Test-Time Collective Classification over Multi-Agent Networks

Ping Hu
EPFL
ping.hu@epfl.ch

Mert Kayaalp
UBS-IDSIA AI Lab
mert.kayaalp@idsia.ch

Ali H. Sayed
EPFL
ali.sayed@epfl.ch

Abstract

Motivated by the challenges of joint training in heterogeneous multi-agent systems and by the benefits of collective decision-making observed in the social sciences, we propose a framework for test-time collaboration among independently trained agents. We study a distributed binary classification task in which agents, potentially differing in architecture, feature space, and modality, must coordinate to produce collective predictions. This coordination is achieved at test time by exchanging local beliefs through a decentralized decision-making protocol. We analyze the generalization performance of this collective classification framework and establish theoretical error bounds. The results quantify the cost of independent training, demonstrate the benefits of collective action, and reveal how network structure and aggregation rules shape classification accuracy.

1 Introduction

Collaboration in multi-agent systems has often been coupled with collaborative learning (e.g., federated learning [1], multi-agent reinforcement learning [2]). Yet in many applications, joint training is impractical [3]. In sociotechnical settings, for instance, human involvement can preclude learning a fixed, pretrained collaboration policy. Motivated by this observation, in this work, we consider a two-phase framework for distributed classification in which agents learn independently on local data and collaborate only at test time to infer a common label.

Specifically, during training, each agent independently trains a classifier on its own private dataset, potentially using different model architectures and input modalities. This reflects trends and constraints in modern ML: independent training is increasingly common in large-scale, privacy-sensitive settings such as edge deployments [4] and foundation-model pipelines [5]. For example, LLMs and domain-specific experts are often trained separately across companies, clients, or devices and coordinated only at inference time [6]. During the testing phase, each agent receives a private input that is conditionally independent given the true class, and exchanges predictions with its graph neighbors to collectively infer the label.

1.1 Main contributions

We formalize the *independent training + collaborative inference* paradigm for supervised classification in multi-agent networks. Our framework allows heterogeneous models across agents, each trained on possibly private and distinct data modalities. To enable collaboration during test time without raw data exchange and a centralized controller, we employ the DeGroot model from social sciences [7], originally proposed to describe opinion consensus formation by iterative local averages, to the multi-agent ML setting by treating agents' soft predictions (e.g., class probability vectors) as the counterparts of opinions. We address three questions in this work:

39th Conference on Neural Information Processing Systems (NeurIPS 2025)
Workshop on Algorithmic Collective Action (ACA@NeurIPS 2025).

- *How does independent training affect performance?* In Lemma 1, we compare independent local training with centralized training through the consistent training condition. We establish probability bounds under both setups, which highlight how the number of training samples per agent characterizes the performance gap. The centralized bound is strictly stronger and captures the price of independent training in the form of weaker consistency guarantees.
- *Is collective action beneficial to individual agents?* In Theorem 1, we compare cooperative and non-cooperative inference by deriving error bounds for both settings. The analysis shows that cooperation leads to strictly tighter guarantees on the classification error and relaxes the conditions required for consistency. These results demonstrate that collective action not only improves predictive performance but also enables learning when the classification task is locally infeasible for some agents.
- *What if the communication rounds are limited?* In Theorem 2, we establish the error bounds for collaborative inference with a finite number of communication rounds. Unlike the collective prediction under sufficient communication, these bounds are agent-dependent and explicitly reflect how network topology and combination weights influence the guarantees.

This work contributes to the theme of algorithmic collective action (ACA), where algorithms mediate coordination among agents/participants in complex systems [8]. Here, coordination occurs through communication among distributed learners: although each agent has only local data and an independently trained model, structured interactions shaped by network topology and combination weights enable the group to achieve improved collective prediction. To avoid confusion, we note that this setting differs from the formal ACA definition commonly used in the ML literature, which refers to scenarios where a subset of external users strategically modifies their data contributions to influence a platform's learning outcome [9]. Nevertheless, both perspectives share an interest in understanding how individual algorithmic behaviors combine to produce collective outcomes, situating our work within the broader study of collective dynamics in algorithmic systems.

1.2 Related literature

Our test-time collective classification framework is closely related to the distributed ML literature. Prior work in this area has largely focused on collaboratively training a shared global model using data distributed across multiple devices or computational nodes. Two dominant approaches are federated learning, which relies on a central server to aggregate locally trained updates [10, 11], and decentralized learning, which dispenses with the server and synchronizes models through peer-to-peer communication over a fixed network graph [12, 13]. Despite topological differences, both paradigms aim to produce a single global model during training. At inference time, this model is either executed on a central server or deployed across devices, so the system effectively operates as a centralized decision-maker. A small body of work has explored collaboration at inference time, where a set of independently trained models exchange information with each other to improve accuracy via trust-score design [3, 14]. However, they all assume that all agents observe the same test input, which restricts agents' heterogeneity to model parameters or inductive biases. This setup closely parallels ensemble learning and multiple classifier combination discussed later.

In contrast, we study a setting where each agent receives its own private testing sample, assumed conditionally independent given the class label. This formulation naturally accommodates heterogeneous feature spaces (e.g., multimodal sensors or domain-specific models) and enables distributed decision-making without requiring feature alignment. Collaborative inference under this setting has received limited attention, and our work provides a formal algorithmic and theoretical treatment.

Another related line of research is multiple classifier combination, which studies how to aggregate the predictions of multiple models to improve performance. While our framework is distributed and collaborative in nature, it bears a strong conceptual connection to this literature. Specifically, when all agents follow the DeGroot model for information aggregation at test time, their predictions converge (in the limit of infinite communication) to a consensus that is a weighted average of the local classifier outputs. This resembles the behavior of a centralized fusion rule, a central topic in the literature on multiple classifier combination. Classical techniques in this area include voting schemes, bagging, boosting, and stacking [15, 16]. These methods form the foundation of ensemble learning, which combines multiple base classifiers to improve generalization and robustness [17, 18]. More advanced approaches, such as mixture-of-experts models, perform input-dependent aggregation through gating networks [5]. These techniques, however, typically operate in a centralized setting during test time and assume a common feature representation across experts.

In contrast, our framework operates in a fully-distributed environment during both training and testing phases. When communication is limited to a finite number of rounds, consensus may not be reached, and predictions remain agent-dependent. This dynamic, iterative process differs fundamentally from the static, one-shot aggregation used in most ensemble methods.

Notation: We use boldface font to denote random variables and normal font for their realizations, e.g., $\boldsymbol{h}$ and h. $\mathbb{E}$ and $\mathbb{P}$ denote the expectation and probability operators, respectively.

