sets with an empty intersection such that any of its proper subcollections have a non-empty intersection. Notationally, say that 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; i.e., $\cap_{j=1}^{m} C_j = \varnothing$ and $\cap_{j \neq i} C_j \neq \varnothing$ for any $i \in [m]$. The *Helly number* ω of $\mathcal{C}$ is then the largest number h such that there exists an h-Helly family for $\mathcal{C}$. We will mostly work with this latter definition of the Helly number. Note that for some spaces, there will exist arbitrarily large Helly families, in which case the Helly number is undefined.[1]

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 monophonic convexity. Each party P starts with an input $v_{\mathsf{in}}^P \in V$ 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 byzantine parties. Hence, an agreement problem is defined by a collection of properties taking this into account: a validity condition, an agreement condition, and a termination condition. Write V_{in} and V_{out} for the set of inputs v_{in}^P and respectively outputs v_{out}^P of the *honest* parties P. A convex agreement problem has the following validity condition:

Convex Validity: $V_{\mathsf{out}} \subseteq \langle V_{\mathsf{in}} \rangle$ (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: $|V_{\mathsf{out}}| = 1$ (honest parties obtain the same output).

Given a metric *dist* on the universe V, one can alternatively define:

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

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.

[1] We do not 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.

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

Our work is concerned with two problems, CC and AA, defined below:

Definition 2 (Convex Consensus)**.** *A protocol* Π *is a* (t_s, t_a)*-secure* CC *protocol if it achieves Probabilistic Termination, Convex Validity and Exact Agreement when up to* t_s *parties are corrupted if it runs in a synchronous network, and when up to* t_a *parties are corrupted if running in an asynchronous network.*

Definition 3 (Approximate Agreement)**.** *A protocol* Π *is a* (t_s, t_a)*-secure* AA *protocol if it achieves Termination, Convex Validity and, for every predefined* $d > 0$,[2] *Distance-*d *Agreement when up to* t_s *parties are corrupted if it runs in a synchronous network, and when up to* t_a *parties are corrupted if running in an asynchronous network.*

3 Resilience Lower Bounds Using the Helly Number

In this section, we establish necessary conditions for achieving CC in the network-agnostic model. Concretely, we show that each of the following conditions is needed: $n > \omega \cdot t_s$, $n > 2 \cdot t_s + t_a$, and $n > \omega \cdot t_a + t_s$. Section 4 will show that these conditions are also sufficient.

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. These already imply that $n > \omega \cdot t_s$ and $n > (\omega + 1) \cdot t_a$ are required in the in the network-agnostic model. Afterward, we move towards conditions that are only required in the network-agnostic model. We note that a previously-known resilience bound [37, Theorem 13] given in terms of the Carathéodory number of the space is incompatible with our results: in general, there is no relation between the Carathéodory number and the Helly number. This bound turns out to be incorrect (detailed discussion in Appendix B.2).

Before formally showing our lower bounds, we introduce the notion of *adversarial families*, which will enable us to give general scenario-based arguments [36]. Roughly, these are families of pairwise-disjoint sets such that if the honest parties start with inputs from these sets, then Convex Validity forces them to output values from these sets, breaking Exact Agreement. The formal definition follows; see Figure 2 for an example.

Definition 4. *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* $i \in [m]$.[3]

Note that the pairwise-disjoint condition is equivalent to $\cap_{\ell=1}^{m} \langle A \setminus A_\ell \rangle = \varnothing$.[4] We add that, in the example of Figure 2, sets A_i are singletons, but requiring this would strictly decrease the power of adversarial sets in general. The following technical lemma, and the two following it, will be the main tools used to get impossibility results. The techniques used in its proof, supplied in Appendix B.1, are similar in spirit to the proofs for $\mathbb{R}^D$ in [32].

[2] $d \geq 1$ for standard graph distance, as there $d < 1$ is equivalent to $d = 0$.

[3] 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.

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

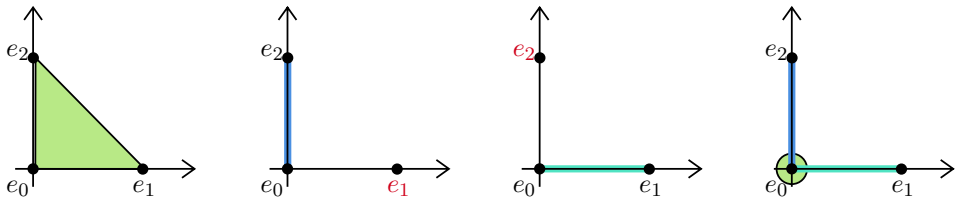


Figure 2: Consider $\mathbb{R}^2$ with straight-line convexity and the vectors $e_0, e_1, e_2 = (0,0), (1,0), (0,1)$, illustrated in the first figure. Define $A_i = \{e_i\}$ for $i = 0, 1, 2$ to be singleton sets for the previous (and hence convex sets). Then, $\mathcal{A} = \{A_0, A_1, A_2\}$ is a 3-adversarial family. To see why, first note that by definition $A = \{e_0, e_1, e_2\}$. By symmetry, it suffices to check the condition for $i = 0$; i.e., to show that $A_0 = \langle A \setminus A_1 \rangle \cap \langle A \setminus A_2 \rangle$. Simplifying, this amounts to $A_0 = \langle \{e_0, e_2\} \rangle \cap \langle \{e_0, e_1\} \rangle$. The second figure illustrates $\langle \{e_0, e_2\} \rangle$, the third illustrates $\langle \{e_0, e_1\} \rangle$, and the forth $\langle \{e_0, e_2\} \rangle \cap \langle \{e_0, e_1\} \rangle$, which is precisely A_0, as required. The illustrations come from [24].

Lemma 5. *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 Validity and (Probabilistic) Termination has 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 $v^i_{\mathsf{out}} \in A_i$.*

The following two technical lemmas give similar guarantees, but in the network-agnostic model. The proof of the first is similar to that for $\mathbb{R}$ in [23], while that of the second is an extension of the asynchronous part of Lemma 5. The proofs are included in Appendix B.1.

Lemma 6. *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 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 $v^1_{\mathsf{out}} \in A_1$ and $v^2_{\mathsf{out}} \in A_2$.*

Lemma 7. *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 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 has 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 $v^i_{\mathsf{out}} \in A_i$.*

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

Lemma 8. *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 5 and 7, 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 9. *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, 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 8. □

We now leverage Lemmas 5, 8 and 9 to get the following result, generalizing those in [37] by removing the strong requirement of a convex geometry.

Theorem 10. *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 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 8, there is an ω-adversarial family for $\mathcal{C}$. Since $m \leq \omega$, using Lemma 9, let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be an m-adversarial family for $\mathcal{C}$. Consider a protocol Π satisfying Convex 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. □

By leveraging Lemmas 6, 7, 8 and 9, we similarly get the following result. The proof is included in Appendix B.1. There, we also show that adversarial families are useful more generally by swiftly recovering previously-known lower bounds for AA in $\mathbb{R}^D$.

Theorem 11. *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, no (deterministic or randomized) n-party protocol satisfying Convex 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.*

4 Achieving Optimal-Resilience Convex Consensus

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

Theorem 12. *If $t_a \leq t_s$ and $n > \max(\omega \cdot t_s, 2 \cdot t_s + t_a, \omega \cdot t_a + t_s)$, there is a protocol achieving (t_s, t_a)-secure CC assuming PKI. The protocol has expected round complexity $O(1)$. If ℓ denotes the universe elements' size in bits and κ is the security parameter, its expected communication complexity is $O(n^3 \cdot \ell + n^4 \cdot \kappa)$ bits. If threshold signatures are available, the expected communication complexity reduces to $O(n^3 \cdot \ell + n^3 \cdot \kappa)$ bits.*

To set up the intuition for our construction, we recall the outline of the synchronous protocol for $\mathbb{R}^D$ with straight-line convexity of [41]. The synchronous model offers powerful communication primitives (i.e., Synchronous Broadcast [18]), enabling the parties to distribute their inputs and obtain an *identical view* of the inputs. This view consists of a set of value-sender pairs, out of which $n - t_s$ correspond to honest parties. Then, the parties derive a *safe area* inside the honest inputs' convex hull by intersecting the convex hulls of all subsets of $n - t_s$ values received, as defined below. We extend the convex hull operator to value-sender sets straightforwardly: ignore party identities and take the convex hull of the values.

Definition 13 (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 valid. Hence, parties may output any such value chosen by some deterministic criterion.

