Resilient Byzantine Agreement with Predictions

Julien Dallot
TU Berlin
Berlin, Germany

Darya Melnyk
TU Berlin
Berlin, Germany

Tijana Milentijević
TU Berlin
Berlin, Germany

Stefan Schmid
TU Berlin & Weizenbaum Institute
Berlin, Germany

Patrik Welters
Humboldt-Universität zu Berlin
Berlin, Germany

Abstract

The Byzantine Agreement (BA) problem is a fundamental task in distributed computing where nodes in a network need to agree on a common output in the presence of arbitrary (worst-case) node failures. In real-world applications, nodes can be monitored over time to provide ML predictions of faulty behavior in a network. In this work, we answer the question of whether predictions can improve the fault tolerance of a distributed system without sacrificing correctness when the predictions are wrong. We consider a learning-augmented study of Byzantine agreement in which each node receives, in addition to its input bit, a predicted set of honest nodes. We focus on *algorithmic resilience* — the maximum number of faulty nodes an algorithm can tolerate — and present algorithms and impossibility results whose resilience depends on the accuracy of the predictor. As our first main result, we bring a complete characterization of the consistency–robustness tradeoffs in both the non-authenticated and authenticated settings: for n nodes and a parameter $\alpha \in [0, 1[$, we present algorithms that tolerate up to $\alpha \cdot n$ faulty nodes when the predictor is correct (*consistency*), and up to $\frac{1-\alpha}{2} \cdot n - 1$ faulty nodes when the predictor is arbitrarily wrong (*robustness)*. In the authenticated setting, the robustness bound improves to $(1-\alpha) \cdot n - 1$. We prove matching impossibility results, showing that these tradeoffs are optimal and independent of the particular prediction system. Our second main result concerns *smoothness*, i.e., how fast resilience degrades as the predictor makes more errors. We show that resilience never drops abruptly and decreases at a slow pace: in particular, for small errors, each additional mispredicted node costs at most one unit of resilience.

1 Introduction

Byzantine Agreement (BA) [1, 2] is a most fundamental and intensively studied problem in distributed computing. It requires nodes to agree on a common output despite arbitrary, worst-case node failures, called Byzantine behavior in the literature. This work explores how predictions about the other nodes' behavior can be used to improve the resilience of BA. Indeed, in practice, nodes can often monitor and collect data about other nodes, and use machine-learned classifiers, anomaly detectors, or reputation algorithms, to guess how trustworthy other nodes are. This is a natural model for many distributed systems operating over longer time periods, such as blockchain: suspicious behavior of the participants can be monitored and, to some extent, detected [3]. However, with new participants and new capabilities of these systems, mistakes in the prediction are not preventable.

Motivated by Machine Learning (ML), learning-augmented algorithms have already received much attention and have been applied successfully in various domains already. Early work originated in the context of online algorithms, where predictions about the future events are used by algorithms to

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

overcome worst-case bounds [4], and many significant insights have been obtained how to improve the competitive ratio of different online algorithms [5, 6, 7, 8, 9, 4]. But also data structures [10] and most recently also distributed computing tasks [11, 12] have been improved with predictions. In particular, Ben-David et al. [12] studied whether nodes can benefit from predictions about the correctness of other nodes in Byzantine Agreement to speed up agreement, hence focusing on the *efficiency*.

In this paper, we are interested in the *resilience* of learning-augmented algorithms. In particular, we initiate the study of whether predictions about node behaviors can allow us to tolerate more node failures in Byzantine Agreement. This is challenging since predictions are inherently unreliable. And while accurate predictions can substantially improve the resilience, relying on them unconditionally can compromise correctness when they are wrong. Hence, as with any learning-augmented algorithm, a useful learning-augmented Byzantine agreement protocol should navigate the trade-off between *consistency*, i.e., benefit when predictions are correct, and *robustness*, i.e., not suffer when predictions are of poor quality or even adversarial. Beyond these extremes, we aim for *smoothness*: the protocol should lose resilience gradually, rather than abruptly, as the number of incorrect predictions increases.

1.1 Contributions

This paper initiates the study of whether and how ML predictions can be used to improve the resilience of algorithms for the Byzantine Agreement problem. Specifically, we ask: given n nodes and t faults, can we use predictions to overcome the upper bounds of $n > 3t$ (without authentication) [2] and $n > 2t$ (with authentication) [1] on the resilience of the agreement protocols? We answer these questions affirmatively.

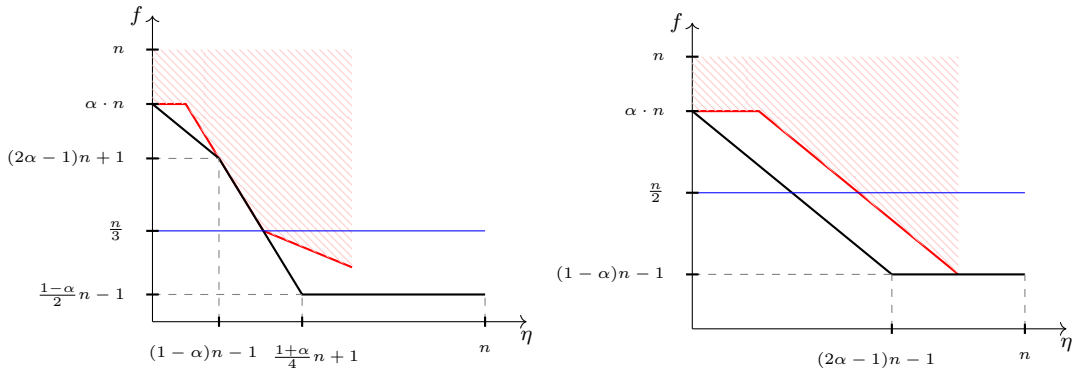
We design learning-augmented algorithms which, given a prediction of which nodes are trustworthy (honest), surpass any prediction-free algorithm in case of accurate predictions (i.e., consistency) while ensuring provable resilience guarantees *against any prediction* (i.e., robustness). In addition, we study the resilience of our algorithms as the prediction accuracy declines (i.e., smoothness) and provide resilience guarantees for any fraction of accurately predicted nodes. Our algorithms are tuned with a *trust parameter* $\alpha \in [0, 1[$ which is the desired consistency (tolerated fraction of faulty nodes for good predictions) and interpolates between prudent or risky algorithms depending on how much one trusts the predictor. We provide theoretical results for four settings: we consider two communication models, where the communication channels can either be authenticated or not; and we consider two models of predictions, a global model in which the predictions are publicly available to all nodes (e.g., broadcast by an ML monitoring system) and a local model in which predictions are only available locally. Our results on consistency-robustness trade-offs are summarized in Figure 1. Starting from existing, prediction-free agreement algorithms, we build learning-augmented algorithms to achieve better resilience bounds in the non-authenticated setting (Section 3 and Section 4) as well as the authenticated setting (Appendix C) and finally derive impossibility results for local predictions (Section 6).

Our algorithms build on classical Byzantine agreement protocols, using predictions to select a common set of nodes on which the agreement is executed. In the non-authenticated case, we use the Phase King algorithm [13], which consists of phases of rotating leaders (kings) and uses majority-based voting to guarantee valid agreement when fewer than one third of the nodes are Byzantine. In the authenticated case, we build on the Dolev Strong [14] algorithm, which uses chains of digital signatures to authenticate forwarded messages and ensure consistent broadcast in the presence of Byzantine nodes. In this case, a valid agreement can be achieved when strictly less than a half of the nodes is Byzantine.

Characterizing consistency and robustness. Our first technical contribution is a complete characterization of the possible consistency and robustness trade-offs under global predictions for non-authenticated and authenticated channels. Results in the global setting are summarized in Table 1. We derive our tight characterization with indistinguishability arguments [1, 2]: we show that, given any algorithm with a certain consistency α, we can construct a set of configurations and prediction inputs among which at least one defeats any algorithm with a robustness too high ($\frac{1-\alpha}{2}$ for the non-authenticated case, $1 - \alpha$ for the authenticated case). These results show that trade-offs cannot be avoided when it comes to resilience: unlike round complexity [12], it is not possible to use predictions to strictly improve an algorithm in all cases, for instance by running a "back-up" algorithm that

Pred. Type	Communications	**Consistency**	**Robustness**	**Impossible Robustness**
Global	Non-Auth	$\alpha \in \left[\frac{1}{3}, 1\right[$	$\frac{1-\alpha}{2} - \frac{1}{n}$ (Thm.1)	$\frac{1}{n}\left\lceil\frac{(1-\alpha)n}{2}\right\rceil$ (Thm.3)
	Auth	$\alpha \in \left[\frac{1}{2}, 1\right[$	$(1-\alpha) - \frac{1}{n}$ (Thm.5)	$\frac{1}{n}\lceil(1-\alpha)n\rceil$ (Thm.7)
Local	Non-Auth	$\alpha \in \left[\frac{1}{3}, \frac{1}{2}\right[$	(unknown)	(unknown)
	Non-Auth and Auth	$\alpha \in \left[\frac{1}{2}, 1\right[$	0	$\frac{1}{n}$ (Thm.9)

Table 1: Our results on consistency-robustness trade-offs. On each row, the first two columns indicate the setting, the third column shows the range of relevant consistencies (i.e., greater than the resilience of prediction-free algorithms), the fourth column shows our robustness for the target consistency α, and the last column shows an upper-bound (impossibility result) on the robustness for a consistency of α.