2 Problem formulation

We are given a network of K agents indexed by k and a binary classification task. The agents and class labels are denoted by $\mathcal{K} \triangleq \{1, 2, \ldots, K\}$ and $\Gamma \triangleq \{+1, -1\}$, respectively. Each agent k holds a local training set $\mathcal{D}_k$ of N_k labeled pairs $(\boldsymbol{h}_{k,n}, \boldsymbol{\gamma}_{k,n})$, with feature space $\mathcal{H}_k$ and class conditional distribution $p_k(h|\gamma)$ for each $\gamma \in \Gamma$. The agents are *heterogeneous* in that they may differ in feature spaces $\mathcal{H}_k$ or distributions $p_k(h|\gamma)$ given the same label. We assume a uniform prior over Γ. For binary classification, the logit function (i.e., the log-posterior ratios) is sufficient for prediction.

In the training phase, each agent trains a classifier *independently* to learn its local logit function. The output of this classifier is characterized by the function f_k. The optimal function $\boldsymbol{f}_k^o$ is learned using the training set $\mathcal{D}_k$ via the empirical risk minimization principle:

$$\boldsymbol{f}_k^o = \operatorname*{arg\,min}_{f_k \in \mathcal{F}_k} \boldsymbol{R}_{k,\mathsf{emp}}(f_k) \triangleq \frac{1}{N_k} \sum_{n=1}^{N_k} \Phi(\boldsymbol{\gamma}_{k,n} f_k(\boldsymbol{h}_{k,n})) \tag{1}$$

where $\mathcal{F}_k : \mathcal{H}_k \mapsto \mathbb{R}$ is the admissible family determined by the classifier structure, and $\Phi : \mathbb{R} \mapsto \mathbb{R}_+$ is the loss function. We impose some assumptions on Φ and $\mathcal{F}_k$ during training (see Appendix A.2). The training phase is completed by implementing a debiasing operation [19] to $\boldsymbol{f}_k^o$ to mitigate the biases incurred during solving (1). Specifically, given any $h \in \mathcal{H}_k$, the (real-valued) output of the classifier is

$$\boldsymbol{c}_k(h) \triangleq \boldsymbol{f}_k^o(h) - \boldsymbol{\mu}_{k,\mathsf{emp}}(\boldsymbol{f}_k^o), \quad \text{where } \boldsymbol{\mu}_{k,\mathsf{emp}}(\boldsymbol{f}_k^o) \triangleq \frac{1}{N_k} \sum_{n=1}^{N_k} \boldsymbol{f}_k^o(\boldsymbol{h}_{k,n}). \tag{2}$$

In the prediction phase, each agent receives a testing sample $\boldsymbol{h}_k^* \in \mathcal{H}_k$ whose true label is $\boldsymbol{\gamma}^*$. The goal of the agents is to make a *collective* prediction about $\boldsymbol{\gamma}^*$ by iteratively updating their real-valued decision statistics through communication with neighbors. Let t index the communication round during this collaboration process, and denote by $\boldsymbol{\lambda}_{k,t}$ the decision statistic of agent k at round t. The DeGroot model for information aggregation gives the following recursion for each agent k at each round t [7]:

$$\boldsymbol{\lambda}_{k,t} = \sum_{\ell=1}^{K} a_{\ell k} \boldsymbol{\lambda}_{\ell,t-1}, \quad \text{where } \boldsymbol{\lambda}_{\ell,0} \triangleq \boldsymbol{c}_\ell(\boldsymbol{h}_\ell^*), \forall \ell \in \mathcal{K}. \tag{3}$$

Here, $a_{\ell k}$ is the combination weight agent k assigns to its neighbor ℓ, which satisfies $\sum_{\ell=1}^{K} a_{\ell k} = 1$, $a_{\ell k} > 0 \ \forall \ell \in \mathcal{N}_k$, and $a_{\ell k} = 0 \ \forall \ell \notin \mathcal{N}_k$, where $\mathcal{N}_k$ denotes the neighboring set of agent k. We assume that the communication network is strongly connected, such that the matrix $A = [a_{\ell k}]$ admits a Perron vector π with $A\pi = \pi$, $\sum_{k=1}^{K} \pi_k = 1$, and $\pi_k > 0, \forall k \in \mathcal{K}$ (see Appendix A.2).

To evaluate the performance of this framework, we also introduce a feasibility condition $\mathsf{R}^o < \Phi(0)$ associated with the classification task (see Appendix A.2), defined in terms of the target risk:

$$\mathsf{R}^o \triangleq \sum_{k=1}^{K} \pi_k \mathsf{R}_k^o, \quad \text{where } \mathsf{R}_k^o \triangleq \inf_{f_k \in \mathcal{F}_k} \mathbb{E}_{(\boldsymbol{h}_{k,n}, \boldsymbol{\gamma}_{k,n})} \Phi(\boldsymbol{\gamma}_{k,n} f_k(\boldsymbol{h}_{k,n})). \tag{4}$$

This condition ensures that the local feature vectors are collectively informative for the classification task, but no single agent's feature vector is required to be sufficiently informative on its own.

3 Main results

In this section, we analyze the classification performance of the proposed framework. Following the distributed averaging rule (3), it is known from [7] that all agents will agree on a *common* decision statistic denoted by $\boldsymbol{\lambda}_{\mathsf{ave}}$ after *sufficient* communication. That is, it holds almost surely (a.s.) that

$$\boldsymbol{\lambda}_{\mathsf{ave}} \triangleq \lim_{t\to\infty} \boldsymbol{\lambda}_{k,t} \overset{\text{a.s.}}{=} \sum_{\ell=1}^{K} \pi_\ell \boldsymbol{c}_\ell(\boldsymbol{h}_\ell^*). \tag{5}$$

With a consensus on the decision statistic, the agents make a collective prediction $\widehat{\boldsymbol{\gamma}} \in \Gamma$ by

$$\widehat{\boldsymbol{\gamma}} \triangleq \operatorname{sgn}(\boldsymbol{\lambda}_{\mathsf{ave}}). \tag{6}$$

We now turn to the performance of this collective decision $\widehat{\boldsymbol{\gamma}}$. As a first step, we examine the training performance of the network classifiers via a consistent training condition. This analysis provides a key ingredient for establishing the accuracy of $\widehat{\boldsymbol{\gamma}}$ and, at the same time, allows us to quantify the price of independent training, which is an important question in collaborative learning.