Technical assumptions. Implementing such a protocol requires mild assumptions about $\mathcal{C}$: it should be possible to (i) store elements from V and to send them in messages; (ii) compute and intersect convex hulls; (iii) deterministically select a point from the safe area. *Only to express communication complexity bounds, we will also need that $|V| \leq 2^\ell$ for some ℓ.*

Identical views in asynchrony. Building towards our solution achieving network-agnostic guarantees, we first identify the challenges posed by translating the outline above to the purely asynchronous model (where $t_s = t_a$ and $n > (\omega + 1) \cdot t_a$). Given a primitive that provides the parties with an identical view 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 are corrupted. Then, honest parties derive the safe area $safe_{t_a}(\mathcal{M})$ inside their inputs' convex hull, and afterwards take a deterministic decision to obtain the same output.

Achieving the required identical view deterministically is impossible in the asynchronous model [21], but randomization allows for a simple solution by employing a primitive introduced in [7]. This primitive archives *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 [8, 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.

Exploiting the advantages of synchrony. While the standard definition of ACS provides identical views when the network is asynchronous, the synchronous model still has a crucial advantage that needs to be used to achieve higher resilience. 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. In contrast, in the asynchronous setting, t_a corrupted parties may *replace* t_a honest parties: the honest parties' messages get delayed for sufficiently long, while the t_a corrupted parties follow the protocol correctly, but with inputs of their choice. To match the condition $n > \max(\omega \cdot t_s, 2 \cdot t_s + t_a, \omega \cdot t_a + t_s)$ in the network-agnostic model, we hence need an additional property in a synchronous network: all honest values must be included in the output set. Consequently, we propose the following enriched definition:

Definition 14 (Agreement on a Core-Set)**.** *Let* Π *be a protocol where every party* P *holds an input* v_P *and outputs 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*: If* P *and* P' *are honest,* $(v, P'') \in \mathcal{M}_P$ *and* $(v', P'') \in \mathcal{M}_{P'}$*, then* $v = v'$*.*[5]
T*-Output Size: If an honest party* P *outputs* $\mathcal{M}_P$*, then* $\left|\mathcal{M}_P\right| \geq n - T$*.*
Honest Core*: If an honest party* P *outputs* $\mathcal{M}_P$*, then* $(v_{P'}, P') \in \mathcal{M}_P$ *for every honest* P'*.*

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

- *Validity, Consistency, Exact Agreement, Honest Core, Probabilistic Termination when running in a synchronous network where at most* t_s *parties are corrupted;*
- *Validity, Consistency, Exact Agreement,* t_s*-Output Size,*[6] *Probabilistic Termination when running in an asynchronous network where at most* t_a *parties are corrupted.*

Section 5 describes a protocol Π_{ACS} realizing the theorem below (given PKI). We will also describe a protocol for $t_a \leq t_s < n/3$ without PKI, which is suitable for the case $\omega \geq 3$.

Theorem 15. *If* $n > 2 \cdot t_s + t_a$ *and* $t_a \leq t_s$*, there is a* (t_s, t_a)*-secure* ACS *protocol* Π_{ACS} *(assuming PKI). The protocol has expected round complexity* $O(1)$*. If* ℓ *denotes the universe elements' size in bits and* κ *is the security parameter, its expected communication complexity is* $O(n^3 \cdot \ell + n^4 \cdot \kappa)$ *bits. If threshold signatures are available, the expected communication complexity reduces to bits.*

If parties distribute their values via Π_{ACS}, they agree on a 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, at most k of these values are corrupted due to Honest Core. To cover both cases, parties locally compute their safe areas as $S := \mathit{safe}_{\max(k,t_a)}(\mathcal{M})$ and deterministically decide on an output $v_{\mathsf{out}} \in S$. Note that, by definition, S is indeed inside the honest inputs' convex hull. In addition, since the input space has Helly number ω and $n > \max(\omega \cdot t_s, \omega \cdot t_a + t_s)$, the safe area can be shown to be non-empty, so such v_{out} can be chosen. The proof of this result, stated below, is contained in Appendix D.

Lemma 16. *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,* $\mathit{safe}_{\max(k,t_a)}(\mathcal{M}) \neq \varnothing$*.*

We may now conclude the section by providing the formal code of our CC protocol.

[5] Note that this implies that each party appears at most once as a sender in $\mathcal{M}_P$.
[6] This is intentional: we do not require the stronger property of t_a-Output Size.

Protocol Π_{CC}

Code for party P with input v_{in}

1: Join Π_{ACS} with input v_{in}. Upon obtaining output $\mathcal{M}$:
2: $k := \left|\mathcal{M}\right| - (n - t_s)$; $S := safe_{\max(k,t_a)}(\mathcal{M})$.
3: Choose $v_{\mathsf{out}} \in S$ according to a public, predetermined, deterministic rule.
4: Output v_{out}.

Proof of Theorem 12. Π_{ACS} provides the parties with the same set $\mathcal{M}$ of $n - t_s + k$ value-sender pairs, with $0 \leq k \leq t_s$. $\mathcal{M}$ contains at most $\max(k, t_a)$ values from byzantine parties: in a synchronous network, this holds 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 honest inputs. By definition, $M_H \in restrict_{\max(k,t_a)}(\mathcal{M})$, so $S \subseteq \langle M_H \rangle$, i.e., honest parties obtain a safe area S that is included in their inputs' convex hull. Lemma 16 ensures that S is non-empty, and therefore honest parties agree on the same value v_{out} in the honest inputs' convex hull. Consequently, Π_{CC} is (t_s, t_a)-secure CC protocol. □

5 Agreement on a Core-Set

In this section, we describe the protocol realizing Theorem 15. We first focus on the easier case $t_a \leq t_s < n/3$: we begin by describing the asynchronous protocol of [8] (as presented in [33]), which fulfills all properties outlined in Definition 14, except for Honest Core in a synchronous network. We adapt this protocol to satisfy Honest Core as well, obtaining (t_s, t_a)-secure ACS when $t_a \leq t_s < n/3$. Note that our CC lower bound of $n > \max(\omega \cdot t_s, 2 \cdot t_s + t_a, \omega \cdot t_a + t_s)$ implies $t_a \leq t_s < n/3$ for $\omega \geq 3$, so the adapted protocol suffices if the latter is true. For $\omega = 2$, however, the adapted protocol is insufficient. Consequently, we then move on to the case $n > 2 \cdot t_s + t_a$, for which we present a novel construction.

We will utilize Reliable Broadcast (rBC) and Byzantine Agreement (BA) as building blocks. We include their definitions in the network-agnostic model below.

Definition 17 (Reliable Broadcast)**.** *Let* Π *denote a protocol where a designated party* S *(the sender) holds a value* v_S*, and every party* P *may output a value* v_P*. Consider the following properties:*
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* $v_P = v_{P'}$*.*
Honest Termination*: If* S *is honest, all honest parties obtain outputs.*
Conditional Termination*: If an honest party* P *outputs, all honest parties obtain outputs.*

We say that Π *is a* (t_s, t_a)*-secure* rBC *protocol if it achieves Validity, Consistency, Honest Termination, and Conditional Termination up to* t_s *corruptions if it runs in a synchronous network, and up to* t_a *corruptions if it runs in an asynchronous network.*

Definition 18 (Byzantine Agreement)**.** *Let* Π *be a protocol where every party* P *holds a bit as input and may output a bit, and consider the following property:*
Weak Validity*: If all honest parties hold input* b*, no honest party outputs* $b' \neq b$*.*

Then, Π *is a* (t_s, t_a)*-secure* BA *protocol if it achieves Weak Validity, Exact Agreement, and Probabilistic Termination up to* t_s *corruptions if it runs in a synchronous network, and up to* t_a *corruptions if it runs in an asynchronous network.*

The asynchronous ACS protocol of [8]. We describe the protocol of [8], following the variant presented in [33]. We denote this protocol by $\Pi_{a\mathsf{ACS}}$. We highlight once again that $\Pi_{a\mathsf{ACS}}$ is designed for the purely asynchronous model: it assumes a single threshold $t < n/3$ and seeks properties that hold under asynchrony with at most t corrupted parties. To use this protocol in the hybrid model with $t_a \leq t_s < n/3$, one can set $t := t_s$. When this is done, $\Pi_{a\mathsf{ACS}}$ achieves all properties of being a (t_s, t_a)-secure ACS protocol as per Definition 14 (in fact, even (t_s, t_s)-secure), with the exception of Honest Core under synchrony.

In $\Pi_{a\mathsf{ACS}}$, every party first distributes its input value via rBC. Concretely, this is done using Bracha's protocol [12], denoted by $\Pi_{a\mathsf{rBC}}$, which achieves (t, t)-secure rBC for $t < n/3$. Due to asynchronous communication, $\Pi_{a\mathsf{rBC}}$ only guarantees that parties receive a value if the sender is honest. As a result, at least $n - t$ values eventually get delivered to all parties, but these values might still be received at vastly different times, leading to inconsistent views if parties were to output the first $n - t$ values they received.