(a) Non-Authenticated Channels

(b) Authenticated Channels

Figure 1: The Smoothness curve under global predictions for the non-authenticated and authenticated settings (obtained for $\alpha = 0.8$). In each plot, the black curve is the algorithm's smoothness, i.e., the worst-case resilience knowing the predictive error is η, the red curve and the dashed red region are impossible regions, and the blue horizontal line is the optimal resilience of a prediction-free algorithm. Observe that the resilience has no sudden jumps in the range of almost accurate predictions. The descending curves show great tolerance to prediction errors and allow for a clear supremacy of our algorithms compared to standard prediction-free ones on a wide range of imperfect predictions.

could take over if the learning-augmented algorithm fails — in our case, it is impossible to detect that the learning-augmented algorithm has failed in the first place.

Inaccurate predictions are still beneficial. Our second main result is on the *smoothness* [4] of our algorithms, i.e., how their resilience evolves when errors are gradually introduced in the prediction. As noted by Boyar et al [11], this metric is of particular interest for distributed systems: when n is large, the consistency metric loses practical value since perfectly predicting the behavior of every node becomes increasingly unlikely. Following [12], we define an *error function* $\eta \in [\![0, n]\!]$ (Definition 5) that counts the number of mispredicted nodes. Then, we derive resilience guarantees for our algorithms for each possible value of the error function in both non-authenticated and authenticated settings. Our main result is a linear decrease of the resilience for low values of η: each new error loses (in the worst case) one unit of resilience in both non-authenticated and authenticated settings. Overall, our results show a clear supremacy of learning-augmented algorithms compared to prediction-free ones, *even for a wide range of imperfect predictions*; our results are summarized in Figure 1.

Impossibility results for local predictions. Our last contribution is a strong impossibility result in the case of local prediction, i.e., when each node can obtain its own input prediction (and therefore predictions can vary between two nodes). We show that, in this local prediction framework, any algorithm with a consistency of $1/2$ must have a robustness of 0. This result directly implies that local predictions do not provide better guarantees in the authenticated case, from a consistency and

robustness perspective. For the non-authenticated case, the following interesting question remains open for future work: is it possible to obtain non-trivial consistency and robustness trade-offs for a consistency inside $[1/3, 1/2[$?

2 Model

Problem Setup. We consider the fundamental problem of Byzantine agreement, where n nodes in a distributed system are given a binary input value and need to agree on a common output value by communicating with each other. Each node has a unique ID, let the IDs be $1, 2 \dots n$, for simplicity. The underlying communication network is synchronous and fully connected. We assume that the nodes all start the protocol at the same time. We therefore can interpret the synchronous communication as communication in rounds, where in each round, a node can (1) send any sequence of bits to its neighbors, (2) receive messages from its neighbors, and (3) update its local state.

An (unknown) subset of nodes $F \subseteq [\![n]\!]$ are assumed to show (Byzantine) *faults*, meaning they may deviate from the algorithm in an arbitrary (worst-case) manner. Those faulty nodes may collude without limit to prevent the other nodes from reaching agreement. We use f to denote the number of faulty nodes in the system. While f is unknown, we assume that the nodes have access to a known upper-bound on f, called t; in this paper, we set t so that the problem is simply impossible to solve if $f > t$. We denote the subset of *honest* nodes, i.e., non-faulty, by $H = [\![n]\!] \setminus F$.

Authentication. In this paper, we focus on two standard models, one with *non-authenticated* communication channels and the other with *authenticated* communication channels. In the non-authenticated setting, the messages transmitted over the communication channels are in plain, non-encrypted text. A faulty node can therefore freely alter its communication history: claiming to have sent or received messages that never existed, denying actual messages, or presenting different histories to different nodes. In the authenticated setting (PKI model) however, nodes can digitally sign their messages, and the adversary is assumed computationally bounded, making it infeasible to forge or alter signed messages. The adversary can still read each message, but it cannot fabricate a false communication history.

Prediction Model. In addition to their standard input values, this paper assumes the nodes receive an additional input, the *prediction* — provided by, e.g., a monitoring system that detects suspicious behaviors — indicating which nodes are honest.

Definition 1 (Prediction)**.** *A prediction is a set $P \subseteq [\![n]\!]$ which indicates the honest nodes in the system. Predictions are unreliable and may arbitrarily differ from the actual set of honest nodes H.*

We consider two prediction models: *global* and *local*. In the *global model*, all nodes receive the same prediction P (in that case, we therefore omit the node subscript). This captures, e.g., a centralized monitoring system that flags nodes for suspicious behaviors and broadcasts its assessment of which nodes are trustworthy. In the *local model*, each node receives its own prediction, computed independently; the predictions may differ arbitrarily across nodes (same model as in [12]).

For the most part of this paper (Section 3 and Appendix C), we focus on the global model; the definitions below therefore concern global predictions only. We present impossibility results for local predictions in Section 6 where we define analogous definitions for local predictions.

Our goal is to design synchronous algorithms that use predictions to solve the *Byzantine agreement problem*. We now formally define such our problem.

Definition 2 (Learning-Augmented Byzantine Agreement Problem)**.** *Each honest node i receives two inputs: (1) a* binary value *v_i and (2) a* prediction *P_i. A learning-augmented algorithm ALG solves the Byzantine agreement problem if it satisfies:*

- ***Termination:** All honest nodes terminate in finite time.*
- ***Agreement:** All honest nodes terminate with the same output bit.*
- ***Validity:** The output of every honest node is the input value of some honest node.*

*We say that ALG reached a **valid agreement** once it terminated in all honest nodes with the agreement and validity properties holding.*

Evaluation Metrics. In this paper, we evaluate algorithms using the standard *resilience* metric: the maximum number of faulty nodes the algorithm can tolerate. Without predictions, the optimal resilience bounds are well-known: $\lceil n/3 \rceil - 1$ faulty nodes without authentication, and $\lceil n/2 \rceil - 1$ with authentication.

We study how these bounds improve when nodes have access to predictions. Following the learning-augmented algorithms literature [4], we use three metrics defined hereafter: *consistency*, *robustness*, and *smoothness*. Consistency measures resilience under correct predictions:

Definition 3 (Consistency)**.** *Let $\alpha \in [0, 1[$. A learning-augmented algorithm ALG is α-consistent if, for every configuration with at most $\alpha \cdot n$ faulty nodes, there exists a prediction assignment under which ALG reaches valid agreement.*

Robustness measures resilience regardless of the predictions:

Definition 4 (Robustness)**.** *Let $\beta \in [0, 1]$. A learning-augmented algorithm ALG is β-robust if, for every configuration with at most $\beta \cdot n$ faulty nodes and every prediction assignment, ALG reaches valid agreement.*

Consistency and robustness are two extremes: the former assumes perfect predictions, the latter assumes nothing about them. Smoothness interpolates between the two, capturing how resilience degrades as prediction quality worsens. Smoothness is parameterized by an error function η that measures the quality of the prediction.

Definition 5 (Error Function)**.** *Let H be the set of honest nodes and P a global prediction. The* error *of P is:*

$$\eta(H, P) = |H \setminus P| + |P \setminus H|$$

That is, $\eta(H, P)$ counts all mispredictions: nodes in H but not P (honest but wrongly predicted faulty) and nodes in P but not H (faulty but predicted honest).

Definition 6 (Smoothness)**.** *The* smoothness *of algorithm ALG is the function $s : [\![0, n]\!] \to \mathbb{N}$ where $s(i)$ is the maximum number of faulty nodes ALG can tolerate when the prediction error satisfies $\eta(H, P) = i$.*

Notations. Let i and j be two integers with $i \leq j$. By $[\![i, j]\!]$ we refer to the set $\{i, i+1 \dots j-1, j\}$, by $[\![i]\!]$ we refer to the set $\{1, 2 \dots i\}$. Let $i \in \mathbb{R}$ be a real number, we write $\lfloor i \rfloor$ to refer to the *inferior integer part* of i, that is, the largest integer number smaller than i; likewise, we write $\lceil i \rceil$ to refer to the *superior integer part* of i, which is the smallest integer number larger than i. For convenience, we extend the range notations to real numbers: let $i, j \in \mathbb{R}$ with $i \leq j$, $[\![i, j]\!]$ refers to the set $\{\lceil i \rceil, \lceil i \rceil + 1 \dots \lfloor j \rfloor - 1, \lfloor j \rfloor\}$ and $[\![i]\!]$ refers to the set $\{1, 2 \dots \lfloor i \rfloor - 1, \lfloor i \rfloor\}$.

3 Reaching Valid Agreement with Non-Authenticated Channels

This section introduces our learning-augmented algorithm pred_ba (Algorithm 1) for non-authenticated channels and global predictions. Then, we prove that pred_ba admits a trade-off between consistency and robustness dependent on the input target consistency $\alpha \in [1/3, 1[$ (Theorem 1) and establish the smoothness curve of pred_ba shown in Figure 1a (Theorem 2).

Our algorithm pred_ba (Algorithm 1) proceeds in two phases. In the first phase, all honest nodes agree on a common subset $L \subseteq [\![n]\!]$ based on the prediction. In the second phase, they run a standard Byzantine agreement algorithm restricted to the nodes in L, called **active** nodes; nodes outside L are passive and adopt the value decided by the active nodes, if any. Our learning-augmented algorithm uses the classic Phase King protocol from [13], note that any algorithm that also reaches valid agreement against $\lceil n/3 \rceil - 1$ faulty nodes or fewer can be used instead.