3.1 Price of independent training

For a given training process that results in a set of K trained models f_k, one for each agent k, we evaluate its performance based on the consistency of the classifier network in the large-sample limit. Specifically, we examine the following condition for the *training* phase:

$$\boldsymbol{\mu}^+(f) > \delta \text{ and } \boldsymbol{\mu}^-(f) < -\delta \tag{7}$$

where $\delta > 0$ is a positive constant referred to as the *decision margin*, and f denotes dependence on the collection of f_k. The formal definitions of $\boldsymbol{\mu}^+(f)$ and $\boldsymbol{\mu}^-(f)$ are provided in Appendix A.1. We will refer to condition (7) as the *δ-margin consistent training condition* in the sequel. The performance of training is then evaluated by the probability of satisfying this condition:

$$P_{c,\delta} \triangleq \mathbb{P}\big(\boldsymbol{\mu}^+(\boldsymbol{f}^o) > \delta, \boldsymbol{\mu}^-(\boldsymbol{f}^o) < -\delta\big) \tag{8}$$

where we use the notation $\boldsymbol{f}^o$ for the optimal models obtained via (1) during the training phase. To assess the cost of independent training in our framework, we compare $P_{c,\delta}$ under distributed local training and centralized training. We focus on homogeneous agents that share the same feature space and model class. In the heterogeneous case, agents train on different features with different models, which makes a centralized benchmark ill-defined. Suppose a total of N_{tol} samples is available to the network. In centralized training, all samples are used to train a single classifier, whereas in distributed training they are partitioned into disjoint subsets $\mathcal{D}_k$, with agent k training on N_k samples. The key distinction is therefore the number of training samples per classifier. Using a result from [20], we obtain the following comparison between centralized and distributed independent training.

Lemma 1 (**Price of independent training**)**.** *Let ρ and ρ^{cen} denote the Rademacher complexities of the classifier network under distributed and centralized training, respectively. Assume that $0 \le \delta < \delta_{\max}$ and $\rho, \rho^{\mathsf{cen}} < \mathcal{E}_\Phi(\mathsf{R}^o, \delta)$, where $\delta_{\max}$ and $\mathcal{E}_\Phi(\mathsf{R}^o, \delta)$ are two constants determined by the given task. Under Assumptions 1–4 in Appendix A.2, the probability $P_{c,\delta}$ in* (8) *admits the following guarantees:*

- *For* independent training*, where each agent k trains on its own dataset of size N_k, we have*

$$P_{c,\delta} \ge 1 - 2\exp\left\{-\frac{8N_{\max}}{\alpha^2\beta^2}\Big(\mathcal{E}_\Phi(\mathsf{R}^o, \delta) - \rho\Big)^2\right\}, \tag{9}$$

  with $N_{\max} = \max_k N_k$, $\alpha \triangleq \sum_{k=1}^{K} \frac{\pi_k N_{\max}}{N_k}$, and β the bound of f_k from Assumption 2.

- *For* centralized training*, where all N_{tol} samples are used to train a single classifier, the bound becomes*

$$P_{c,\delta} \ge 1 - 2\exp\left\{-\frac{8N_{\mathsf{tol}}}{\beta^2}\Big(\mathcal{E}_\Phi(\mathsf{R}^o, \delta) - \rho^{\mathsf{cen}}\Big)^2\right\}. \tag{10}$$

As established in Appendix A.3.1, $\rho^{\mathsf{cen}} \le \rho$ for many commonly used classifiers. Since $N_{\max} < N_{\mathsf{tol}}$ and $\alpha \ge 1$, the centralized guarantee in (10) is strictly better than its distributed counterpart in (9). The gap between these bounds reflects the price of independent distributed training.

3.2 Benefits of collective classification

Let $\mathcal{M} \triangleq \{\gamma^* \widehat{\gamma} \leq 0\}$ denote the event of misclassification for the collective prediction $\widehat{\gamma}$ from (6), and let $\mathcal{C}_\delta$ denote the event of δ-margin consistent training in (7). The probability of error $P_e \triangleq \mathbb{P}[\mathcal{M}]$ during the inference phase can be upper bounded as

$$P_e = \mathbb{P}\big(\mathcal{M} \cap \mathcal{C}_\delta\big) + \mathbb{P}\big(\mathcal{M} \cap \overline{\mathcal{C}_\delta}\big) \leq \mathbb{P}\big(\mathcal{M}|\mathcal{C}_\delta\big) + \mathbb{P}\big(\overline{\mathcal{C}_\delta}\big). \tag{11}$$

Since $\mathbb{P}\big(\overline{\mathcal{C}_\delta}\big)$ can be bounded using Lemma 1, we establish the bound for $\mathbb{P}\big(\mathcal{M}|\mathcal{C}_\delta\big)$ as follows.

Theorem 1 (Classification error). *Suppose the same assumptions as those in Lemma 1. Assume that the agents follow the distributed learning rule* (3) *and label their testing samples $\boldsymbol{h}_k^*$ with $\widehat{\gamma}$ according to* (6)*. Then, we have*

$$\mathbb{P}\big(\mathcal{M}|\mathcal{C}_\delta\big) \leq \exp\left\{-\frac{\delta^2}{2\beta^2 \sum_{k=1}^K \pi_k^2}\right\}. \tag{12}$$

We now compare the proposed cooperative prediction with the non-cooperative case, where agents act independently during test time (i.e., $K = 1$ and π reduces to $\pi_1 = 1$). First, under event $\mathcal{C}_\delta$, the bound on $\mathbb{P}(\mathcal{M}|\mathcal{C}_\delta)$ is always tighter with collaboration since $\frac{1}{K} \leq \sum_{k=1}^K \pi_k^2 < 1$ for any A. This gain stems from the conditional independence of local testing samples $\boldsymbol{h}_k^*$: each sample provides independent evidence about the underlying label. In the homogeneous agents setting, the $\boldsymbol{h}_k^*$ variables are i.i.d. draws and sufficient communication enables each agent to aggregate information from K samples, as opposed to only one in the non-cooperative case. Moreover, the tightest bound in (12) is achieved when the Perron vector is uniform, i.e., $\pi_k = \frac{1}{K}$, which corresponds to a doubly-stochastic combination policy A. Importantly, the simple averaging rule for constructing A, where each agent assigns equal weights to all neighbors, does not ensure a uniform π for arbitrary topologies. Several useful rules for constructing doubly-stochastic A are available in the literature [21–23]. Second, cooperation also relaxes the conditions required in performance guarantees. Specifically, the event $\mathcal{C}_\delta$ may not hold locally when some agents' features are uninformative (e.g., $\boldsymbol{c}_k(h) = 0$ for all $h \in \mathcal{H}_k$). Yet at the network level, in view of (7) and (16), $\mathcal{C}_\delta$ is attainable as long as enough agents provide informative features. The feasibility condition in Assumption 4 also reflects this: while the task may be locally infeasible for some agents (i.e., $\mathsf{R}_k^o = \Phi(0)$), it remains feasible *globally* if at least one agent performs better.