Then, to decide on a common output set, the parties will use BA to agree on which values they received and should be included in the output set. We utilize the protocol $\Pi_{a\mathsf{BA}}$ of Mostefaoui et al. [35], which achieves (t, t)-secure BA when $t < n/3$. There will be n parallel invocations of $\Pi_{a\mathsf{BA}}$ — one for each party. When a party P receives a value v from P' via $\Pi_{a\mathsf{rBC}}$, it joins the $\Pi_{a\mathsf{BA}}$ invocation corresponding to P' with input 1. Semantically, if the $\Pi_{a\mathsf{BA}}$ invocation of a party P' returns output 1, then the value distributed by P' via $\Pi_{a\mathsf{rBC}}$ should be included in the output set. Note that, when this happens, the Weak Validity property of $\Pi_{a\mathsf{BA}}$ ensures that at least one honest party P has joined the $\Pi_{a\mathsf{BA}}$ invocation for party P' with input 1. That is, P has received a value v from P', and $\Pi_{a\mathsf{rBC}}$'s Conditional Termination ensures that all honest parties eventually receive the same value v from P'. In simple terms, the value of P' is *worth waiting for*, and parties wait until they receive it before completing the protocol.

Eventually, at least $n-t$ invocations result in output 1 (suppose not, then, since at least $n-t$ honest values are eventually delivered to all honest parties, the honest parties will all join at least $n - t$ invocations of $\Pi_{a\mathsf{BA}}$ with input 1, guaranteeing that those invocations terminate with output 1). To complete the protocol, we still need that all $\Pi_{a\mathsf{BA}}$ invocations complete. Then, whenever some party P observes $n-t$ invocations completing with output 1, it should join all remaining invocations with input 0. Hence, once all honest parties join all $\Pi_{a\mathsf{BA}}$ invocations, all invocations are guaranteed to terminate. Upon observing that all $\Pi_{a\mathsf{BA}}$ invocations have terminated, each party P outputs the set of (at least $n - t$) value-sender pairs corresponding to the $\Pi_{a\mathsf{BA}}$ invocations that returned output 1 (after waiting to receive any outstanding values through $\Pi_{a\mathsf{rBC}}$). This way, the protocol ensures that all honest parties obtain an identical view, achieving all properties required by Definition 14 for asynchronous networks. Note that the Validity property follows immediately from the guarantees of $\Pi_{a\mathsf{rBC}}$.

Adjustments for the network-agnostic model. We now return to the network-agnostic model and adapt $\Pi_{a\mathsf{ACS}}$ to achieve our ACS definition for the parameter range $t_a \leq t_s < n/3$. As previously established, $\Pi_{a\mathsf{ACS}}$ already satisfies all properties required to be a (t_s, t_a)-secure ACS protocol by setting $t := t_s$, except for Honest Core in a synchronous network.

To see why the Honest Core property does not hold, consider a scenario where the network is synchronous, and the t_s corrupted parties follow the protocol correctly with inputs of their choice. All messages are delivered immediately, except for the messages sent by t_s of the honest parties: these are delivered exactly after Δ time. The remaining honest parties complete the protocol before time Δ, and the values of t_s honest parties are

missing from the output set. To prevent this, we impose a waiting time to ensure that, if the network is synchronous, all honest parties' messages are received. Running $\Pi_{a\mathsf{rBC}}$ in the synchronous model guarantees additional properties, established in [24]: when an honest party sends a value via $\Pi_{a\mathsf{rBC}}$, all parties receive this output within $c_{a\mathsf{rBC}} \cdot \Delta$ time, where $c_{a\mathsf{rBC}} := 3$. Then, to achieve Honest Core, we impose a waiting period of at least $c_{a\mathsf{rBC}} \cdot \Delta$ time before allowing the parties to participate in $\Pi_{a\mathsf{BA}}$ invocations with input 0. This way, if the network is synchronous, all honest parties join every honest party's $\Pi_{a\mathsf{BA}}$ invocation with input 1, and these invocations return 1. Hence, all honest parties' values are included in the output set.

We include below the adapted $\Pi_{a\mathsf{ACS}}$, which achieves (t_s, t_a)-secure ACS for $t_a \leq t_s < n/3$. In fact, it achieves (t_s, t_s)-secure ACS: it is an asynchronous protocol with an added waiting period to ensure the Honest Core property under synchrony. The protocol incurs expected constant round complexity, and expected communication complexity $O(n^3 \cdot \ell)$, where ℓ denotes the universe elements' size. For the formal analysis, see Appendix C.2.

Adapted protocol $\Pi_{a\mathsf{ACS}}$ [8, 33]

Code for party P with input v ($\Pi_{a\mathsf{rBC}}$ and $\Pi_{a\mathsf{BA}}$ invocations use $t := t_s$)

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

Achieving ACS when $n > 2 \cdot t_s + t_a$. Finally, we describe our solution for the general case. We utilize building blocks designed specifically for this setting: the (t_s, t_a)-secure rBC protocol Π_{rBC} of Momose and Ren [34], and the (t_s, t_a)-secure BA protocol Π_{BA} of Deligios, Hirt and Liu-Zhang [15, Corollary 2]. We add that these protocols assume and need PKI.

While the $t_a \leq t_s < n/3$ setting enabled a solution based on tweaks to previously known protocols, the $n > 2 \cdot t_s + t_a$ case introduces different challenges. In particular, one detail we omitted when presenting $\Pi_{a\mathsf{ACS}}$ concerns protocol composability. Namely, Definition 18 of BA protocols assumes that honest parties join the protocol simultaneously in a synchronous network. In the outline of [8] and in $\Pi_{a\mathsf{ACS}}$, this is, in fact, not the case. However, when $t_s = t_a$, this assumption is not strictly required because the asynchronous guarantees step in whenever honest parties are unable to join simultaneously. In contrast, when $t_s > t_a$, (t_s, t_a)-secure BA does not have to provide any guarantees in a synchronous network with t_s corruptions if honest parties are unable to join simultaneously. This is especially problematic when $t_s \geq n/3$ because (t_s, t_s)-secure BA protocols do not exist. The Conditional Termination property of Π_{rBC} is too weak to ensure that honest parties are ready to join the BA invocations simultaneously when the network is synchronous — a challenge that we need to overcome.

Our goal is then to allow the parties to join each invocation of Π_{BA} at the same time when the network is synchronous — this refers both to invocations where parties

join with input 1, and to invocations where parties join with input 0. We will do so by enabling the parties to decide their input bit for each invocation of Π_{BA} independently of the other invocations' outcomes. On top of this property, we still need to guarantee t_s-Output Size when the network is asynchronous, which amounts to ensuring that at least $n - t_s$ invocations of Π_{BA} return 1. Moreover, when the network is synchronous, the Π_{BA} invocations of honest parties have to output 1 to ensure the Honest Core property.

To enable the honest parties to safely decide each input bit for the Π_{BA} invocations independently, realizing the previous desiderata, we introduce a network-agnostic version of a primitive known as *Gather* (GTHR) [2,13]. This is a slightly weaker, but deterministic variant of ACS: GTHR relaxes Exact Agreement by only requiring that honest parties' output sets have at least $n - t_s$ values in common. Our definition of GTHR, provided below, additionally requires the previous Honest Core property to hold under synchrony. Moreover, we require that honest parties obtain outputs simultaneously if the network is synchronous.

Definition 19 (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 14.*

T*-Common Core: If all honest parties obtain outputs, then $\left|\bigcap_{P \text{ honest}} \mathcal{M}_P\right| \geq n - T$.*

Simultaneous Termination*: The honest parties obtain outputs simultaneously.*

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

- *Validity, Consistency, Honest Core 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 Appendix C.3, we provide a construction achieving our GTHR definition, as stated below. This is obtained by adding one step to the network-agnostic Overlap All-to-All Broadcast protocol of [23], while using insights from asynchronous variants of GTHR [2].

Theorem 20. *If $n > 2 \cdot t_s + t_a$ and $t_a \leq t_s$, there is a (t_s, t_a)-secure GTHR protocol Π_{GTHR} (assuming PKI). The protocol has round complexity $O(1)$. If ℓ denotes the universe elements' size in bits and κ is the security parameter, it achieves a communication complexity of $O(n^3 \cdot \ell + n^4 \cdot \kappa)$ bits (can be reduced to $O(n^3 \cdot \ell + n^3 \cdot \kappa)$ with threshold signatures).*

Then, our ACS protocol proceeds as follows: the parties distribute their inputs using the Π_{GTHR} protocol realizing Theorem 20. When obtaining outputs $\mathcal{M}_{\mathsf{GTHR}}$ (potentially different for different parties), each party P joins the n invocations of Π_{BA}. 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} now hold. Since the value-sender pairs $\mathcal{M}_{\mathsf{GTHR}}$ obtained by the honest parties intersect in $n - t_s$ pairs, at least $n - t_s$ of the Π_{BA} invocations result in output 1. In addition, if the network is synchronous, Π_{GTHR}'s Honest Core ensures that every invocation corresponding to an honest party returns 1.