Controlling the number of active nodes. Ideally, L equals the set of honest nodes H so valid agreement is easily guaranteed. In the consistency scenario, this is simply obtained by setting $L = P$. However, that strategy that fully trusts P is not robust: if P contains too many faulty nodes then reaching valid agreement becomes impossible. To address this, pred_ba enforces that L has a minimum size of $\lceil 3(1-\alpha)/2 \cdot n \rceil - 1$. If P does not already satisfy this cardinality constraint, the algorithm deterministically adds nodes into L until it reached the minimum size. This simple lower-bound on the number of active nodes guarantees a trade-off between consistency and robustness:

Algorithm 1: pred_ba(v_x, P, α)

// This algorithm runs in node x
Input: the input bit v_x, prediction set $P \subseteq [\![n]\!]$, trust parameter $\alpha \in [1/3, 1[$
// Initialize L to the predicted set
$L \leftarrow P$
// Add nodes to L if needed
$y \leftarrow 1$
if $|L| < \frac{3}{2} \cdot (1-\alpha) \cdot n - 1$ **then**
  repeat
    if $y \notin L$ **then**
      $L \leftarrow L \cup \{y\}$
    $y \leftarrow y + 1$
  until $|L| \geq \frac{3}{2} \cdot (1-\alpha) \cdot n - 1$;
// Run the agreement algorithm
$t \leftarrow \lceil |L|/3 \rceil$
if $x \in L$ **then**
  Run the Phase King algorithm with input bit v_x and with the other nodes in L
  Once an output value was found, broadcast it
  Output: the output of the Phase King algorithm
else
  Wait the number of rounds taken by the Phase King algorithm with $|L|$ nodes and $t-1$ faulty nodes to terminate
  Output: a value received at least $|L| - t + 1$ times, otherwise output the initial input bit

Theorem 1 (Consistency-Robustness)**.** *pred_ba (Algorithm 1) is α-consistent and $\left(\frac{1-\alpha}{2} - 1/n\right)$-robust.*

Proof. We first prove the claim on consistency: we assume that the number of faulty nodes f is lower or equal than $\alpha \cdot n$ and that $P = H$. Since $|H| \geq (1-\alpha) \cdot n$ and Algorithm 1 only adds nodes to P, we have $H \subseteq L$. If $L = H$, there are no faulty nodes in L and the claim holds. Otherwise, by construction from Algorithm 1, we have

$$|L| = \lceil \frac{3}{2} \cdot (1-\alpha) \cdot n \rceil - 1 < \frac{3}{2}(1-\alpha)n \leq \frac{3}{2}|H|.$$

Note that the padded nodes are Byzantine. Since $H \subseteq L$, this implies that $|F \cap L| = |L| - |H| < \frac{|L|}{3}$. Thus, strictly fewer than one third of the nodes are Byzantine inside L, so the Phase King algorithm reaches valid agreement and claimed consistency follows.

We now prove the claimed robustness. We assume that the number of faulty nodes f is lower or equal than $\lfloor \frac{1}{2} \cdot (1-\alpha) \cdot n \rfloor - 1$, the predicted set P is arbitrary. We now use that $|L| \geq \frac{3}{2} \cdot (1-\alpha) \cdot n - 1$ and lower-bound the fraction of faulty nodes in L:

$$|F \cap L| \;\leq\; f \;\leq\; \left\lfloor \frac{1}{2} \cdot (1-\alpha) \cdot n \right\rfloor - 1 \;\leq\; \frac{1}{3} \cdot \left\lfloor \frac{3}{2} \cdot (1-\alpha) \cdot n \right\rfloor - 1 \;\leq\; \frac{|L|}{3} - \frac{2}{3} \;<\; \frac{|L|}{3}.$$

This proves that the Phase King algorithm reaches valid agreement in L and the claim follows. □

The consistency measure implies quite an extreme requirement on the predictor's accuracy and may thus rarely apply, especially when the number of nodes is large and a perfect identification of each node is a challenging task. We now answer the question: how resilient is pred_ba when the predictor is *almost* accurate, say, 99% of the nodes are correctly identified? We answer this question in general, for any fraction of the nodes, by deriving a smoothness curve:

Theorem 2 (Smoothness)**.** *Let $\alpha \in [1/3, 1[$. Let s be the function defined below. For readability, the cases omit an implicit lower bound: $s(\eta)$ is in fact the maximum of the displayed expression and*

the robustness $\frac{1-\alpha}{2} \cdot n - 1$.

$$s(\eta) = \begin{cases} \alpha \cdot n - \eta & \text{if } \eta \in [\![0, (1-\alpha) \cdot n - 1]\!], \\ n - 2\eta - 1 & \text{if } \eta \in \left[\!\left[(1-\alpha) \cdot n - 1, \frac{1+\alpha}{4} \cdot n + 1\right]\!\right], \\ \frac{1-\alpha}{2} \cdot n - 1 & \text{if } \eta \in \left[\!\left[\frac{1+\alpha}{4} \cdot n + 1, n\right]\!\right] \end{cases}$$

Assuming an error of η*, pred_ba (Algorithm 1) with trust parameter* α *reaches valid agreement against* $s(\eta)$ *faulty nodes or fewer.*

Proof Idea. In the following, we present the general proof idea. The full proof of this theorem can be found in Section A.1.

No resilience drop. There is no sudden drop in the resilience as the error increases, neither a critical quantity of mispredictions triggering a drastic phase transition. Instead, the curve is smooth: as long as the number of errors remains within a constant fraction of the nodes (from 0 to $(1-\alpha) \cdot n - 1$), each new misprediction incurs at worst the loss of one unit of resilience. To explain this, we position ourselves on a point of the smoothness curve somewhere between $\eta = 0$ to $\eta = (1-\alpha) \cdot n - 2$. Let C and P be a configuration and a prediction that fit our current error. Now make one of the predicted nodes faulty. Notice that doing this has two simultaneous effects: both the error η and the number of faulty nodes f were incremented by one. Clearly, pred_ba now cannot guarantee the same resilience as before: the faulty node we added into P will also be an active node in L and, as pred_ba was already at its resilience limit before, the faulty nodes are now numerous enough to prevent valid agreement. But what happens if we slightly reduce the number of faulty nodes? We now turn a faulty node into a honest node. By doing so, *the error must change*: if the node is in P then η was decremented by one, otherwise η is incremented. The key argument of the proof is that, in order to keep a constant error, *not only the faulty nodes must change, but also the prediction*; in our case, we must first convert the faulty node (outside of P) into a honest node, and then insert that node inside of P. Although it is now accurately predicted, this new honest node is not enough to balance our initial insertion of a faulty node; this is because, in the non-authenticated case, one requires at least two honest nodes for each faulty node to reach valid agreement. Thus, we turn one other faulty node into a honest node (decrementing f by one overall and causing the claimed decrease rate) which now guarantees valid agreement.

The phase transition. Interestingly, the smoothness curve is not simply linear. Past an error of $\eta = (1-\alpha) \cdot n - 1$, the smoothness curve becomes steeper: in the worst case, each new misprediction now incurs the loss of two units of resilience. Our impossibility result on smoothness Theorem 4 shows that this transition is not only *necessary* (at the transition point, our smoothness curve touches the impossibility zone) but is also *optimal* for a constant fraction of the errors (see Figure 1a). The transition occurs when mispredictions are numerous enough to consummate the initial advantage that comes with perfect predictions. At the consistency point ($\eta = 0$), every predicted node is honest. Then as the error grows, these correctly predicted honest nodes are replaced by faulty ones (in the worst case), eroding that advantage. By the transition point, the first phase (slope -1) has consumed this advantage entirely: all nodes that were originally honest and well-predicted may now be faulty. Beyond this point, the honest-to-faulty ratio in the *predicted set* P is itself at the edge of what valid agreement requires — not only in the active set L. Since P no longer has enough honest nodes to absorb an additional faulty insertion from $L \setminus P$, one extra faulty node must be removed to restore the balance, which explains the steeper slope of -2. □

4 Impossibility Results with Non-Authenticated Channels

This sections shows impossibility results for any learning-augmented algorithm: we prove that our algorithm pred_ba reaches the optimal consistency-robustness trade-off (Theorem 3) and derive impossibility results for the smoothness metric (Theorem 4).