3.3 Finite number of communication rounds

In this part, we consider that only t rounds of communication are allowed during test time. According to (3), the decision statistic for agent k at the t-th communication round is given by

$$\boldsymbol{\lambda}_{k,t} = \sum_{\ell=1}^K [A^t]_{\ell k} \boldsymbol{\lambda}_{\ell,0} = \sum_{\ell=1}^K [A^t]_{\ell k} \boldsymbol{c}_\ell(\boldsymbol{h}_\ell^*). \tag{13}$$

After communication, agent k makes its prediction as $\widehat{\gamma}_{k,t} = \mathrm{sgn}(\boldsymbol{\lambda}_{k,t})$. In the absence of consensus on $\boldsymbol{\lambda}_{k,t}$, the predictions will be agent-dependent. We denote by $\mathcal{M}_{k,t} \triangleq \{\gamma^* \widehat{\gamma}_{k,t} \leq 0\}$ the event of misclassification for prediction $\widehat{\gamma}_{k,t}$. The following bound on $\mathbb{P}(\mathcal{M}_{k,t}|\mathcal{C}_\delta)$ is then established.

Theorem 2 (Finite-time classification error). *Suppose the same assumptions as those in Lemma 1. Assume that the agents follow the distributed learning rule* (3) *within* t *rounds of communication, and label their testing samples $\boldsymbol{h}_k^*$ with $\widehat{\gamma}_{k,t}$. Then, we have*

$$\mathbb{P}\big(\mathcal{M}_{k,t}|\mathcal{C}_\delta\big) \leq \exp\left\{-\frac{\big(\delta - 2\beta K C(A,\sigma)\sigma^t\big)^2}{2\beta^2 \sum_{\ell=1}^K \big([A^t]_{\ell k}\big)^2}\right\}, \quad \forall t \geq \frac{\log \frac{\delta}{2\beta K C(A,\sigma)}}{\log \sigma} \tag{14}$$

where σ is a constant satisfying $\sigma_A < \sigma < 1$ and $C(A,\sigma)$ is a constant depending on A and σ.

The bound in (14) is both time- and agent-dependent. In addition to the Perron vector π, the second-largest magnitude σ_A among all eigenvalues of A plays a key role in (14) through the parameters σ and $C(A,\sigma)$. This aligns with its role in the DeGroot model, where σ_A characterizes the convergence rate of consensus [22].

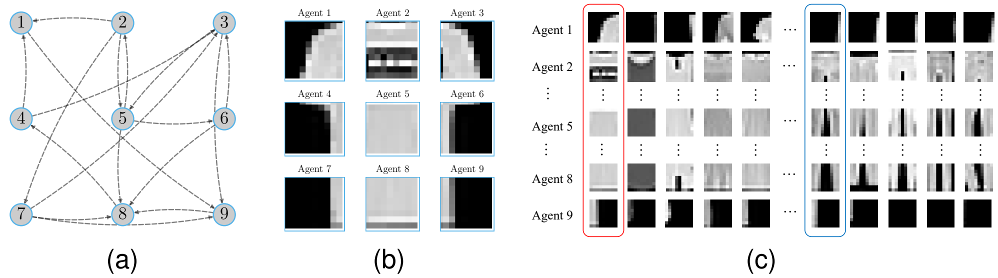

Figure 1: (a) Communication network. (b) Local observations of each agent. (c) Local datasets.

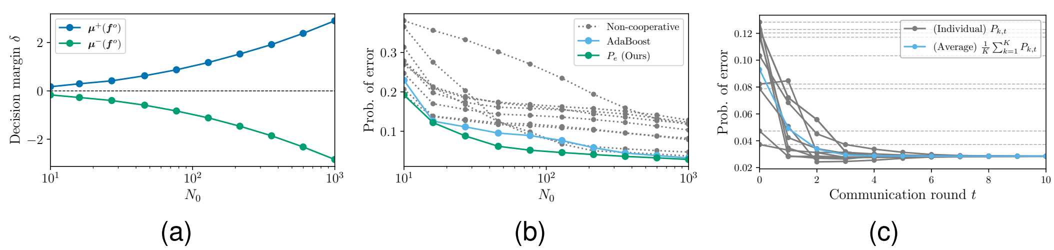

Figure 2: (a) Decision margins. (b) Classification errors. (c) Evolution of prediction errors.

4 Numerical simulations

We implement a binary classification task of distinguishing between T-shirts and trousers using the FashionMNIST dataset [24]. A network of 9 spatially distributed agents is considered, each observing a different portion of an image from the dataset. The communication network topology, the observation map of the agents, and an example of the local training datasets are shown in Fig. 1. For simplicity, we assume $N_k = N_0, \forall k \in \mathcal{K}$. In Fig. 2a, we present the two expected decision statistics, $\boldsymbol{\mu}^+(\boldsymbol{f}^o)$ and $\boldsymbol{\mu}^-(\boldsymbol{f}^o)$ defined in (16), achieved by the network under different N_0. It is obvious that the achieved decision margin δ in (7) increases with N_0, yielding a more informative network for the subsequent prediction. In Fig. 2b, we evaluate the classification error across varying N_0 under three scenarios: (i) non-cooperative learning, (ii) AdaBoost involving a cooperative sequential training, and (iii) the proposed collective classification framework. From Fig. 2b, collaboration improves classification performance, as the error is highest in the non-cooperative scenario. In Fig. 2c, we show the evolution of the error $P_{k,t}$ during the collaboration process when $N_0 = 1000$. Here, $P_{k,t}$ denotes the classification error associated with the prediction $\widehat{\gamma}_{k,t}$ of agent k after t rounds of communication. It is worth noting that $P_{k,0}$ at $t = 0$ corresponds to the classification error incurred by agent k in the non-cooperative scenario (shown as light gray dashed lines in Fig. 2c). The average of $P_{k,t}$ across all agents is also shown in Fig. 2c, which decreases over the communication rounds before converging. This highlights the performance improvement in the network achieved through collective prediction.

5 Concluding remarks

Given the growing use of AI in socio-technical systems, collaborative multi-agent decision-making without joint training will become increasingly important. In this work, we established theoretical bounds that characterize the cost of independent training and highlight the benefits of cooperation for individual agents in binary classification. Future work includes developing distributed decision rules beyond the simple DeGroot model, such as adaptive or strategically designed aggregation mechanisms, with particular emphasis on settings involving malicious or self-interested agents.

Acknowledgments

Mert Kayaalp's work was supported by UBS Switzerland AG and its affiliates through the UBS-IDSIA AI Lab.

References

[1] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[2] Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. *Multi-agent reinforcement learning: A selective overview of theories and algorithms*, pages 321–384. Springer, 2021.