An important and subtle caveat in the above is 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}. Although Π_{BA} ensures that at least one honest party P has received a value from P' via Π_{GTHR}, this may not be the case for all honest parties. To address this, we state an additional property provided by our implementation of Π_{GTHR}: if parties wait for sufficiently long even after obtaining outputs via Π_{GTHR}, they will (consistently) receive

the missing values as well. This is because, internally to Π_{GTHR}, parties distribute their inputs via Π_{rBC}.

Observation 21. *Let P and P' denote two honest parties, and let $\mathcal{M}$ and $\mathcal{M}'$ denote their internal message sets in Π_{GTHR}. If $(v, P'') \in \mathcal{M}$, then, eventually, $(v, P'') \in \mathcal{M}'$ as well.*

We may now present our ACS protocol. We defer the analysis to Appendix C.4.

Protocol Π_{ACS}

Code for party P with input v

Join Π_{GTHR} with input v. When receiving output $\mathcal{M}_{\mathsf{GTHR}}$ from Π_{GTHR}:
    For each party P':
        $b_{P'} := 1$ if $(v', P') \in \mathcal{M}_{\mathsf{GTHR}}$ for some v' and 0 otherwise.
        Join the Π_{BA} invocation for P' with input $b_{P'}$.
    When receiving v' from P' via Π_{rBC} [initiated in Π_{GTHR}], add (v', P') to $\mathcal{M}_{\mathsf{GTHR}}$.
    When obtaining outputs in all invocations of Π_{BA}:
        $\mathcal{P} :=$ the set of parties whose Π_{BA} invocations returned output 1.
        When $(v', P') \in \mathcal{M}_{\mathsf{GTHR}}$ for every $P' \in \mathcal{P}$:
            Output $\mathcal{M} :=$ the set of pairs $(v', P') \in \mathcal{M}_{\mathsf{GTHR}}$ with $P' \in \mathcal{P}$.

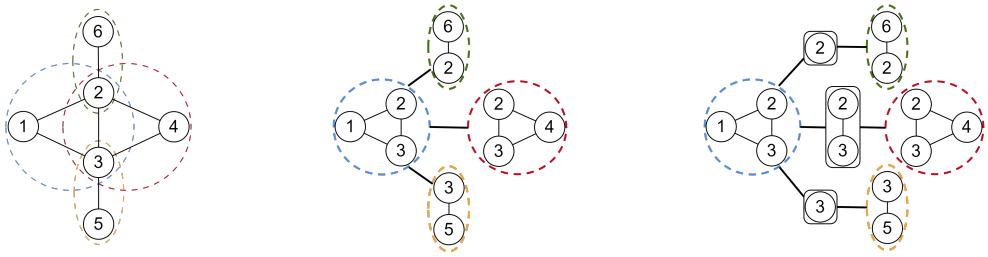
6 Approximate Agreement on Chordal Graphs

We investigate the previously known deterministic protocol that efficiently achieves chordal graph AA with monophonic path convexity [37, Section 4.2], finding that it is sadly incorrect. Before diving into the algorithm itself and presenting our solution tailored to the network-agnostic setting, we establish a few preliminaries, which we keep minimal for understanding this section, deferring further details and intuition to Appendix A.

Monophonic convexity. We first describe the convexity space we are concerned with. Unless stated otherwise, all graphs considered are finite, undirected, connected and simple. Given a graph $G = (V, E)$, a subset $C \subseteq V$ is *monophonically convex* if, for any two vertices $u, v \in C$, and any induced path P from u to v in G, all vertices in P are in C. Induced paths (also called chordless paths) are paths with no short-circuit edges. Note that these are not necessarily shortest paths. Convex hulls are then defined as usual: the intersection of all convex sets containing the set. Unless stated otherwise, in this section, by "convex" we mean "monophonically convex." For this convexity notion, the Helly number ω equals the size of the largest clique in G [19, 25]. We also point out the following correspondence between cliques and free sets, proven in Appendix A, which will be crucial for designing our AA protocol.

Lemma 22. *Let $G = (V, E)$ be a graph and $S \subseteq V$ be a subset of its vertices. Then, under monophonic convexity, S is a free set if and only if S induces a clique in G.*

Chordal graphs. 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 definitions; most notably, they are the graphs that have a simplicial vertex whose removal yields another chordal graph. Reasoning inductively, they are hence also the graphs admitting a *perfect elimination order*, which is a total order $\succ$ on V such that every $v \in V$ 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) 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.

The AA protocol of [37]. The protocol of [37, 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, the protocol 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 [37, 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 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 [37, 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 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: $S_T := safe_t^T(\mathcal{M})$; $b_{\mathsf{it}} = center(S)$; $S_G := safe_t^G(\mathcal{M})$. Select $v_{\mathsf{it}} \in S_G \cap b_{\mathsf{it}}$ arbitrarily.
5: **end for**
6: 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 cannot 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 [37, Lemma 11 in the full version [38]]. The core of the proof hinges on an argument showing that there is some vertex $u \in b_{\mathsf{it}}$ that is, roughly speaking, covered by many paths between the parties' values $v_{\mathsf{it}-1}$. More precisely, there are at least $t+1$ pairs of parties from which P has received values in iteration it, such that the two values $v_{\mathsf{it}-1}$ in each pair have an induced path in G between them that passes through u. The proof of this fact is correct [37, Lemma 10 in the full version [38]]. 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 the lemma seems unlikely to hold.

Additionally, 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 hybrid AA protocol. We now describe an AA protocol for chordal graphs. We note that [37] (Section 3) additionally includes an (asynchronous) AA protocol template that is parameterized by a function ϕ, meant to be instantiated depending on the convexity space. Our protocol matches this template but is tailored to the network-agnostic model.

Our protocol does not directly rely on the clique tree decomposition. This simplifies notation when it comes to convex hulls, as all convex hulls are now on G. Following the general outline of AA protocols [1, 17, 37], our protocol proceeds 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 $S := \mathit{safe}_{\max(k,t_a)}(\mathcal{M})$, 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. For this computation, we use a variant of the update rule of [5] (Section 7.2), which in their case facilitates AA on chordal graphs in the wait-free model: 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 extreme point (which is indeed possible by Lemma 22). 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 20. This is shown with the help of the lemma below, which is proven in Appendix D.

Lemma 23. *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 $\bigcap_{i=1}^K \mathit{safe}_{\max(k_i,t_a)}(\mathcal{M}_i) \neq \varnothing$.*

We present our protocol Π_{Chordal} and its guarantees below. Although our protocol Π_{Chordal} utilizes techniques from [5, 37], the proof of Theorem 24 requires additional insights. Note that even though our protocol uses a similar value computation rule to that of [5], the protocol of [5] is designed in the wait-free atomic snapshot model. This model ensures the strong property that, for any two parties, one's view is a subset of the other. Moreover, it is only concerned with crashes as opposed to byzantine failures, which drops the need of computing safe areas altogether. Hence, in the message-passing model with byzantine failures, proving correctness requires arguments relying on weaker properties in comparison to [5]. In addition, arguments for asynchronous AA protocols do not directly apply to the network-agnostic setting: the safe areas defined in the network-agnostic model lose a natural monotonicity property (namely, it is no longer the case that the more values you receive, the larger your safe area is). Due to this missing property, showing that the honest parties' safe areas intersect and hence their values get closer in each iteration relies on more elaborate arguments. For the formal proof, see Appendix E.

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: $k := \left|\mathcal{M}\right| - (n - t_s)$; $S := safe_{\max(k,t_a)}(\mathcal{M})$.
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.

Theorem 24. *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)-secure deterministic protocol achieving Monophonic Convex Validity, Termination and Agreement within Graph Distance 1.*

7 Conclusions

We investigated the necessary and sufficient conditions for achieving CC in the network-agnostic model, providing a necessary and sufficient condition for solvability. We have seen that, for any convexity space with Helly number ω, achieving CC, or, more precisely, Convex Hull 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 networks. In the network-agnostic 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 (t_s, t_a)-secure CC protocol Π_{CC} by making use of randomization, which can be seen to be necessary due to the FLP result [21], and assuming PKI only for the particular case $\omega = 2$ (where cryptographic setup is needed when $t_s \geq n/3$ [27]).

In the process, we proposed two communication primitives for the network-agnostic model, which may be of independent interest. These are variants of ACS and GTHR, which allow each party to distribute its input so that parties obtain consistent views on the original inputs. These stronger properties enabled us to ensure the synchronous resilience guarantees of CC in the network-agnostic 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, [11] uses a less general form of ACS).

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] Manuel Alcántara, Armando Castañeda, David Flores-Peñaloza, and Sergio Rajsbaum. The topology of look-compute-move robot wait-free algorithms with hard termination. *Distributed Computing*, 32(3):235–255, 2019. doi:10.1007/s00446-018-0345-3.