Theorem 3 (Impossibility on Consistency and Robustness)**.** *Let* $\alpha \in [0, 1[$*. Any learning-augmented algorithm for non-authenticated channels with a consistency greater or equal to* α *has a robustness strictly lower than* $\lceil \frac{1-\alpha}{2} \rceil$*. This result is independent from the chosen prediction system.*

Proof idea. We derive this impossibility result with an *indistinguishability argument* [13]: assuming an algorithm is α-consistent, we derive three configurations and a strategy from the faulty nodes

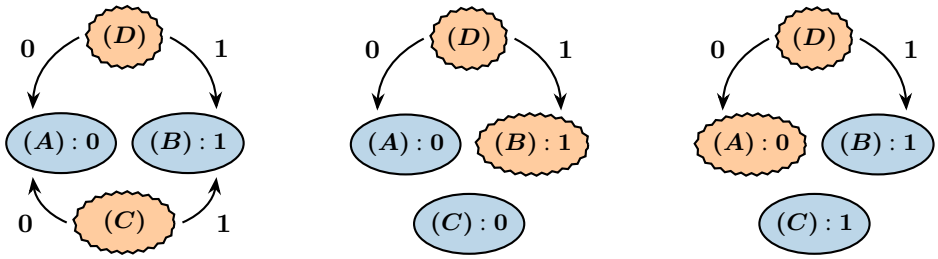


Figure 2: In each configuration the nodes are given the same global prediction $P = A \cup B \cup C$. A missing arrow between a faulty and a honest node means, the behavior of the faulty node towards that honest node is not of importance for the proof. The sizes of the subsets are the following: $|A| = |B| = |C| = \eta$ and $|D| = n - 3 \cdot \eta$

so that, in at least one configuration, either, validity, agreement (see Definition 2) or robustness requirements do not hold. The full proof of this theorem can be found in Section B.1. □

The smoothness metric is inherently interesting because prediction systems are not perfect and will mostly act between the bounds of consistency and robustness. We here show variable impossibility results that limit any algorithm that satisfies optimal consistency robustness tradeoff.

Theorem 4 (Impossibility on Smoothness)**.** *Let $\alpha \in [1/3, 1[$, let ALG be an algorithm that is α-consistent. We consider the function $\bar{s}$ defined as follows.*

$$\bar{s}(\eta) = \begin{cases} \alpha \cdot n + 1 & if\, \eta \in [\![0, \frac{(1-\alpha)}{2} \cdot n]\!], \quad (1) \\ n - 2 \cdot \eta & if\, \eta \in [\![\frac{(1-\alpha)}{2} \cdot n, \frac{1}{3} \cdot n]\!], \quad (2) \\ \frac{1}{2} \cdot n - \frac{1}{2} \cdot \eta - 2 & if\, \eta \in [\![\frac{1}{3} \cdot n, \alpha \cdot n]\!] \quad (3) \end{cases}$$

The smoothness function of ALG is strictly lower than the piece (1) and (2) of $\bar{s}$; moreover, if the smoothness of ALG matches (2) for $\eta \in [\![(1-\alpha) \cdot n, \frac{n}{3}]\!]$, then its smoothness cannot be above (3).

Proof idea. In the following, we present a proof idea for this theorem. The full proof is presented in Section B.2. The proof for each interval of $\bar{s}(\eta)$ works by creating indistinguishable initial configurations. Figure 2 shows the specific configurations for interval (2) in Theorem 4. The figure depicts different subsets of nodes as ovals. Orange ovals with curly borders depict faulty nodes and blue ovals depict honest nodes. The honest nodes further have their input bit written in the oval, whereas the arrows originating from faulty subsets describe the strategy of the faulty nodes. In this case they behave as if they have the specific input bit indicated on the arrow.

The proof for interval (2) works by creating two opposite configurations (b) and (c), where subsets A and B must output 0 and 1 respectively. The impossibility is then created in configuration (a). If the Byzantine nodes in all configurations have a common strategy, they can make subset A believe configuration (a) and (b) are the same. The same holds for subset B with configurations (a) and (c). Because of the indistinguishability subset A must output 0 and subset B must output 1 in (a). This breaks the agreement condition. From this we can see that at least one of the configurations must not work. By carefully choosing the sizes of the subsets and the prediction set, we can create the configurations such that each of them has the same error η and number of faults f. We conclude that no algorithm can solve Byzantine agreement with this resilience for the specific error. □

5 Results in the Authenticated Setting

In this section, we present our results for global prediction in the authenticated setting. We use a similar approach as in the non-authenticated setting: the nodes decide on a set of active nodes L inside which is run a well-known agreement algorithm, in our case the adaptation of the Dolev-Strong algorithm [14] — their algorithm solves a different problem (making node agree on a value that a particular node, the sender, broadcasts to the network) but can be easily adapted to our setting

(see [15] for instance). The algorithm (Algorithm 2) and the proofs can be found in the technical appendix.

Theorem 5. *Let $\alpha \in [1/2, 1[$ and n be the number of nodes. Algorithm 2 with input trust parameter α is α-consistent and $(1 - \alpha - 1/n)$-robust.*

Theorem 6 (Smoothness)**.** *Let $\alpha \in [1/2, 1[$. We consider the function s defined as follows.*

$$s(\eta) = \max\{\alpha \cdot n - \eta, (1 - \alpha) \cdot n - 1\}$$

For an error of η, Algorithm 2 with trust parameter α reaches valid agreement against $s(\eta)$ faulty nodes.

Theorem 7 (Impossibility for Authenticated Consistency and Robustness)**.** *Let $\alpha \in [1/2, 1[$. In the authenticated framework, any learning-augmented algorithm with consistency α has a robustness strictly lower than $\frac{1}{n}\lceil(1 - \alpha)n\rceil$. This result is independent from the chosen prediction system.*

Theorem 8 (Impossibility on Smoothness)**.** *Let $\alpha \in [1/2, 1[$, let ALG be an algorithm that is α-consistent and achieves the optimal $(1 - \alpha - 1/n)$-robustness. We consider the function $\bar{s}$ defined as follows.*

$$\bar{s}(\eta) = \begin{cases} \alpha \cdot n + 1 & \text{if } \eta \in [\![0, (1 - \alpha) \cdot n]\!], \quad (1) \\ n - \eta & \text{if } \eta \in [\![(1 - \alpha) \cdot n, \alpha \cdot n]\!], \quad (2) \end{cases}$$

The smoothness function of ALG is strictly lower than $\bar{s}$.

6 Byzantine Agreement with Local Predictions

So far, we assumed that the prediction is global, i.e., all nodes receive the same predicted set of honest nodes. However, in practice in decentralized applications, different information might be available to different nodes. Consider a monitoring tool in which predictions are produced from locally observed traffic. In this case, nodes would obtain different views of the system and compute different predictions. In the setting with local predictions, the prediction is no longer a single global set, but a collection of sets, one per node.

Definition 7 (Local Prediction)**.** *A local prediction is a vector $\mathbf{P} = (P_1, \ldots, P_n)$ where for every node $i \in [\![n]\!]$, $P_i \subseteq [\![n]\!]$ is the prediction received by node i. The interpretation is that node i receives P_i as input and treats it as the predicted set of honest nodes.*

In contrast to the global case, two honest nodes may now receive different predictions and nodes do not know the predictions received by others. Hence, we update the error function η.

Definition 8 (Error Function for Local Predictions)**.** *Let H be the set of honest nodes in configuration C, and let $\mathbf{P} = (P_1, \ldots, P_n)$ be a set of local predictions. The* prediction error *is*

$$\eta(C, \mathbf{P}) = \sum_{i \in H} (|P_i \setminus H| + |H \setminus P_i|)$$

That is, we measure the total number of mispredicted nodes across all honest nodes. Note that in the worst case, the error function is in order of $\Theta(n^2)$.

Next, we present the main impossibility result for local predictions. This result shows that, unlike for global predictions, there is little room to improve the resilience of algorithms under local predictions.

Theorem 9. *Let $\alpha \in [1/2, 1[$. Under local predictions, any algorithm that is α-consistent is 0-robust. This result holds for both non-authenticated and authenticated frameworks and any prediction system.*

Proof sketch. (Refer to the full proof in Appendix D). The proof uses the fact that nodes can receive different predictions to separate the nodes into two distinct groups, A and B. We first consider the following two configurations: (1) A is honest with input 0 while B is faulty and pretends to have input 1, and (2) A is faulty and pretends to have input 0 while B is honest with input 1. Assume that the algorithm is α-consistent, with $\alpha \geq \frac{1}{2}$: hence, there exists two local predictions, P_1 and P_2 such that, A agree on 0 when given P_1 in configuration (1) and B agree on 1 when given P_2 in configuration (2) (both by validity). Now, consider the following configuration (3): all nodes are honest, A has input 0 and B has input 1. Give P_1 to A and give P_2 to B. A cannot distinguish between (1) and (3); similarly, B cannot distinguish between (2) and (3). Hence, A and B respectively output 0 and 1 in configuration (3), contradicting agreement. □

7 Related Work

Algorithms with predictions. Introduced by Lykouris and Vassilvitskii [4] for online caching, the framework of algorithms with predictions augments algorithms with an additional input — the *prediction* — that may carry useful information about the problem instance. No assumption is made on its quality: it can be perfect, adversarial, or anything in between. The framework has since been applied to data structures, mechanism design, and distributed algorithms (see [16] for an overview), and recently entered distributed computing with the seminal work of Boyar, Ellen, and Larsen [11].

Byzantine agreement. Byzantine agreement was introduced in [1, 2], establishing that up to $t < n/3$ failures can be tolerated without authentication, and up to $t < n/2$ with it [17]. For synchronous communication, $t + 1$ rounds are necessary in the non-authenticated setting [18], a bound later extended to the authenticated case [14]. For asynchronous communication, the FLP impossibility result shows that even a single crash failure cannot be tolerated by a deterministic algorithm [19]. Our algorithms build on the Phase King protocol [13], though any synchronous Byzantine agreement protocol could serve as a drop-in replacement. Byzantine agreement with Predictions was first proposed by Ben-David, Dzulfikar, Ellen and Gilbert [12]. In this work, the authors present how classification predictions can help develop more efficient Byzantine agreement protocols. Note that, unlike in this work, the agreement algorithms in [12] do not provide any robustness guarantees on the resilience of the protocols.