[3] Celestine Mendler-Dünner, Wenshuo Guo, Stephen Bates, and Michael I. Jordan. Test-time collective prediction. In *Advances in Neural Information Processing Systems*, volume 34, pages 13719–13731, 2021.

[4] Selim F. Yilmaz, Burak Hasircioğlu, Li Qiao, and Deniz Gündüz. Private collaborative edge inference via over-the-air computation. *IEEE Transactions on Machine Learning in Communications and Networking*, 3:215–231, 2025.

[5] Prateek Yadav, Colin Raffel, Mohammed Muqeeth, Lucas Caccia, Haokun Liu, Tianlong Chen, Mohit Bansal, Leshem Choshen, and Alessandro Sordoni. A survey on model moerging: Recycling and routing among specialized experts for collaborative learning. *Transactions on Machine Learning Research*, 2025.

[6] Costas Mavromatis, Petros Karypis, and George Karypis. Pack of LLMs: Model fusion at test-time via perplexity optimization. In *Conference on Language Modeling*, 2024.

[7] Morris H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69 (345):118–121, 1974.

[8] Mancur Olson Jr. *The Logic of Collective Action: Public Goods and the Theory of Groups*, volume 124. Harvard University Press, 1971.

[9] Moritz Hardt, Eric Mazumdar, Celestine Mendler-Dünner, and Tijana Zrnic. Algorithmic collective action in machine learning. In *International Conference on Machine Learning*, pages 12570–12586. PMLR, 2023.

[10] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[11] Jakub Konečnỳ, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv:1610.05492*, 2016.

[12] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[13] Zhanhong Jiang, Aditya Balu, Chinmay Hegde, and Soumik Sarkar. Collaborative deep learning in fixed topology networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[14] Omar A Alzubi, Jafar Ahmad Abed Alzubi, Sara Tedmori, Hasan Rashaideh, and Omar Almomani. Consensus-based combining method for classifier ensembles. *The International Arab Journal of Information Technology*, 15(1):76–86, 2018.

[15] Robi Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, 2006.

[16] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons, 2014.

[17] Peter L. Bartlett, Yoav Freund, Wee Sun Lee, and Robert E. Schapire. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5): 1651–1686, 1998.

[18] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC, 1st edition, 2012.

[19] Virginia Bordignon, Stefan Vlaski, Vincenzo Matta, and Ali H. Sayed. Learning from heterogeneous data based on social interactions over graphs. *IEEE Transactions on Information Theory*, 69(5):3347–3371, 2023.

[20] Ping Hu, Virginia Bordignon, Mert Kayaalp, and Ali H. Sayed. Non-asymptotic performance of social machine learning under limited data. *Signal Processing*, 230:109849, 2025.

[21] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[22] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

[23] Ali H. Sayed. Adaptation, learning, and optimization over networks. *Foundations and Trends® in Machine Learning*, 7(4-5):311–801, 2014.

[24] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017.

[25] Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[26] Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[27] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT press, 2nd edition, 2018.

[28] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Proceedings of The 28th Conference on Learning Theory*, volume 40, pages 1376–1401. PMLR, 2015.

[29] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012.

[30] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

A Technical Appendices and Supplementary Material

A.1 Definitions

First, we provide the definitions of $\boldsymbol{\mu}^+(f)$ and $\boldsymbol{\mu}^-(f)$ that determine the δ-margin consistent training condition in (7).

Definitions of $\boldsymbol{\mu}^+(f)$ and $\boldsymbol{\mu}^-(f)$ For each agent k and each function $f_k \in \mathcal{F}_k$, we denote by $\boldsymbol{\mu}_k^+(f_k)$ and $\boldsymbol{\mu}_k^-(f_k)$ the *expected* decision statistics generated by classifier $\boldsymbol{c}_k$, which is constructed from applying the debiasing operation associated with the training set $\mathcal{D}_k$ to function f_k—see (2), under the two classes:

$$\boldsymbol{\mu}_k^+(f_k) \triangleq \mathbb{E}_{\boldsymbol{h}_k \sim p_k(\cdot|+1)} \boldsymbol{c}_k(\boldsymbol{h}_k), \quad \boldsymbol{\mu}_k^-(f_k) \triangleq \mathbb{E}_{\boldsymbol{h}_k \sim p_k(\cdot|-1)} \boldsymbol{c}_k(\boldsymbol{h}_k). \tag{15}$$

The expected decision statistics for the *network* are defined as the following weighted averages:

$$\boldsymbol{\mu}^+(f) = \sum_{k=1}^{K} \pi_k \boldsymbol{\mu}_k^+(f_k), \quad \boldsymbol{\mu}^-(f) = \sum_{k=1}^{K} \pi_k \boldsymbol{\mu}_k^-(f_k) \tag{16}$$

where the argument f refers to the dependence of the networked quantities on the set of functions $\{f_k\}$. The inclusion of the Perron vector π in (16) is consistent with the collective prediction nature of the testing phase, where all agents reach a consensus on prediction through $\boldsymbol{\lambda}_{\mathsf{ave}}$ in (5).

Referring to the definition provided in (16), condition (7) requires that the expected decision statistic $\boldsymbol{\mu}^+(f)$ (or $\boldsymbol{\mu}^-(f)$) generated by the after-training classifier network f is positive (or negative) when the testing feature vector $\boldsymbol{h}^*$ is drawn from class $+1$ (or -1), and that its distance from the decision boundary at 0 exceeds δ. Importantly, the decision boundary at 0 corresponds to a prediction based on the sign of $\boldsymbol{\lambda}_{\mathsf{ave}}$.

In the following, we provide the definition for some variables that appear in Lemma 1.

Rademacher complexity First, we introduce the notation for the Rademacher complexity of a general function class $\mathcal{G}$ [25]. Let $S \triangleq \{z_1, z_2, \ldots, z_m\}$ be a fixed sample set of size m. The *empirical* Rademacher complexity of $\mathcal{G}$ with respect to S is defined as

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \mathbb{E}_{\boldsymbol{r}} \sup_{g \in \mathcal{G}} \left| \frac{1}{m} \sum_{n=1}^{m} \boldsymbol{r}_n g(z_n) \right| \tag{17}$$

where $\boldsymbol{r} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_m)$, with $\boldsymbol{r}_n$s independent uniform random variables taking values in Γ. The Rademacher complexity of $\mathcal{G}$, denoted by $\mathfrak{R}_m(\mathcal{G})$, is the expectation of the empirical Rademacher complexity over all sample sets S of size m:

$$\mathfrak{R}_m(\mathcal{G}) \triangleq \mathbb{E}_S \widehat{\mathfrak{R}}_S(\mathcal{G}). \tag{18}$$