[4] Andreea B. Alexandru, Erica Blum, Jonathan Katz, and Julian Loss. State machine replication under changing network conditions. In *Advances in Cryptology – ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part I*, page 681–710, Berlin, Heidelberg, 2023. Springer-Verlag. doi:10.1007/978-3-031-22963-3_23.

[5] 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.

[6] 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.

[7] Michael Ben-Or, Ran Canetti, and Oded Goldreich. Asynchronous secure computation. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, page 52–61, New York, NY, USA, 1993. Association for Computing Machinery. doi:10.1145/167088.167109.

[8] Michael Ben-Or, Boaz Kelmer, and Tal Rabin. Asynchronous secure computations with optimal resilience (extended abstract). In *Proceedings of the Thirteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '94, page 183–192, New York, NY, USA, 1994. Association for Computing Machinery. doi:10.1145/197917.198088.

[9] Erica Blum, Jonathan Katz, and Julian Loss. Synchronous consensus with optimal asynchronous fallback guarantees. In *Theory of Cryptography*, volume 11891 of *Lecture Notes in Computer Science*, pages 131–150, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-36030-6_6.

[10] 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.

[11] 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.

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

[13] 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.

[14] Armando Castañeda, Sergio Rajsbaum, and Matthieu Roy. Convergence and covering on graphs for wait-free robots. *Journal of the Brazilian Computer Society*, 24(1):1, Jan 2018. doi:10.1186/s13173-017-0065-8.

[15] Giovanni Deligios, Martin Hirt, and Chen-Da Liu-Zhang. Round-efficient byzantine agreement and multi-party computation with asynchronous fallback. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography*, pages 623–653, Cham, 2021. Springer International Publishing.

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

[17] 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.

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

[19] Pierre Duchet. Convex sets in graphs, ii. minimal path convexity. *Journal of Combinatorial Theory, Series B*, 44(3):307–316, 1988. URL: https://www.sciencedirect.com/science/article/pii/0095895688900391, doi:10.1016/0095-8956(88)90039-1.

[20] 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.

[21] 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.

[22] Matthias Fitzi and Juan A. Garay. Efficient player-optimal protocols for strong and differential consensus. In Elizabeth Borowsky and Sergio Rajsbaum, editors, *22nd ACM PODC*, pages 211–220. ACM, July 2003. doi:10.1145/872035.872066.

[23] 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.

[24] Diana Ghinea, Chen-Da Liu-Zhang, and Roger Wattenhofer. Multidimensional approximate agreement with asynchronous fallback. In *Proceedings of the 35th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '23, page 141–151, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3558481.3591105.

[25] Robert E. Jamison and Richard Nowakowski. A helly theorem for convexity in graphs. *Discrete Mathematics*, 51(1):35–39, 1984. URL: https://www.sciencedirect.com/science/article/pii/0012365X84900219, doi:10.1016/0012-365X(84)90021-9.

[26] Simon Holmgaard Kamp and Jesper Buus Nielsen. Byzantine agreement decomposed: Honest majority asynchronous atomic broadcast from reliable broadcast. Cryptology ePrint Archive, Paper 2023/1738, 2023. https://eprint.iacr.org/2023/1738. URL: https://eprint.iacr.org/2023/1738.

[27] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[28] 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.

[29] Christoph Lenzen and Julian Loss. Optimal clock synchronization with signatures. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing*, PODC'22, page 440–449, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519270.3538444.

[30] Shihao Liu. The Impossibility of Approximate Agreement on a Larger Class of Graphs. In Eshcar Hillel, Roberto Palmieri, and Etienne Rivière, editors, *26th International Conference on Principles of Distributed Systems (OPODIS 2022)*, volume 253 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.OPODIS.2022.22, doi:10.4230/LIPIcs.OPODIS.2022.22.

[31] 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.

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

[33] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. The honey badger of BFT protocols. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 31–42. ACM Press, October 2016. doi:10.1145/2976749.2978399.

[34] 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.

[35] Achour Mostefaoui, Hamouma Moumen, and Michel Raynal. Signature-free asynchronous byzantine consensus with $t < n/3$ and $o(n^2)$ messages. In *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*, PODC '14, page 2–9, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2611462.2611468.

[36] Gil Neiger. Distributed consensus revisited. *Information Processing Letters*, 49(4):195–201, 1994. URL: https://www.sciencedirect.com/science/article/pii/0020019094900116, doi:10.1016/0020-0190(94)90011-6.

[37] 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.

[38] Thomas Nowak and Joel Rybicki. Byzantine approximate agreement on graphs, 2019. arXiv:1908.02743.

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

[40] Lewis Tseng and Nitin H. Vaidya. Asynchronous convex hull consensus in the presence of crash faults. In Magnús M. Halldórsson and Shlomi Dolev, editors, *33rd ACM PODC*, pages 396–405. ACM, July 2014. doi:10.1145/2611462.2611470.

[41] 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.

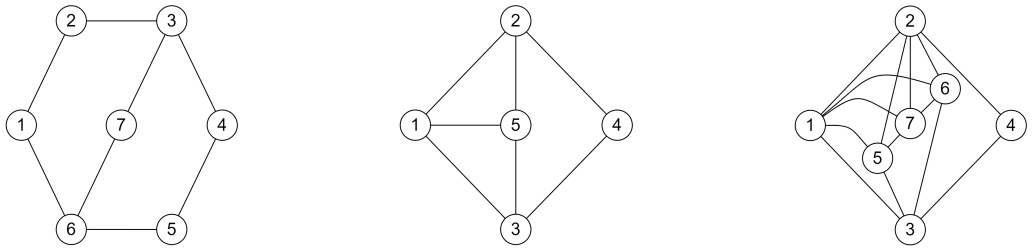
Appendix

A Preliminaries: Additional Details

A.1 Graph 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 4a, 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 4b 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 4c, for both convexity notions the set $S = \{1, 4\}$ would evolve as follows: $\{1, 4\} \to \{1, \ldots, 4\} \to \{1, \ldots, 6\} \to \{1, \ldots, 7\} = \langle\{1, 4\}\rangle$.



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

Figure 4: 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.

An important observation is that free sets correspond to cliques for our two graph

convexity notions. This is stated and proven in the following lemma. Note that this trivially implies Lemma 22.

Lemma 25. *Let $G = (V, E)$ be a graph and $S \subseteq V$ be a subset of its vertices. Then, under both geodesic and 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. □

A.2 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 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 [20], 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. 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.

B Resilience Lower Bounds

B.1 Impossibility Results Using Adversarial Families

Lemma 5. *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 Validity and (Probabilistic) Termination has 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 $v^i_{\mathsf{out}} \in A_i$.*

Proof. First, consider the synchronous case. Write $A = \cup\mathcal{A}$ and consider a protocol Π satisfying Convex 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 $v^P_{\mathsf{out}} \in A_i$. Assume otherwise, then, since $A_i = \cap_{\ell \neq i} \langle A \setminus A_\ell \rangle$, there exists $k \neq i$ such that $v^P_{\mathsf{out}} \notin \langle A \setminus A_k \rangle$. 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 $v^P_{\mathsf{out}} \notin \langle A \setminus A_k \rangle$, contradicting Convex Validity. Therefore, we get that $v^P_{\mathsf{out}} \in A_i$ 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 $v^P_{\mathsf{out}} \in A_i$ for any $P \in G_i$, where $i \in [m]$. □

Lemma 6. *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 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 $v^1_{\mathsf{out}} \in A_1$ and $v^2_{\mathsf{out}} \in A_2$.*

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 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 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_a^1 and G_a^2. The virtual copies in G_a^1 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_a^2 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 $v_{\mathsf{out}}^1 \in A_1$ and resp. $v_{\mathsf{out}}^2 \in A_2$, completing the proof. □

Lemma 7. *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 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 has a terminating execution where there are honest parties $P_1, \ldots, P_m$ such that the output v_{out}^i of party P_i satisfies $v_{\mathsf{out}}^i \in A_i$.*

Proof. Consider a protocol Π satisfying Convex 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 5. □

Theorem 11. *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, no (deterministic or randomized) n-party protocol satisfying Convex 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.*

Proof. For the case $2 \leq n \leq 2 \cdot t_s + t_a$, by Lemma 8, there is an ω-adversarial family for $\mathcal{C}$. Since $2 \leq \omega$, using Lemma 9, let $\mathcal{A} = \{A_1, A_2\}$ be a 2-adversarial family for $\mathcal{C}$. Consider a protocol Π satisfying Convex Validity and Termination. By Lemma 6, 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 8, there is an ω-adversarial family for $\mathcal{C}$. Since $m \leq \omega$, using Lemma 9, let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be an m-adversarial family for $\mathcal{C}$. Consider a protocol Π satisfying Convex Validity and Termination. By Lemma 7, 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. □

We add that adversarial families can also be used to recover more (known) impossibility results. For instance, we give a short proof below of the classic AA lower bounds in $\mathbb{R}^D$ with straight-line convexity [31,32,41]. Similarly, one can also recover the requirement of $n > 2 \cdot t_s + t_a$ for $\mathbb{R}$ in the network-agnostic model [23].