Detection of Byzantine failures. How the detection of faults can help us design more efficient agreement algorithms has been studied under different failure detection mechanisms. Failure detectors have been introduced by Chandra and Toueg [20] to solve consensus with crash failures. The idea is that failure detectors may be unreliable at first, but provide a perfect prediction after some amount of time. Malkhi and Reiter [21] were the first to extend the definition of failure detectors to agreement with Byzantine failures. This work has been later extended in [22, 23]. Another type of detecting Byzantine behavior is via accountability [24]. The notion of accountability has been introduced for dependable distributed systems with the idea to use a scoring system that incentivizes honest behavior and punishes deviations from this behavior. Haeberlen, Kouznetsov, and Druschel [25] presented PeerReview, a system that provides accountability for distributed systems with Byzantine failures. In [26], several reasons are provided for implementing fault detection in distributed systems.

8 Limitations and Future Work

Our work considers synchronous binary Byzantine Agreement with a fixed set of n nodes and an upper bound t on the number of Byzantine faults. Proposed algorithms focus on resilience, not efficiency: their runtime and communication complexity depend on the underlying Byzantine agreement subroutines.

An interesting future direction would be to find algorithms using local predictions with $\alpha > 1/3$ and $\beta > 0$. Based on this, the following open questions arise: Can one design protocols whose round complexity degrades smoothly with prediction error, while their resilience also degrades smoothly? Can prediction accuracy be leveraged to simultaneously approach optimal resilience and optimal time complexity?

Acknowledgements

Research supported by the German Research Foundation (DFG), Schwerpunktprogramm SPP 2378: Resilience in Connected Worlds: Mastering Failures, Overload, Attacks, and the Unexpected, ReNO-2 (511099228), 2025-2029.

References

[1] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[2] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, April 1980.

[3] Christian Cachin and Marko Vukolić. Blockchain consensus protocols in the wild. *arXiv preprint arXiv:1707.01873*, 2017.

[4] Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3302–3311. PMLR, 2018.

[5] Anupam Gupta, Debmalya Panigrahi, Bernardo Subercaseaux, and Kevin Sun. Augmenting online algorithms with $varepsilon$-accurate predictions. *Advances in neural information processing systems*, 35:2115–2127, 2022.

[6] Marek Eliáš, Haim Kaplan, Yishay Mansour, and Shay Moran. Learning-augmented algorithms with explicit predictors. *Advances in Neural Information Processing Systems*, 37:97972–98008, 2024.

[7] Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ml predictions. *Advances in Neural Information Processing Systems*, 31, 2018.

[8] Hailiang Zhao, Xueyan Tang, Peng Chen, and Shuiguang Deng. Learning-augmented algorithms for the bahncard problem. *Advances in Neural Information Processing Systems*, 37:113234–113281, 2024.

[9] Mohammadreza Daneshvaramoli, Helia Karisani, Adam Lechowicz, Bo Sun, Cameron N Musco, and Mohammad Hajiesmaili. Near-optimal consistency-robustness trade-offs for learning-augmented online knapsack problems. In *Forty-second International Conference on Machine Learning*, 2025.

[10] Ziyad Benomar and Christian Coester. Learning-augmented priority queues. *Advances in Neural Information Processing Systems*, 37:124163–124197, 2024.

[11] Joan Boyar, Faith Ellen, and Kim S. Larsen. Brief announcement: Distributed graph algorithms with predictions. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, PODC '25, page 322–325, 2025.

[12] Naama Ben-David, Muhammad Ayaz Dzulfikar, Faith Ellen, and Seth Gilbert. Byzantine agreement with predictions. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, PODC '25, page 3–14, 2025.

[13] P. Berman, J.A. Garay, and K.J. Perry. Towards optimal distributed consensus. In *30th Annual Symposium on Foundations of Computer Science*, pages 410–415, 1989.

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

[15] Ittai Abraham, Srinivas Devadas, Danny Dolev, Kartik Nayak, and Ling Ren. Efficient synchronous byzantine consensus. *arXiv preprint arXiv:1704.02397*, 2017.

[16] Alexander Lindermayr and Nicole Megow. ALPS - algorithms with predictions, 2022. Accessed: 24-Jan-2026.

[17] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, page 218–229, 1987.

[18] Michael J. Fischer and Nancy A. Lynch. A lower bound for the time to assure interactive consistency. *Information Processing Letters*, 14(4):183–186, 1982.

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

[20] Tushar Deepak Chandra and Sam Toueg. Unreliable failure detectors for reliable distributed systems. *J. ACM*, 43(2):225–267, March 1996.

[21] Dahlia Malkhi and Michael Reiter. Unreliable intrusion detection in distributed computations. In *Proceedings of the 10th IEEE Workshop on Computer Security Foundations*, CSFW '97, page 116, USA, 1997. IEEE Computer Society.

[22] Kim Potter Kihlstrom, Louise E. Moser, and P. M. Melliar-Smith. Byzantine fault detectors for solving consensus. *The Computer Journal*, 46(1):16–35, 2003.

[23] Roberto Baldoni, Jean-Michel Hélary, Michel Raynal, and Lenaik Tangui. Consensus in byzantine asynchronous systems. *Journal of Discrete Algorithms*, 1(2):185–210, 2003. SIROCCO 2000.

[24] Aydan R. Yumerefendi and Jeffrey S. Chase. The role of accountability in dependable distributed systems. In *Proceedings of the First Conference on Hot Topics in System Dependability*, HotDep'05, page 3, USA, 2005. USENIX Association.

[25] Andreas Haeberlen, Petr Kouznetsov, and Peter Druschel. Peerreview: practical accountability for distributed systems. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles 2007, SOSP 2007, Stevenson, Washington, USA, October 14-17, 2007*, pages 175–188. ACM, 2007.

[26] Andreas Haeberlen, Petr Kouznetsov, and Peter Druschel. The case for byzantine fault detection. In *Proceedings of the Second Conference on Hot Topics in System Dependability*, HotDep'06, page 5, USA, 2006. USENIX Association.

A Proofs for Resilience Guarantees with Non-Authenticated Channels

A.1 Proof of Theorem 2

Theorem 2 (Smoothness)**.** *Let $\alpha \in [1/3, 1[$. Let s be the function defined below. For readability, the cases omit an implicit lower bound: $s(\eta)$ is in fact the maximum of the displayed expression and the robustness $\frac{1-\alpha}{2} \cdot n - 1$.*

$$s(\eta) = \begin{cases} \alpha \cdot n - \eta & \text{if } \eta \in [\![0, (1-\alpha) \cdot n - 1]\!], \\ n - 2\eta - 1 & \text{if } \eta \in \left[\!\left[(1-\alpha) \cdot n - 1, \frac{1+\alpha}{4} \cdot n + 1\right]\!\right], \\ \frac{1-\alpha}{2} \cdot n - 1 & \text{if } \eta \in \left[\!\left[\frac{1+\alpha}{4} \cdot n + 1, n\right]\!\right] \end{cases}$$

Assuming an error of η, pred_ba (Algorithm 1) with trust parameter α reaches valid agreement against $s(\eta)$ faulty nodes or fewer.

Proof. Let $\alpha \in [1/3, 1[$, we consider Algorithm 1 with trust parameter α. We note $\eta_F = |P \setminus H|$ the number of faulty nodes in P and $\eta_H = |H \setminus P|$ the number of honest nodes that are not in P. Note that $\eta = \eta_F + \eta_H$. We will prove that the claim holds in three cases, one each of the three pieces of the function s.

Case 1. We assume that (1) $\eta \in [\![0, (1-\alpha) \cdot n - 1]\!]$, that (2) the number of faulty nodes f is lower or equal than $\alpha \cdot n - \eta$ and that (3) $s(\eta)$ (i.e., $\alpha \cdot n - \eta$) is strictly larger than the robustness value $(1-\alpha)/2 \cdot n - 1$. Hypothesis (2) implies a lower-bound on the number of honest nodes:

$$|H| \geq (1-\alpha) \cdot n + \eta$$

We now upper-bound the number of faulty nodes in L. A faulty node in L can be either already in P (there are η_F of them) or added in the padding process from P to L. First, we give an upper bound on the number of faulty nodes in P. We first re-express the number of honest nodes inside P:

$$|H \cap P| = |H| - |H \setminus P|$$

Combining the above two inequalities with our initial assumption (1) that $\eta < (1-\alpha) \cdot n$ gives

$$|H \cap P| \quad > \quad 2\eta - \eta_H \quad \geq \quad 2 \cdot \eta_F \implies |P| > 3 \cdot \eta_F$$

This result already proves our claim in the case where the number of padded nodes is zero, that is, if $|P| \geq \frac{3}{2} \cdot (1-\alpha) \cdot n - 1$ (and thus $L = P$) since then strictly fewer than a third of the nodes in L are faulty which guarantees that the King algorithm reaches valid agreement.