Based on these notation, we define the individual Rademacher complexity of the function class $\mathcal{F}_k$ under the training set size N_k at each agent k as

$$\rho_k \triangleq \mathfrak{R}_{N_k}(\mathcal{F}_k). \tag{19}$$

The network Rademacher complexity for the classifier network under the given training setup in Section 2 is defined as the weighted average of the individual Rademacher complexities:

$$\rho \triangleq \sum_{k=1}^{K} \pi_k \rho_k. \tag{20}$$

Function $\mathcal{E}_\Phi(\mathsf{R}^o, \delta)$ From Eqs. (A.16), (A.18), and (A.20) in [20], for any $0 \le \delta < \delta_{\max}$, the expression for $\mathcal{E}_\Phi(\mathsf{R}^o, \delta)$ is given by

$$\mathcal{E}_\Phi(\mathsf{R}^o, \delta) \triangleq \frac{d_\delta^\star - \delta}{4} = \frac{\Phi(d_\delta^\star) - \mathsf{R}^o}{8 L_\Phi} \tag{21}$$

where $d_\delta^\star$ is the solution to the equation $d - \delta - \frac{\Phi(d) - \mathsf{R}^o}{2L_\Phi} = 0$. An important property of d_δ is that it increases with δ. Under Assumption 1 in Appendix A.2, the function $\mathcal{E}_\Phi(\mathsf{R}^o, \delta)$ decreases as δ grows. Together with the maximum feasible δ, i.e., $\delta_{\max}$, this function characterizes the inherent difficulty of the classification task.

Parameter $\delta_{\max}$ According to Eq. (A.21) from [20], $\delta_{\max}$ is defined as the largest δ for which there exists a solution $d_\delta^\star$ satisfying $d_\delta^\star < d_{\mathsf{R}}$, where $d_{\mathsf{R}} \triangleq \inf\{x : \Phi(x) = \mathsf{R}^o\}$. Since Φ is non-increasing under Assumption 1, d_{R} is non-increasing in the optimal risk R^o. Hence, $\delta_{\max}$ tends to increase as R^o decreases. The feasibility condition in Assumption 4 (see Appendix A.2) is required to guarantee the existence of $\delta_{\max} > 0$.

A.2 Assumptions

In this work, we impose the following assumptions on the training and testing phases of our framework. These assumptions are widely adopted in the literature of supervised classification and distributed learning over graphs [23, 26, 27].

Assumption 1 (Conditions on the loss function). *The loss function Φ is convex, non-increasing, and differentiable at* 0 *with $\Phi'(0) < 0$. Also, it is L_Φ-Lipschitz.*

Assumption 2 (Boundedness of functions). *There exists a constant $\beta > 0$ such that for each agent $k \in \mathcal{K}$ and each function $f_k \in \mathcal{F}_k$, $|f_k(h)| \le \beta$ for all $h \in \mathcal{H}_k$.*

Assumption 3 (Strongly-connected graph). *The graph of the communication network is strongly connected. That is, there exist paths with positive combination weights between any two distinct agents in both directions (these trajectories need not be the same), and at least one agent has a self-loop, i.e., $a_{mm} > 0$ for some agent m. Then, the combination matrix $A = [a_{\ell k}]$ is primitive and admits a Perron vector π satisfying:*

$$A\pi = \pi, \quad \sum_{k=1}^{K} \pi_k = 1, \ \textit{and}\ \pi_k > 0, \quad \forall k \in \mathcal{K}. \tag{22}$$

Furthermore, the second-largest magnitude among all eigenvalues of A, denoted by σ_A, is strictly smaller than 1.

Assumption 4 (Feasibility). *The target risk R^o for the network satisfies $\mathsf{R}^o < \Phi(0)$, with R^o defined by*

$$\mathsf{R}^o \triangleq \sum_{k=1}^{K} \pi_k \mathsf{R}_k^o, \quad \textit{where } \mathsf{R}_k^o \triangleq \inf_{f_k \in \mathcal{F}_k} \mathbb{E}_{(\boldsymbol{h}_{k,n}, \boldsymbol{\gamma}_{k,n})} \Phi(\boldsymbol{\gamma}_{k,n} f_k(\boldsymbol{h}_{k,n})). \tag{23}$$

Assumption 1 guarantees that the loss function Φ is classification-calibrated [26, 27]. Assumption 2 requires the classifier outputs to be bounded to ensure a well-posed analysis. Such an assumption is standard in statistical learning theory, as it facilitates the use of fundamental mathematical tools, in particular concentration inequalities [27]. It is also consistent with practical models, where outputs are typically probabilities (bounded in $[0, 1]$) or decision statistics that can be normalized without loss of generality. Assumption 3 ensures that under the DeGroot model, information originating from any agent can eventually diffuse throughout the entire network [23]. Assumption 4 requires the existence of at least one agent whose feature vectors are informative for classification. If, for every agent k, the optimal risk satisfies $\mathsf{R}_k^o = \Phi(0)$, then by Assumption 1, the optimal classifier reduces to the trivial rule $f_k = 0$, which assigns labels $+1$ and -1 with equal probability. In this case, the agents collectively observe no informative features, and cooperation offers no benefit.

A.3 Proofs

A.3.1 Proof of Lemma 1

The bound in (9) for the distributed independent training was established in [20]. In the homogeneous agents setting, the only difference between distributed and centralized training lies in the number of training samples per classifier, namely N_k. To derive the bound in (10) for centralized training, we therefore examine how N_k influences the parameters in (9). Let $\mathcal{H}$ and $\mathcal{F}$ denote the common feature space and function class shared by all agents, respectively.

(i) Network complexity ρ: In the homogeneous agents setting, $\rho_k = \mathfrak{R}_{N_k}(\mathcal{F})$ for each $k \in \mathcal{K}$. Since centralized training uses all N_{tol} training samples, we have $\rho^{\mathsf{cen}} = \mathfrak{R}_{N_{\mathsf{tol}}}(\mathcal{F})$. For many standard base models, such as feedforward neural networks with norm-constrained weights and kernel machines in bounded RKHS balls, the Rademacher complexity decreases on the order of $1/\sqrt{m}$ with the training set size m [19, 25, 28]. Since $N_k < N_{\mathsf{tol}}$, it follow that $\rho^{\mathsf{cen}} \leq \rho_k$ for all agents k. By definition (20), $\rho = \sum_{k=1}^{K} \pi_k \rho_k$ under distributed training, which implies $\rho^{\mathsf{cen}} \leq \rho$.