Theorem 26. *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 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 5 any protocol Π satisfying Convex 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.2 Comparison with [37, Theorems 17 and 13]

In this section, we compare our impossibility results with the related [37, Theorems 17 and 13]. We find that our results generalize the aforementioned, with the exception of the first part of [37, Theorems 13], to which our findings are orthogonal. However, we exhibit an error in the proof of this part, rendering the result false in general.

Theorem 27 ([37, Theorem 17])**.** *Let $\mathcal{C}$ be a convex geometry with Helly number ω. If the network is synchronous and $n \leq \omega \cdot t$, then no n-party protocol satisfies Convex Validity, Termination and Exact Agreement.*

Contrasting this with Theorem 10, 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 28 ([37, Theorem 13])**.** *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 Validity and Termination, then:*

1. *If $n \leq (c+1) \cdot t$ there is an execution where the honest outputs do not form a free set in $\mathcal{C}$.*
2. *If $n \leq (\omega+1) \cdot t$ 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 10, 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 Validity and Termination in some convex geometries.

Part 1 of Theorem 28, 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 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 29 ([37, Lemma 15 of the full version [38]])**.** *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 28. To do so, we will use the fact [39, 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, when the network is asynchronous, convex consensus be solved assuming $n > 4 \cdot t$, while Part 1 of Theorem 28 would imply that it can not be solved for $n \leq (k + 3) \cdot t$, which are incompatible statements for k large enough.

The keen-eyed reader might note that [37, 38] require the universe to be finite, which is not the case for our counterexample. To also construct a counterexample with a finite

[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 [37, Lemma 16 of the full version [38]], where A is assumed to be irredundant.

universe, we will use the same technique, replacing $\mathbb{R}^2$ with straight-line convexity by a finite convexity space $\mathcal{X}$. The requirements for our technique to apply are mild since increasing k keeps the Helly number constant while strictly increasing the lower bound on the Carathéodory number, provided the Carathéodory number of $\mathcal{X}$ is at least 3. Hence, for k sufficiently large, the Carathéodory number of $\mathcal{C}_k = \bigoplus_{\ell=1}^{k} \mathcal{C} = \bigoplus_{\ell=1}^{k} \mathcal{X}$ will exceed its Helly number.[8] It remains to construct a finite $\mathcal{X}$ with Carathéodory number at least 3. This is not at all difficult: let the universe be four points $A, B, C, D \in \mathbb{R}^2$ such that A, B, C form an equilateral triangle and D is the center of the triangle, and the convexity notion be inherited from $\mathbb{R}^2$ with straight-line convexity; i.e., $\langle \{A, B, C\} \rangle = \{A, B, C, D\}$. This space has a Carathéodory number of at least 3 because the set $\{A, B, C\}$ is irredundant (in fact exactly 3 because $\{A, B, C, D\}$ is not irredundant).

C Communication Primitives

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

C.1 Reliable Broadcast

It will be useful to consider the rBC definition of [24], which makes the termination time explicit.

Definition 30 (Reliable Broadcast with explicit termination time)**.** *Let Π be a protocol where a designated party S (the sender) holds a value v_S, and every party P may output a value v_P. Consider the following properties:*
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')-secure 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 when up to t_a corruptions if it runs in an asynchronous network.

Our protocols will make use of two rBC protocols. The first one is Bracha's protocol [12], which does not assume PKI. The theorem below follows from the analysis of [24].

Theorem 31 (Bracha [12])**.** *If $n > 3t$. there is a $(t, t, c_{\mathsf{rBC}}, c'_{\mathsf{rBC}})$-secure rBC protocol Π_{arBC}, where $c_{\mathsf{rBC}} := 3$ and $c'_{\mathsf{rBC}} := 2$. Π_{arBC} achieves round complexity $O(1)$. If ℓ denotes the universe elements' size in bits, it achieves a communication complexity of $O(n^2 \cdot \ell)$ bits.*

The second protocol is that of Momose and Ren [34]. The theorem below follows from the analysis of [23].

[8] Note that for finite $\mathcal{X}$ the Helly number is always well-defined.

Theorem 32 (Momose and Ren [34]). *Assume that $n > 2 \cdot t_s + t_a$ and $t_s \geq t_a$. Then, there is an n-party protocol achieving $(t_s, t_a, c_{\mathsf{rBC}}, c'_{\mathsf{rBC}})$-secure rBC (assuming PKI), where $c_{\mathsf{rBC}} := 3$ and $c'_{\mathsf{rBC}} := 1$. The protocol has round complexity $O(1)$. If ℓ denotes the universe elements' size in bits and κ is the security parameter, it achieves a communication complexity of $O(n^2 \cdot \ell + n^3 \cdot \kappa)$ bits. If, in addition, threshold signatures are available, the communication complexity reduces to $O(n^2 \cdot \ell + n^2 \cdot \kappa)$.*

C.2 Analysis of $\Pi_{a\mathsf{ACS}}$

In this section, we analyze the (adapted) protocol $\Pi_{a\mathsf{ACS}}$ described in Section 5 and show that it achieves (t_s, t_a)-secure ACS when $t_s, t_a < n/3$. We prove the theorem below.

Theorem 33. *There is a (t_s, t_a)-secure ACS protocol for $t_s, t_a < n/3$.*

We separate the proof into the analysis of $\Pi_{a\mathsf{ACS}}$ in the synchronous setting only, and then in the asynchronous setting only.

Lemma 34. *When running in a synchronous network where at most t_s of the parties involved are corrupted, $\Pi_{a\mathsf{ACS}}$ achieves Validity, Exact Agreement, t_s-Output Size, Probabilistic Termination, and Honest Core.*

Proof. First, Validity follows immediately from the properties of $\Pi_{a\mathsf{rBC}}$. In addition, the values are received consistently.

Since the network is synchronous, at least the $n - t_s$ honest invocations of $\Pi_{a\mathsf{rBC}}$ terminate by time $\tau_{\mathtt{start}} + c_{a\mathsf{rBC}} \cdot \Delta$. Hence, by time $\tau_{\mathtt{start}} + c_{a\mathsf{rBC}} \cdot \Delta$, every honest party joins (at least) the $n - t_s$ invocations of $\Pi_{a\mathsf{BA}}$ corresponding to honest parties with input 1.

$\Pi_{a\mathsf{BA}}$'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 $\Pi_{a\mathsf{BA}}$ 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 $\Pi_{a\mathsf{rBC}}$. Then, $c'_{a\mathsf{rBC}}$-Conditional Termination ensures that all honest parties receive this value.

Hence, $\Pi_{a\mathsf{BA}}$ has allowed the honest parties to agree on a set of at least $n - t_s$ parties $\mathcal{P}$. This set 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, t_s-Output Size, Probabilistic Termination, and Honest Core hold. □

Lemma 35. *When running in an asynchronous network where at most t_a of the parties involved are corrupted, Π_{ACS} achieves Validity, Exact Agreement, t_s-Output Size, and Probabilistic Termination.*

Proof. Similarly to the proof for the synchronous case, Validity follows immediately from $\Pi_{a\mathsf{rBC}}$'s Validity property. In addition, the values received are consistent.

We first show that at least $n - t_s$ invocations of $\Pi_{a\mathsf{BA}}$ terminate with output 1. Note that no honest party joins $\Pi_{a\mathsf{BA}}$ with input 0 until $n - t_s$ of the $\Pi_{a\mathsf{BA}}$ invocations have terminated with output 1. Assuming that this is never the case, we note that $c_{a\mathsf{rBC}}$-Honest Termination ensures that at least the $\Pi_{a\mathsf{rBC}}$ invocations of honest senders terminate, and honest parties may join the $\Pi_{a\mathsf{BA}}$ invocations corresponding to honest senders with input 1. Therefore, eventually, $n - t_s$ of the Π_{BA} invocations indeed return 1.

If the $\Pi_{a\mathsf{BA}}$ invocation corresponding to some party P returns 1, Weak Validity ensures that at least one honest party has joined this invocation with input 1, and therefore has

received a value from P via $\Pi_{a\mathsf{rBC}}$. Then, $\Pi_{a\mathsf{rBC}}$'s Consistency and $c'_{a\mathsf{rBC}}$-Conditional Termination properties ensure that all parties eventually receive the same value from P.

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

C.3 Gather

In the following, we describe our protocol Π_{GTHR} realizing Theorem 20, restated below.