We now assume the converse and suppose that the number of padded nodes is strictly positive, which implies that $|L| = \lceil \frac{3}{2} \cdot (1-\alpha) \cdot n \rceil - 1$. From the first inequality, we derive that $|H \cap P| = |H| - \eta_H \geq (1-\alpha) \cdot n$ and thus:

$$|L| = \lceil \frac{3}{2} \cdot (1-\alpha) \cdot n \rceil - 1 < \frac{3}{2} \cdot (1-\alpha) \cdot n \leq \frac{3}{2} |H \cap P|$$

We can now conclude by dividing the nodes of L into three groups, the honest nodes in P, the faulty nodes in P, and the padded nodes (assumed faulty in the worst case):

$$\begin{aligned} |L| = |H \cap P| + \eta_F + |L \setminus P| &< \frac{3}{2} \cdot |H \cap P| \\ \implies 2 \cdot (\eta_F + |L \setminus P|) &< |H \cap P| \\ \implies 3 \cdot (\eta_F + |L \setminus P|) &< |L| \qquad \text{(adding } \eta_F + |L \setminus P| \text{ to both sides)} \end{aligned}$$

We obtain that the number of faulty nodes in L, which is at most the number of faulty nodes in P plus the number of padded nodes (i.e., $\eta_F + |L \setminus P|$) makes up for strictly less than a third of the nodes in L. It follows that the King algorithm run inside L reaches valid agreement, proving the claim.

Case 2. We assume that (1) $\eta \in [\![(1-\alpha)\cdot n - 1, \frac{1+\alpha}{4}\cdot n + 1]\!]$, that (2) the overall number of faulty nodes is strictly smaller than $n - 2\cdot\eta$ and (3) $s(\eta)$ is strictly larger than the robustness value $(1-\alpha)/2\cdot n - 1$. Assumption (2) gives that

$$|H| > 2\cdot\eta$$

Using the identity $|H| = |H\cap P| + \eta_H$ and rearranging the terms gives

$$|H\cap P| \quad > \quad 2\cdot\eta - \eta_H \quad \geq \quad 2\cdot\eta_F.$$

This last inequality enables us to conclude in case $L = P$ as then strictly less than $1/3$ of the nodes in L are faulty which ensures the King algorithm reaches valid agreement. We now assume $P \subsetneq L$ which implies $|P| < \frac{3}{2}\cdot(1-\alpha)\cdot n - 1$. We use the strict lower-bound on $|H|$ obtained at the beginning of this case and obtain

$$|H\cap P| > \eta$$

which, coupled with our initial assumption that $\eta \geq (1-\alpha)\cdot n - 1$, gives

$$|H\cap P| \geq (1-\alpha)\cdot n$$

From this point on, we can use the same computations as the previous case to conclude.

Case 3. We assume that $s(\eta) = (1-\alpha)/2\cdot n - 1$. This case immediately holds by using the robustness of the algorithm (Theorem 1). □

B Proofs for Impossibility Results with Non-Authenticated Channels

B.1 Proof of Theorem 3

Theorem 3 (Impossibility on Consistency and Robustness)**.** *Let $\alpha \in [0,1[$. Any learning-augmented algorithm for non-authenticated channels with a consistency greater or equal to α has a robustness strictly lower than $\lceil\frac{1-\alpha}{2}\rceil$. This result is independent from the chosen prediction system.*

Proof. Let ALG be an algorithm with a consistency no lower than α. We split the nodes into three sets A, B and C such that $|C| = \min\{\lfloor\alpha n\rfloor, n-2\}$, $|A| = \lfloor(n-|C|)/2\rfloor$ and $|B| = \lceil(n-|C|)/2\rceil$. Note that A and B are non-empty and $\max\{|A|,|B|\} = \lceil\frac{1-\alpha}{2}n\rceil$.

We consider the following Configuration (1): nodes in A are honest with inputs 0; nodes in B are honest with inputs 1; nodes in C are faulty, they act as honest nodes with input 0 when interacting with A, and as honest nodes with input 1 when interacting with B. ALG is α-consistent, it therefore exists a prediction that makes ALG reach valid agreement in Configuration (1).

We now exhibit two other configurations, (2) and (3), with $|B|$ and $|A|$ faulty nodes, respectively and show that ALG fails to reach valid agreement in at least one of configuration (1), (2) or (3). We consider configurations (2) and (3) defined as follows: in Configuration 2, nodes in A and C are honest with inputs 0, nodes in B are faulty and run ALG with input 1; in Configuration (3), nodes in B and C are honest with inputs 1, nodes in A are faulty and run ALG with input 0.

In both configurations, all nodes receive the same prediction as in Configuration (1). By validity, all honest nodes must output 0 in Configuration (2) (since all honest inputs are 0) and 1 in Configuration (3) (since all honest inputs are 1). However, A cannot tell (1) and (2) apart; likewise, B cannot tell (1) and (3) apart. As A and B have the same outputs in (1), it follows that they also have the same outputs in (2) and (3), contradicting validity for one of (2) or (3). We showed that ALG is either not α-consistent or cannot tolerate $\lceil\frac{(1-\alpha)n}{2}\rceil$ Byzantine faults under arbitrary predictions. Hence the robustness of any α-consistent algorithm is strictly smaller than $\frac{1}{n}\lceil\frac{(1-\alpha)n}{2}\rceil$, which proves the claim. □

B.2 Proof of Theorem 4

The following proofs use Figure 3, Figure 2 and Figure 4 for visualization. Those figures show subsets of nodes. The orange ovals with curly borders depict Byzantine subsets while the blue ovals depict a subset of correct nodes. Further, an arrow between a faulty subset of nodes and a honest subset of nodes means that the faulty nodes pretend to be honest nodes with input bit the label of the arrow toward the honest subset of nodes.

Theorem 4 (Impossibility on Smoothness)**.** *Let $\alpha \in [1/3, 1[$, let ALG be an algorithm that is α-consistent. We consider the function $\bar{s}$ defined as follows.*

$$\bar{s}(\eta) = \begin{cases} \alpha \cdot n + 1 & if\, \eta \in [\![0, \frac{(1-\alpha)}{2} \cdot n]\!], \quad (1) \\ n - 2 \cdot \eta & if\, \eta \in [\![\frac{(1-\alpha)}{2} \cdot n, \frac{1}{3} \cdot n]\!], \quad (2) \\ \frac{1}{2} \cdot n - \frac{1}{2} \cdot \eta - 2 & if\, \eta \in [\![\frac{1}{3} \cdot n, \alpha \cdot n]\!] \quad (3) \end{cases}$$

The smoothness function of ALG is strictly lower than the piece (1) and (2) of $\bar{s}$; moreover, if the smoothness of ALG matches (2) for $\eta \in [\![(1-\alpha) \cdot n, \frac{n}{3}]\!]$, then its smoothness cannot be above (3).

Proof. For the smoothness argument we are only interested in algorithms that have the optimal consistency-robustness tradeoff. Therefore, only algorithms that reach $\beta = \frac{1-\alpha}{2} \cdot n - 1$ resilience for any prediction are considered. We first show that (1) is a set of impossible error and resilience points based on a variable choice of $\alpha \in [1/3, 1[$. For that we consider three configurations called (3a), (3b) and (3c) depicted in Figure 3. We divide the nodes into four subsets A, B, C and D and a single node x. The size of D is variable and the size of C is determined by $|C| = \alpha \cdot n - |D| - 1$. The subsets A and B are of equal size, i.e. $|A| = |B| = \frac{(1-\alpha)\cdot n - 1}{2}$. As all subsets must have non-negative sizes the following additional bound applies: $|D| \leq \alpha \cdot n - 1$. Each configuration of Figure 3 is defined with these five subsets.

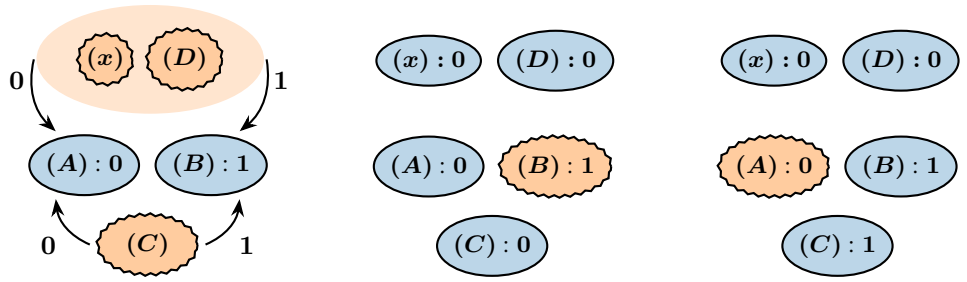
We first show that any algorithm fails in at least one of the configurations (3a), (3b) or (3c) with the same input prediction. Let ALG be an algorithm, we assume that ALG reaches valid agreement in configuration (3b) and (3c). We now use a standard indistinguishability argument: the nodes in A cannot distinguish between (3a) and (3b) while the nodes in B cannot distinguish between (3a) and (3c). However, the nodes in A must output 0 in (3b) to obey validity; for the same reason, the nodes in B must output 1 in (3c). It follows that ALG does not reach valid agreement in (3a), proving the subclaim.

Using what we now proved we show no algorithm can have a resilience better than $\alpha \cdot n$ if it should still satisfy optimal $(\frac{1-\alpha}{2} - 1)$-robustness. For contradiction we assume $(\alpha \cdot n + 1)$-resilience for $\eta > 0$. Let $\eta_a = |D| + 1$ which means the Byzantine subset D and the single Byzantine node x were mispredicted as honest nodes in Figure 3a. We now see the number of Byzantine nodes in (3b) and (3c) is $f = \frac{(1-\alpha)\cdot n - 1}{2}$. Thus we can maximally achieve a robustness of $\beta = \frac{(1-\alpha)\cdot n - 1}{2} - 1$ which is strictly lower than the $(\frac{(1-\alpha)\cdot n}{2} - 1)$-robustness we want to achieve.