(ii) Function $\mathcal{E}_\Phi(\mathsf{R}^o, \delta)$: From its definition in (21), $\mathcal{E}_\Phi(\mathsf{R}^o, \delta)$ depends on the network target risk R^o, the loss function Φ, and the decision margin δ. Since δ is a free parameter in (9) and Φ is fixed across training setups, the only potential difference in $\mathcal{E}_\Phi(\mathsf{R}^o, \delta)$ between centralized and distributed training can arise through R^o. With $\mathcal{H}_k = \mathcal{H}$, the individual target risks R_k^o are identical across agents, so their weighted average R^o is the same in both settings. Therefore, $\mathcal{E}_\Phi(\mathsf{R}^o, \delta)$ is identical for the two training setups.

(iii) Parameter $\delta_{\max}$: As elaborated in Appendix A.1, $\delta_{\max}$ is obtained by solving an equation involving R^o and Φ. Since neither R^o nor Φ depends on N_k, $\delta_{\max}$ is the same in both training setups.

(iv) Parameters α and $N_{\max}$: Since centralized training corresponds to $K = 1$, we have $\alpha = 1$ and $N_{\max} = N_{\mathsf{tol}}$ in this case.

With these parameter dependencies established, we obtain the bound in (10) for centralized training.

A.3.2 Proof of Theorem 1

To simplify notation, we define a centralized classifier $\boldsymbol{c}_{\mathsf{ave}}$ corresponding to the consensus decision statistic $\boldsymbol{\lambda}_{\mathsf{ave}}$ as follows. In view of (5), after sufficient rounds of communication, the distributed averaging rule (3) functions as a *centralized* classifier $\boldsymbol{c}_{\mathsf{ave}}$ denoted by

$$\boldsymbol{c}_{\mathsf{ave}}(\boldsymbol{h}^*) \triangleq \sum_{k=1}^{K} \pi_k \boldsymbol{c}_k(\boldsymbol{h}_k^*) \tag{24}$$

where $\boldsymbol{h}^* \triangleq (\boldsymbol{h}_1^*, \boldsymbol{h}_2^*, \ldots, \boldsymbol{h}_K^*)$ denotes the collection of local testing samples received by all agents.

Since the training phase is independent of the prediction phase, the randomness introduced during training can be “frozen" when analyzing the prediction phase. To enhance clarity, we will use normal font for random variables that are independent of the prediction phase throughout our proof. For example, the classifiers $\boldsymbol{f}_k^o$, $\boldsymbol{c}_k^o$, and $\boldsymbol{c}_{\mathsf{ave}}$ will be denoted by f_k^o, c_k^o, and c_{ave}, respectively. The proof is based on applying McDiarmid's inequality to the converged decision statistic $\boldsymbol{\lambda}_{\mathsf{ave}}$ in (5) (or equivalently, $c_{\mathsf{ave}}(\boldsymbol{h}^*)$ in (24)) under the distributed learning rule (3). Suppose now that the true label of $\boldsymbol{h}_k^*$ is $+1$, i.e., $\boldsymbol{\gamma}^* = +1$. We begin by deriving the necessary conditions for using McDiarmid's inequality. According to definition (24), the expectation of $c_{\mathsf{ave}}(\boldsymbol{h}^*)$ is

$$\mathbb{E}_{\boldsymbol{h}^*}^{(+1)} c_{\mathsf{ave}}(\boldsymbol{h}^*) = \mathbb{E}_{\boldsymbol{h}^*}^{(+1)} \left[\sum_{k=1}^{K} \pi_k c_k(\boldsymbol{h}_k^*) \right] = \sum_{k=1}^{K} \pi_k \mathbb{E}_{\boldsymbol{h}_k^*}^{(+1)} c_k(\boldsymbol{h}_k^*) = \mu^+(f^o), \tag{25}$$

which is greater than δ when the δ-margin consistent training condition (7) is satisfied during the training phase. Here, $\mathbb{E}_{\boldsymbol{x}}^{(+1)}$ represents the expectation operator conditioned on class $+1$, i.e., we assume $\boldsymbol{h}_k^* \sim p_k(\cdot|+1)$ for each k. Given the conditional independence of testing samples across agents, it is clear that $\boldsymbol{h}^*$ consists of K independent random variables. Consider another collection of observations $\widehat{\boldsymbol{h}}$ that differs from $\boldsymbol{h}^*$ only in the k-th sample $\boldsymbol{h}_k^*$, i.e., $\widehat{\boldsymbol{h}}_\ell = \boldsymbol{h}_\ell^*$ for all $\ell \neq k$. The difference between $c_{\mathsf{ave}}(\boldsymbol{h}^*)$ and $c_{\mathsf{ave}}(\widehat{\boldsymbol{h}})$ is bounded as

$$\left| c_{\mathsf{ave}}(\boldsymbol{h}^*) - c_{\mathsf{ave}}(\widehat{\boldsymbol{h}}) \right| = \left| \pi_k c_k(\boldsymbol{h}_k^*) - \pi_k c_k(\widehat{\boldsymbol{h}}_k) \right| = \pi_k \left| f_k^o(\boldsymbol{h}_k^*) - f_k^o(\widehat{\boldsymbol{h}}_k) \right| \leq 2\beta\pi_k \tag{26}$$

where we use the definition of c_k in (2) and the bound on all functions f_k in Assumption 2. Therefore, the decision statistic $c_{\mathsf{ave}}(\boldsymbol{h}^*)$, as a function of K independent random variables $\boldsymbol{h}_k^*$ (one per agent), has a bounded difference $2\beta\pi_k$ w.r.t. changes in the k-th coordinate. Conditioning on the δ-margin consistent training, we have

$$\begin{aligned}
\mathbb{P}\Big(c_{\mathsf{ave}}(\boldsymbol{h}^*) \leq 0 | \boldsymbol{\gamma}^* = +1, \mathcal{C}_\delta\Big) &= \mathbb{P}\Big(c_{\mathsf{ave}}(\boldsymbol{h}^*) - \mathbb{E}_{\boldsymbol{h}^*}^{(+1)} c_{\mathsf{ave}}(\boldsymbol{h}^*) \leq -\mathbb{E}_{\boldsymbol{h}^*}^{(+1)} c_{\mathsf{ave}}(\boldsymbol{h}^*) | \mathcal{C}_\delta\Big) \\
&\overset{\text{(a)}}{=} \mathbb{P}\Big(c_{\mathsf{ave}}(\boldsymbol{h}^*) - \mathbb{E}_{\boldsymbol{h}^*}^{(+1)} c_{\mathsf{ave}}(\boldsymbol{h}^*) \leq -\mu^+(f^o) | \mathcal{C}_\delta\Big) \\
&\overset{\text{(b)}}{\leq} \exp\left\{ -\frac{\big(\mu^+(f^o)\big)^2}{2\beta^2 \sum_{k=1}^{K} \pi_k^2} \right\} \overset{\text{(c)}}{\leq} \exp\left\{ -\frac{\delta^2}{2\beta^2 \sum_{k=1}^{K} \pi_k^2} \right\}
\end{aligned} \tag{27}$$

where in (a) we used the expression for the expectation in (25). In (b), we applied McDiarmid's inequality to $c_{\mathsf{ave}}(\boldsymbol{h}^*)$ based on the following two facts: i) $\mu^+(f^o) > \delta > 0$ conditioned on the event $\mathcal{C}_\delta$ given by (7), and ii) the bounded difference condition is satisfied in view of (26). The last inequality (c) follows from the definition of $\mathcal{C}_\delta$ in (7).