Theorem 20. *If $n > 2 \cdot t_s + t_a$ and $t_a \leq t_s$, there is a (t_s, t_a)-secure GTHR protocol Π_{GTHR} (assuming PKI). The protocol has round complexity $O(1)$. If ℓ denotes the universe elements' size in bits and κ is the security parameter, it achieves a communication complexity of $O(n^3 \cdot \ell + n^4 \cdot \kappa)$ bits (can be reduced to $O(n^3 \cdot \ell + n^3 \cdot \kappa)$ with threshold signatures).*

Π_{GTHR} follows the outline of the Overlap All-to-All Broadcast protocol of [23], and of the initialization subroutine used in the AA protocol of [24, Section 5]. That is, it heavily relies on the *witness technique* [1].

Our protocol Π_{GTHR} will make use of an underlying rBC protocol Π_{rBC}. When instantiated with the protocol of Theorem 32, this will lead exactly to the construction proving Theorem 20. Making use of the rBC protocol of Theorem 31 instead will lead to a (t_s, t_a)-secure GTHR protocol in the $t_s, t_a \leq n/3$ setting that does not assume any cryptographic setup.

Theorem 36. *If $t_a \leq t_s < n/3$, there is a (t_s, t_a)-secure GTHR protocol Π_{GTHR} with no trusted setup.*

As mentioned in Section 5, our protocol Π_{GTHR} adds one more step to the Overlap All-to-All Broadcast (oBC) protocol of [23], which we denote by Π_{oBC}. We need to highlight the difference between the definition of (t_s, t_a)-secure oBC presented in [23] and our definition of (t_s, t_a)-secure GTHR: oBC does not require t_s-Common Core. Instead, the oBC definition of [23] requires the weaker property t_s-Overlap:

***T*-Overlap**: If two honest parties P and P' terminate, then $\left|\mathcal{M}_P \cap \mathcal{M}_{P'}\right| \geq n - T$.

Until we reach the additional step, the protocols Π_{oBC} and Π_{GTHR} are identical. Π_{oBC} (and hence also Π_{GTHR}) heavily relies on the *witness technique* [1]. That is, in both protocols, parties send their values via Π_{rBC}. Then, they report to each other which values they received. When the values reported by a party P match the values received by a party P' via Π_{rBC}, P' marks P as a *witness*. In Π_{oBC}, parties are ready to terminate when (i) sufficient time has passed for honest values to be received via Π_{rBC} in a synchronous network, ensuring Honest Core; (ii) parties have gathered sufficient witnesses to ensure that every two honest parties P and P' have a common witness $P^\star$. This way, P and P' have received the same set of $(n-t_s)$ values reported by $P^\star$, and therefore the t_s-Overlap property holds. To achieve the superior guarantee t_s-Common Core required by GTHR, we will need a stronger termination condition as well.

Common steps of Π_{oBC} and Π_{GTHR}. We now describe the common steps of Π_{oBC} and Π_{GTHR} more precisely. Parties distribute their inputs via Π_{rBC}. When a party receives a value v from P via Π_{rBC}, it adds (v, P) to a set of value-party (or value-sender) pairs $\mathcal{M}$. Additionally, it adds P 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. Even after broadcasting W_0, parties may continue gathering level-zero witnesses. Then, if a party 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 ($W_0' \subseteq W_0$), P marks P' as a *level-one witness* by adding it to its set W_1.

Once each honest party gathers $n - t_s$ level-one witnesses, we have the guarantee that every pair of honest parties has a level-one witness in common. This means that every pair of honest parties has received $n - t_s$ common values via Π_{rBC}. This is the point where Π_{oBC} allows the parties to output the set of value-sender pairs obtained so far and terminate, as (t_s, t_a)-secure oBC is achieved.

Additional step in Π_{GTHR}. In contrast, to achieve the stronger property t_s-Common Core, our GTHR protocol continues: following the insights from the asynchronous GTHR protocol of [2], we obtain that, when $n - t_s$ honest parties hold sets W_1 of size $n - t_s$, then $t_s + 1$ honest parties have a common level-one *honest* witness $P^\star$. 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. P may continue marking parties as level-one witnesses even after sending its set W_1. When receiving a set 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 reported sets W_1 containing $P^\star$ – 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.

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 $\tau_{\mathtt{now}} \geq \tau_{\mathtt{start}} + 2c_{\mathsf{rBC}} \cdot \Delta$ 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 $\tau_{\mathtt{now}} \geq \tau_{\mathtt{start}} + (2c_{\mathsf{rBC}} + c'_{\mathsf{rBC}}) \cdot \Delta$ 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 20 follows immediately from Lemma 40 and Lemma 45, included below.

Analysis in a 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 37. *Let P be an honest party. By time $\tau_{\texttt{start}} + c_{rBC} \cdot \Delta$, P holds a set W_0 containing all $n - t_s$ honest parties.*

Proof. Follows from Π_{rBC}'s c_{rBC}-Honest Termination: all honest parties send their input value at time $\tau_{\texttt{start}}$, and these values are received by time $\tau_{\texttt{start}} + c_{\mathsf{rBC}} \cdot \Delta$. Hence, P adds at least the $n - t_s$ honest parties to its set W_0 by time $\tau_{\texttt{start}} + c_{\mathsf{rBC}} \cdot \Delta$. □

Lemma 38. *Let P and P' denote two honest parties. By time $\tau_{\texttt{start}} + 2c_{rBC} \cdot \Delta$, P has added P' to its set W_1.*

Proof. According to Lemma 37, at time $\tau_{\texttt{start}} + c_{\mathsf{rBC}} \cdot \Delta$, P' has sent its set W_0' to all parties via Π_{rBC}. Hence, P receives W_0' by time $\tau_{\texttt{start}} + 2c_{\mathsf{rBC}} \cdot \Delta$ 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 $\tau_{\texttt{start}} + c_{\mathsf{rBC}} \cdot \Delta$. Then, the Consistency and c'_{rBC}-Conditional Termination properties ensure that P has received these values as well by time $\tau_{\texttt{start}} + (c_{\mathsf{rBC}} + c'_{\mathsf{rBC}}) \cdot \Delta \leq \tau_{\texttt{start}} + 2c_{\mathsf{rBC}} \cdot \Delta$ (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 $\tau_{\texttt{start}} + 2c_{\mathsf{rBC}} \cdot \Delta$. □

Lemma 39. *Let P and P' denote two honest parties. By time $\tau_{\texttt{start}} + (2c_{rBC} + c'_{rBC}) \cdot \Delta$, P has added P' to its set W_2.*

Proof. Lemma 38 ensures that, at time $\tau_{\texttt{start}} + 2c_{\mathsf{rBC}} \cdot \Delta$, 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 $\tau_{\texttt{start}} + (2c_{\mathsf{rBC}} + 1) \cdot \Delta \leq \tau_{\texttt{start}} + (2c_{\mathsf{rBC}} + c'_{\mathsf{rBC}}) \cdot \Delta$.

Note that, if P' has added a party P'' in its set W_1' by time $\tau_{\texttt{start}} + 2c_{\mathsf{rBC}} \cdot \Delta$, 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 $\tau_{\texttt{start}} + (2c_{\mathsf{rBC}} + c'_{\mathsf{rBC}}) \cdot \Delta$, $W_1' \subseteq W_1$ holds and hence P adds P' to W_2. □

Lemma 40. *Π_{GTHR} satisfies Simultaneous Termination, Honest 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 Core property follows from Lemma 37. Finally, Lemma 39 ensures that all honest parties output at time $(2c_{\mathsf{rBC}} + c'_{\mathsf{rBC}}) \cdot \Delta$, hence Simultaneous 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 41. *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 42. *For every honest party P, it eventually holds that $\left|W_2\right| \geq n - t_s$.*

Proof. According to Lemma 41, every honest party eventually sends its set W_1 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 43. *Assume every honest party has sent its set W_1. Then, there is an honest party $P^\star$ that belongs to $t_s + 1$ sets W_1 sent by the honest parties.*

Proof. In the following, we use the term *reported* to refer to parties included in the sets W_1 sent by the honest parties. Hence, we say that P was reported by an honest party P' if P is in the set W_1 sent by P'. If P is an honest party, then this is an *honest report*. (Note that if P was added to the set W_1 after P' sent its set, then P was not reported by P'.)

We then analyze the total number of honest reports, and we prove the result with the help of an averaging argument. Since each of the $n - t_a$ honest parties reports $n - t_s$ parties, we have $(n-t_a)\cdot(n-t_s-t_a)$ honest reports. This implies that there is an honest party $P^\star$ reported by least $(n - t_a) \cdot (n - t_s - t_a)/(n - t_a) = n - t_s - t_a$ honest parties. Then, since $n - t_s - t_a > t_s$, $P^\star$ was reported by at least $t_s + 1$ honest parties, which concludes the proof.

□

Lemma 44. *Assume $\left|W_2\right| \geq n - t_s$ holds for all honest parties. Then, there is common subset of $n - t_s$ value-sender pairs in their sets $\mathcal{M}$.*

Proof. Lemma 43 ensures that there is an honest party $P^\star$ included in the sets W_1 sent by 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 wait until they add $P^\star$ to their set W_1. That is, they are forced to wait 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 each of the honest parties' output sets $\mathcal{M}$. □

Lemma 45. *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 42 ensures Termination, and, together with Lemma 44, ensures t_s-Common Core. □

C.4 Analysis of Π_{ACS}

We include the proof of Theorem 15, restated below.