We now show that (2) is a set of impossible error and resilience points. For that, we consider three configurations called (2a), (2b) and (2c) depicted in Figure 2. We divide the nodes into four subsets called A, B, C and D such that D is of variable size and the rest of the nodes are evenly shared among the other subsets, i.e. $|A| = |B| = |C|$. Each configuration of Figure 2 is defined with those four subsets. We can use the same argument as before to show that any algorithms fails in at least one of the configurations (2a), (2b) or (2c) with the same input prediction.

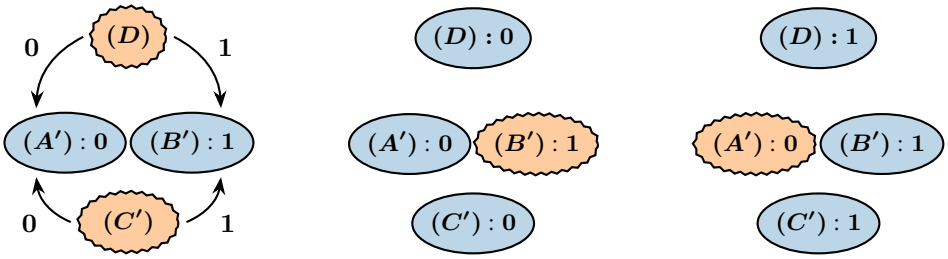
Using what we just proved, we now show no algorithm can have a smoothness $n - 2 \cdot \eta$. Let $\eta = |A|$, notice that each configuration (2a), (2b) and (2c) have error η. The number of faulty nodes is also the same in those three configurations and equals $\eta + |D|$. Noticing that $|D| = n - 3 \cdot \eta$, the number of faulty nodes can be rewritten as $n - 2 \cdot \eta$, proving the claim.

We now show that, if ALG matches (2) (that is, if it has smoothness $n - 2 \cdot \eta - 1$ for $\eta \in [\![n(1-\alpha), n/3]\!]$), then ALG cannot have smoothness (3). We define three configurations (4a), (4b) and (4c) shown in Figure 4. We slightly changed the sets A', B' and C' compared to A, B and C from the above proof for (2) as follows: we took two nodes away from C and inserted one in A and the other in B. Assuming that ALG matches (2), ALG reaches valid agreement in configuration (4a) as, after the slight changes on the subsets, this is strictly below (2). Notice that A' must output 0 in (4b) but cannot distinguish between (4a) and (4b); likewise, B' must output 1 in (4c) but cannot distinguish between (4a) and (4c), it follows that ALG does not respect validity on either (4b) or (4c).



(a) (b) (c)

Figure 3: In each configuration the nodes are given the same global prediction $P = A \cup B \cup D \cup x$. A missing arrow between a faulty and a honest node means, the behavior of the faulty node towards that honest node is not of importance for the proof.



(a) (b) (c)

Figure 4: In each configuration the nodes are given the same global prediction $P = A' \cup B' \cup C'$. Compared to Figure 2, the set C' has two fewer nodes than C and the sets A' and B' each have one more node than respectively A and B.

We now show that ALG cannot have smoothness $n/2 - 1/2 \cdot \eta - 2$ when $\eta \geq n/3$. For that, note that the error η is the same in both configurations (4b) and (4c). We compute:

$$\eta = |D| + |A| = |D| + \frac{n - |D|}{3} + 1 = \frac{n}{3} + \frac{2}{3} \cdot |D| + 1$$

Moreover, both configurations (4b) and (4c) also have the same number of faulty nodes $f = (n - |D|)/3 + 1 = n/2 - 1/2 \cdot \eta - 2$, the claim follows. □

C Reaching Valid Agreement with Authenticated Channels

Algorithm 2: auth_pred_ba(v_x, P, α)

```
// This algorithm runs in node x
```
Input: the input bit v_x, prediction set $P \subseteq [\![n]\!]$, trust parameter $\alpha \in [1/2, 1[$

$L \leftarrow P$
$y \leftarrow 1$
if $|L| < 2 \cdot (1-\alpha) \cdot n - 1$ **then**
  repeat
    if $y \notin L$ **then**
      $L \leftarrow L \cup \{y\}$
    $y \leftarrow y + 1$
  until $|L| \geq 2 \cdot (1-\alpha) \cdot n - 1$;
$t \leftarrow \lceil |L|/2 \rceil$
if $x \in L$ **then**
  Run the Dolev-Strong algorithm among the nodes in L.
  Once Dolev-Strong terminated, broadcast the output.
  Output: the output of the Dolev-Strong algorithm
else
  Wait the number of rounds taken by Dolev-Strong with $|L|$ nodes and $t-1$ faulty nodes to terminate.
  Output: a value received at least $|L| - t + 1$ times, otherwise output the initial input bit

Theorem 5. *Let $\alpha \in [1/2, 1[$ and n be the number of nodes. Algorithm 2 with input trust parameter α is α-consistent and $(1 - \alpha - 1/n)$-robust.*

Proof. Let $\alpha \in [1/2, 1[$. We first show the claim about consistency. We assume that the number of faulty nodes $f = |F|$ is lower or equal to $\alpha \cdot n$ and that the prediction is perfect, i.e., $P = H$. We have that $|P| = |H| \geq (1-\alpha) \cdot n$, hence Algorithm 2 inserts strictly fewer than $(1-\alpha) \cdot n$ nodes in P to obtain L. It follows that strictly more than half of the nodes in L are honest, hence the Dolev-Strong algorithm reaches valid agreement and the claim on consistency follows.

Now on robustness: we assume that there are at most $(1-\alpha) \cdot n - 1$ faulty nodes, the predicted set P is arbitrary. As the set L has size no lower than $\lceil 2(1-\alpha) \cdot n \rceil - 1$, we deduce that strictly more than half of the nodes of L are honest and the claim on robustness follows. □

Theorem 6 (Smoothness)**.** *Let $\alpha \in [1/2, 1[$. We consider the function s defined as follows.*

$$s(\eta) = \max\{\alpha \cdot n - \eta, (1-\alpha) \cdot n - 1\}$$

For an error of η, Algorithm 2 with trust parameter α reaches valid agreement against $s(\eta)$ faulty nodes.

Proof. Let $\alpha \in [1/2, 1[$, we consider Algorithm 2 with trust parameter α. We call η_F the number of faulty nodes in P and η_H the number of honest nodes not in P. By definition, it holds that $\eta = \eta_F + \eta_H$. Clearly, the desired claim holds when $s(\eta) = (1-\alpha) \cdot n - 1$ as a consequence of the algorithm's robustness (Theorem 5). The rest of this proof shows that the claim holds when $s(\eta) > (1-\alpha) \cdot n - 1$. Note that this implies that $\eta \leq \lceil (2\alpha - 1) \cdot n \rceil$ and that the number of faulty nodes f is no greater than $\alpha \cdot n - \eta$. We distinguish between two cases: (1) either the algorithm does not pad P with additional nodes to obtain L (i.e., $P = L$), or (2) the algorithm adds one or more nodes to P (i.e., $P \subsetneq L$).

Case 1. We assume that $P = L$. We first show that the number of honest nodes $|H|$ is strictly greater than η:

$$\begin{aligned} & f \leq \alpha \cdot n - \eta \\ \implies & |H| \geq (1-\alpha) \cdot n + \eta \\ \implies & |H| > \eta \end{aligned}$$

where the last inequality holds since we assumed $\alpha < 1$. We use it to lower-bound the number of honest nodes in L:

$$|H \cap L| = |H| - \eta_H = |H| - \eta + \eta_F < \eta_F$$

The honest nodes are therefore in a strict majority in L and the Dolev-Strong algorithm therefore reaches valid agreement.

Case 2. We assume that $P \subsetneq L$, i.e., the algorithm padded P with a positive number of nodes to obtain L. This implies that $|P| < 2 \cdot (1-\alpha) \cdot n - 1$ and therefore $|L| = 2 \cdot (1-\alpha) \cdot n - 1$. We lower-bound the number of honest nodes inside L:

$$\begin{aligned} |H \cap L| &\geq |H \cap P| \\ &= |H| - \eta_H \\ &\geq |H| - \eta \\ &\geq (1-\alpha) \cdot n \end{aligned}$$

which implies that

$$2|H \cap L| \geq 2(1-\alpha) \cdot n > |L|$$

The honest nodes are in a majority in L and the Dolev-Strong algorithm therefore also reaches valid agreement, the claim follows. □

Theorem 7 (Impossibility for Authenticated Consistency and Robustness)**.** *Let $\alpha \in [1/2, 1[$. In the authenticated framework, any learning-augmented algorithm with consistency α has a robustness strictly lower than $\frac{1}{n}\lceil(1-\alpha)n\rceil$. This result is independent from the chosen prediction system.*

Proof. Let ALG be an algorithm that is α-consistent. We split the nodes in two subsets A and B such that $|A| = \lfloor \alpha \cdot n \rfloor$ and $|B| = \lceil (1-\alpha) \cdot n \rceil$. We consider three configuration, Configuration (1), (2) and (3).