Theorem 15. *If $n > 2 \cdot t_s + t_a$ and $t_a \leq t_s$, there is a (t_s, t_a)-secure ACS protocol Π_{ACS} (assuming PKI). The protocol has expected round complexity $O(1)$. If ℓ denotes the universe elements' size in bits and κ is the security parameter, its expected communication complexity is $O(n^3 \cdot \ell + n^4 \cdot \kappa)$ bits. If threshold signatures are available, the expected communication complexity reduces to bits.*

Proof. In the following, we show that Π_{ACS} achieves (t_s, t_a)-secure ACS when $n > 2 \cdot t_s + t_a$.

First, the Validity and Consistency properties follow immediately from the Validity and Consistency 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 all Π_{BA} invocations corresponding to honest parties result in output 1, and hence all honest values are included in the output sets $\mathcal{M}$, ensuring the Honest Core property. 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 $\mathcal{M}_{\mathsf{GTHR}}$. By Observation 21, eventually, all honest parties receive this value as well. Therefore, all honest parties output the same set $\mathcal{M}$, hence Exact Agreement and Probabilistic Termination hold.

If the network is asynchronous, since Π_{GTHR} achieves Termination, all honest parties eventually join all Π_{BA} invocations and hence agree on a bit for each party. Π_{GTHR}'s t_s-Common Core property ensures that there exist at least $n - t_s$ invocations of Π_{BA} in which all honest parties input 1, 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}. Observation 21 then ensures that all parties eventually receive this value, and therefore all honest parties output the same set $\mathcal{M}$ of at least $n - t_s$ value-sender pairs, hence Exact Agreement, t_s-Output Size and Probabilistic Termination hold. □

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 the following version of the Pigeonhole principle.

Lemma 46 (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 16. *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 $restrict_{t'}(\mathcal{M}) := \{M \subseteq \mathcal{M} : \left|M\right| = \left|\mathcal{M}\right| - t'\}$. 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 46, from which the required $\cap_{i=1}^{\omega} \langle M_i \rangle \neq \varnothing$ naturally follows. To apply the lemma, we need that $\sum_{i=1}^{\omega} \left|M_i\right| > (\omega - 1) \cdot \left|\mathcal{M}\right|$. Since $\sum_{i=1}^{\omega} \left|M_i\right| = \omega(\left|\mathcal{M}\right| - t')$, this amounts to showing that $\omega(\left|\mathcal{M}\right| - t') > (\omega - 1) \cdot \left|\mathcal{M}\right| \iff \left|\mathcal{M}\right| > \omega \cdot t' \iff n - t_s + k > \omega \cdot t'$. Distinguish two cases:

- If $k \geq t_a$, the latter becomes $n - t_s + k > \omega k \iff n - t_s > (\omega - 1) \cdot k$, 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 23, 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 47. *Let $\mathcal{M}_1, \mathcal{M}_2$ denote two sets of value-sender 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 $k_1 = \left|\mathcal{M}_1\right| - (n - t_s) \geq t_a$, and define $k_\cup = \left|\mathcal{M}_1 \cup \mathcal{M}_2\right| - (n - t_s)$. 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 16 then immediately implies that $safe_{k_\cup}(\mathcal{M}_1 \cup \mathcal{M}_2) \neq \varnothing$. Moreover, $restrict_{k_1}(\mathcal{M}_1) = \{M \subseteq \mathcal{M}_1 : \left|M\right| = n - t_s\} \subseteq \{M \subseteq \mathcal{M}_1 \cup \mathcal{M}_2 : \left|M\right| = n - t_s\} = restrict_{k_\cup}(\mathcal{M}_1 \cup \mathcal{M}_2)$. 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 48. *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 equationally:

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

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

Lemma 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 $\left|\mathcal{M}_1\right| - (n - t_s) \leq t_a$. 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 16 implies that $safe_{t_a}(\mathcal{M}_1 \cap \mathcal{M}_2) \neq \varnothing$. Moreover, Lemma 48 implies that $safe_{t_a}(\mathcal{M}_1 \cap \mathcal{M}_2) \subseteq safe_{t_a}(\mathcal{M}_1)$. □

Lemma 50. *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 16, 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 46 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 46 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 23. *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 16, 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 49 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 47 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 50, 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 on Chordal Graphs

In this section, we provide the formal proof of Theorem 24, restated below.

Theorem 24. *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)-secure deterministic protocol achieving Monophonic Convex Validity, Termination and Agreement within Graph Distance 1.*

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 discuss Convex Validity and Agreement.

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 Convex Validity condition is guaranteed, as a direct consequence of Lemma 16 along with the properties of Π_{GTHR}.

Agreement within Graph Distance 1 will follow from Lemma 51, stated below, which shows that H_{it} is a strict subset of $H_{\mathsf{it}-1}$ unless $H_{\mathsf{it}-1}$ already induces a clique, meaning that Agreement within Graph Distance 1 has already been achieved. This way, within $|V|$ iterations, our protocol Π_{Chordal} achieves Agreement within Graph Distance 1, and therefore achieves (t_s, t_a)-secure AA when the condition $n > \omega \cdot t_s + t_a$ is satisfied.

Lemma 51. *Assume $1 \leq \mathsf{it} \leq \mathsf{max\_it}$ and that $H_{\mathsf{it}-1}$ does not form a clique in G. Then, $H_{\mathsf{it}} \subsetneq H_{\mathsf{it}-1}$.*

The proof of Lemma 51 will make use of a few helper lemmas, stated below.

Lemma 52 (Dirac'1961, [16])**.** *Every chordal graph has a simplicial vertex. Every chordal graph that is not a clique has two non-adjacent simplicial vertices.*

Lemma 53. *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 A.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 54. *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 $\langle X \cup Y \rangle = \langle X \cup \langle Y \rangle \rangle$. □

We may now provide the proof of Lemma 51.

Proof of Lemma 51. 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 extreme by Lemma 53. 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 16, $S_{\mathsf{it}}^P \subseteq H_{\mathsf{it}-1}$. The properties of Π_{GTHR} enable us to apply Lemma 23 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_{it}^P. We will show that $v_{\mathsf{it}}^P \neq s$. Since s is extreme, 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 $S_{\mathsf{it}}^P = ex(S_{\mathsf{it}}^P)$, then party P sets $v_{\mathsf{it}}^P = \max S_{\mathsf{it}}^P$. However, as $y \in S_{\mathsf{it}}^P$ and $y \succ s$ it follows that $v_{\mathsf{it}}^P \succ s$, so $v_{\mathsf{it}}^P \neq s$, as claimed.
   (b) If $S_{\mathsf{it}}^P \neq ex(S_{\mathsf{it}}^P)$, then party P picks $v_{\mathsf{it}}^P \in S_{\mathsf{it}}^P \setminus ex(S_{\mathsf{it}}^P)$ arbitrarily. Since $S_{\mathsf{it}}^P \subseteq H_{\mathsf{it}-1}$, by Lemma 54, it follows that $v_{\mathsf{it}}^P \in S_{\mathsf{it}}^P \setminus ex(H_{\mathsf{it}-1})$. As $s \in ex(H_{\mathsf{it}-1})$, this means that $v_{\mathsf{it}}^P \neq s$, 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 $v_{\mathsf{it}}^P \neq a$. If $a \notin S_{\mathsf{it}}^P$, then this is clear, so assume $a \in S_{\mathsf{it}}^P$. As a result, we know that $\{s, a\} \subseteq S_{\mathsf{it}}^P$. Since the edge $s - a$ does not exist in G and its endpoints are contained in S_{it}^P, it follows that S_{it}^P does not induce a clique, meaning by Lemma 25 that S_{it}^P is not free, so $S_{\mathsf{it}}^P \neq ex(S_{\mathsf{it}}^P)$. As a result, party P picks $v_{\mathsf{it}}^P \in S_{\mathsf{it}}^P \setminus ex(S_{\mathsf{it}}^P)$ arbitrarily. Since $S_{\mathsf{it}}^P \subseteq H_{\mathsf{it}-1}$, by Lemma 54, it follows that $v_{\mathsf{it}}^P \in S_{\mathsf{it}}^P \setminus ex(H_{\mathsf{it}-1})$. Since a is a simplicial vertex in the subgraph induced by $H_{\mathsf{it}-1}$, it follows by Lemma 53 that $a \in ex(H_{\mathsf{it}-1})$, so $v_{\mathsf{it}}^P \neq a$, as claimed. To conclude, $a \in H_{\mathsf{it}-1} \setminus H_{\mathsf{it}}$, from which the conclusion follows. □