In Configuration (1), the nodes of B are honest with input 0 while the nodes of A are faulty; the nodes receive a prediction P such that ALG reaches valid agreement (α-consistency), the nodes of A play as if they had input 1 and received prediction P. In Configuration (2), the nodes of B are faulty and the nodes of A are honest with input 1; the nodes receive the same prediction P as in Configuration (1), the nodes of B play as if they had input 0 and received prediction P. In Configuration (3), finally, all the nodes are honest and receive prediction P, the nodes of A have input 1 and the nodes of B have input 0.

We see that B cannot distinguish between Configurations (1) and (3); similarly, A cannot distinguish between Configurations (2) and (3). B must output 0 in Configuration (1) and A must output 1 in Configuration 2, which prevents from reaching valid agreement in Configuration (3). ALG fails to solve the problem in either Configuration 2 or Configuration 3 which shows ALG is not $\frac{1}{n}\lceil(1-\alpha)n\rceil$-robust, proving the claim. □

Theorem 8 (Impossibility on Smoothness)**.** *Let $\alpha \in [1/2, 1[$, let ALG be an algorithm that is α-consistent and achieves the optimal $(1-\alpha-1/n)$-robustness. We consider the function $\bar{s}$ defined as follows.*

$$\bar{s}(\eta) = \begin{cases} \alpha \cdot n + 1 & \text{if } \eta \in [\![0, (1-\alpha) \cdot n]\!], \quad (1) \\ n - \eta & \text{if } \eta \in [\![(1-\alpha) \cdot n, \alpha \cdot n]\!], \quad (2) \end{cases}$$

The smoothness function of ALG is strictly lower than $\bar{s}$.

Proof. Let ALG be an algorithm that is α-consistent and should satisfy optimal $(1-\alpha-\frac{1}{n})$-robustness. We first show that (1) is a set of impossible error and resilience points based on a variable choice of $\alpha \in [1/2, 1[$. For that we consider three configurations called (5a), (5b) and (5c) depicted in Figure 5. We divide the nodes into three subsets A, B and C and a single node x. The size of C is variable and the size of B is determined by $|B| = \alpha \cdot n - C$. The subset A is of size $(1-\alpha) \cdot n - 1$. As all subsets must have non-negative sizes the size of C is limited by $|C| \leq \alpha \cdot n - 1$.

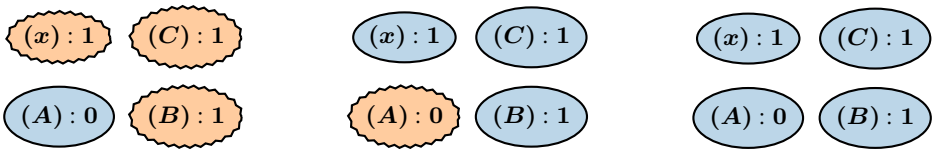


Figure 5: In each configuration the nodes are given the same global prediction $P = A \cup C \cup x$.

We first show that any algorithm fails in at least one of the configurations (5a), (5b) and (5c) with the same input prediction. We see that A cannot distinguish between configurations (5a) and (5c); similarly B cannot distinguish between configurations (5b) and (5c). A must output 0 in configuration (5a) and B must output 1 in configuration (5b) which prevents from reaching valid agreement in configuration (5c).

Using what we now proved we show no algorithm can have a resilience better that $\alpha \cdot n$ if it should still satisfy optimal $(1 - \alpha - \frac{\eta}{n})$-robustness. For contradiction we assume $(\alpha \cdot n + 1)$-resilience for $\eta > 0$. Let $\eta_a = |C| + 1$ which can be understood as a misprediction of the Byzantine subset C and the single node x in configuration (5a). We now see the number of Byzantine nodes in (5b) is $f = (1 - \alpha) \cdot n - 1$. Thus we can maximally achieve a robustness of $\beta = (1 - \alpha) \cdot n - 2$ which is strictly lower than the $((1 - \alpha) \cdot n - 1)$-robustness we want to achieve. From the limits on $|C|$ we can also conclude $\eta_a \leq \alpha \cdot n$.

We now show that (2) is a set of impossible error and resilience points. For that we consider three configurations (a), (b) and (c). We divide the nodes into three subsets A, B and C such that $|C| = c$ and the rest of the nodes are evenly shared between A and B, i.e., $|A| = |B|$. Let the size of C be variable with $|C| \in [0, \alpha \cdot n]$ Each configuration is slightly different and defined as follows. In configuration a, the nodes of C are faulty, the nodes of A are honest with input 0, the nodes of B are honest with input 1, the nodes are given as input a prediction P that make ALG reach valid agreement in this configuration (α-consistency of ALG); the nodes of C are playing as if they had input 0 and prediction P with the nodes of A, and as if they had input 1 and prediction P with the nodes of B. In configuration b, the nodes of B are faulty and play as if they had input 1 with prediction P, the other nodes have input 0 and prediction P. In configuration c, the nodes of A are faulty and play as if they had input 0 with prediction P, the other nodes have input 1 and prediction P.

ALG reaches valid agreement in configuration a as it is α-consistent. However, the nodes of A cannot distinguish between configurations a and b, likewise, the nodes of B cannot distinguish between configurations a and c: ALG does not solve the problem for one of the configurations a and b.

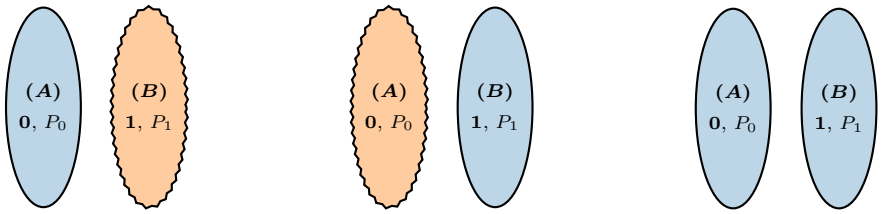
We now show our claim. Notice that configuration a and b both have an error of $\eta = (n + |C|)/2$ and a number of faulty nodes $f = (n - |C|)/2$. We showed that ALG cannot solve the problem against $n - \eta$ faulty nodes, proving the claim. □

D Proof of Theorem 9

Theorem 9. *Let $\alpha \in [1/2, 1[$. Under local predictions, any algorithm that is α-consistent is 0-robust. This result holds for both non-authenticated and authenticated frameworks and any prediction system.*

Proof. Let $\alpha \in [1/2, 1[$, we split the nodes in two sets A and B of equal size assuming n is divisible by two. We consider an algorithm ALG which is α-consistent. Notice that, in particular, ALG is $1/2$-consistent.

We consider the following predictions, (see Figure 6):



(a) Configuration 1 (b) Configuration 2 (c) Configuration 3

Figure 6: Different indistinguishable configurations where P_0 is a consistency prediction for Configuration 1 and P_1 is a consistency prediction for Configuration 2. No Algorithm that reaches agreement in Configurations 1 and 2 can reach agreement in Configuration 3.

- P_0 is a prediction such that ALG reaches valid agreement when the nodes of B are faulty and the nodes of A are honest and have input bit 0.
- P_1 is a prediction such that ALG reaches valid agreement when the nodes of A are faulty and the nodes of B are honest and have input bit 1.

We now consider the following configurations:
Configuration 1:

- Nodes in A are honest with input bit 0 and prediction P_0.
- Nodes in B are faulty nodes and play as if they had input bit 1 and prediction P_1.

Configuration 2:

- Nodes in A are faulty nodes and play as if they had input bit 0 and prediction P_0.
- Nodes in B are honest with input bit 1 and prediction P_1.

Configuration 3:

- Nodes in A are honest with input bit 0 and prediction P_0.
- Nodes in B are honest with input bit 1 and prediction P_1.

As we assumed ALG is α-consistent, the nodes of A agree on 0 in Configuration 1 while the nodes of B agree on 1 on configuration 2. The nodes of A however cannot distinguish between configurations 1 and 3, the same goes for the nodes of B between configurations 2 and 3. The nodes of A and B therefore have different output in Configuration 3 even though there are no faulty nodes, ALG is therefore 0-robust. □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the introduction, the problem statement as well as the main questions that are answered in this work are stated. Section 1.1 in the introduction clearly states the contributions of this paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the Section 2, all the assumptions made in this paper are formally stated. Section 8 discusses the limitations of our model and contribution and outlines possible future directions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The model assumed in the paper is fully described in Section 2. The theorems and lemmas presented in this paper state the model assumptions needed for the respective result. In the main part of the paper, proofs appear for selected results and otherwise proof sketches are presented. In the latter case, full proofs are presented in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [N/A]

Justification: This is a theoretical paper that does not include experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [N/A]

   Justification: This is a theoretical paper that does not include experiments.

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [N/A]

   Justification: This is a theoretical paper that does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: This is a theoretical paper that does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: This is a theoretical paper that does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This is a theoretical contribution that aims to improve the fault-tolerance of Byzantine Agreement protocols (against potentially harmful attacks). As there are no experiments, there are no potential harms caused by the research process or any potential societal impacts. The results in the paper are clearly stated for potential reproducibility or follow-up work.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is a theoretical contribution that does not have any societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This is a theoretical contribution without any experimental part.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: This is a theoretical contribution with original results.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This is a theoretical contribution that does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The research presented in this paper has been derived by the authors of the paper without the use of LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.