Using similar techniques to those in (25)–(27), we can establish the following upper bound for the case $\boldsymbol{\gamma}^* = -1$:

$$\mathbb{P}\big(c_{\mathsf{ave}}(\boldsymbol{h}^*) \geq 0 | \boldsymbol{\gamma}^* = -1, \mathcal{C}_\delta\big) \leq \exp\left\{-\frac{\delta^2}{2\beta^2 \sum_{k=1}^K \pi_k^2}\right\}. \tag{28}$$

Therefore, combining (27) and (28), the bound for $\mathbb{P}\big(\mathcal{M}|\mathcal{C}_\delta\big)$ in Theorem 1 is established from

$$\begin{aligned}&\mathbb{P}\big(\mathcal{M}|\mathcal{C}_\delta\big) = \mathbb{P}\big(\boldsymbol{\gamma}^* c_{\mathsf{ave}}(\boldsymbol{h}^*) \leq 0 | \mathcal{C}_\delta\big) \\ &= \mathbb{P}(\boldsymbol{\gamma}^* = +1)\mathbb{P}\big(c_{\mathsf{ave}}(\boldsymbol{h}^*) \leq 0 | \boldsymbol{\gamma}^* = +1, \mathcal{C}_\delta\big) + \mathbb{P}(\boldsymbol{\gamma}^* = -1)\mathbb{P}\big(c_{\mathsf{ave}}(\boldsymbol{h}^*) \geq 0 | \boldsymbol{\gamma}^* = -1, \mathcal{C}_\delta\big).\end{aligned} \tag{29}$$

From (11), a full characterization of the classification error P_e is obtained by combining the bound (12) in Theorem 1 with the upper bound for $\mathbb{P}(\overline{\mathcal{C}_\delta})$ established in Lemma 1:

$$P_e \leq 2\exp\left\{-\frac{8N_{\max}}{\alpha^2\beta^2}\Big(\mathcal{E}(\mathsf{R}^o, \delta) - \rho\Big)^2\right\} + \exp\left\{-\frac{\delta^2}{2\beta^2 \sum_{k=1}^K \pi_k^2}\right\}. \tag{30}$$

A.3.3 Proof of Theorem 2

Similar to Appendix A.3.2, we use normal font for random variables that are independent of the training phase. The proof proceeds analogously to that of Theorem 1, using McDiarmid's inequality applied to the decision statistic $\boldsymbol{\lambda}_{k,t}$. For this purpose, we introduce the following convergence results of matrix powers for the left-stochastic combination policy A established in [29].

Lemma A (**Powers of left-stochastic matrices**). *Let σ_A denote the largest magnitude among all eigenvalues of the left-stochastic matrix A other than 1. For any positive σ satisfying $\sigma_A < \sigma < 1$, there exists a positive constant $C(A, \sigma)$ depending on A and σ such that*

$$\Big|[A^t]_{\ell k} - \pi_\ell\Big| \leq C(A, \sigma)\sigma^t \tag{31}$$

for all ℓ, k and all $t \in \mathbb{N}$.

We begin with the case of $\boldsymbol{\gamma}^* = +1$, and use the notation $\boldsymbol{\lambda}_{k,t}(\boldsymbol{h})$ to emphasize the dependence of $\boldsymbol{\lambda}_{k,t}$ on a testing sample $\boldsymbol{h}$. The probability of interest in this case is the following:

$$\begin{aligned}\mathbb{P}\big(\mathcal{M}_{k,t}|\boldsymbol{\gamma}^* = +1, \mathcal{C}_\delta\big) &\triangleq \mathbb{P}\Big(\boldsymbol{\lambda}_{k,t}(\boldsymbol{h}^*) \leq 0 | \boldsymbol{\gamma}^* = +1, \mathcal{C}_\delta\Big) \\ &= \mathbb{P}\Big(\boldsymbol{\lambda}_{k,t}(\boldsymbol{h}^*) - \mathbb{E}^{(+1)}_{\boldsymbol{h}^*}\boldsymbol{\lambda}_{k,t}(\boldsymbol{h}^*) \leq -\mathbb{E}^{(+1)}_{\boldsymbol{h}^*}\boldsymbol{\lambda}_{k,t}(\boldsymbol{h}^*) | \mathcal{C}_\delta\Big).\end{aligned} \tag{32}$$

In order to use McDiarmid's inequality, we examine the bounded difference property associated with $\boldsymbol{\lambda}_{k,t}(\boldsymbol{h}^*)$ by introducing another observation collection $\widehat{\boldsymbol{h}}$ that differs from $\boldsymbol{h}^*$ only in $\boldsymbol{h}^*_\ell$. It follows from (13) that

$$\Big|\boldsymbol{\lambda}_{k,t}(\boldsymbol{h}^*) - \boldsymbol{\lambda}_{k,t}(\widehat{\boldsymbol{h}})\Big| = \Big|[A^t]_{\ell k}\big(c_\ell(\boldsymbol{h}^*_\ell) - c_\ell(\widehat{\boldsymbol{h}}_\ell)\big)\Big| \leq 2\beta[A^t]_{\ell k}. \tag{33}$$

Next, we establish a lower bound for the expected value of $\boldsymbol{\lambda}_{k,t}(\boldsymbol{h}^*)$. Using the convergence of A^t in Lemma A, there exist two constants σ and $C(A, \sigma)$ with $\sigma_A < \sigma < 1$, such that the following holds:

$$\begin{aligned}\Big|\mathbb{E}^{(+1)}_{\boldsymbol{h}^*}\boldsymbol{\lambda}_{k,t}(\boldsymbol{h}^*) - \mu^+(f^o)\Big| &= \left|\sum_{\ell=1}^K \big([A^t]_{\ell k} - \pi_\ell\big)\mathbb{E}^{(+1)}_{\boldsymbol{h}^*} c_\ell(\boldsymbol{h}^*_\ell)\right| = \left|\sum_{\ell=1}^K \big([A^t]_{\ell k} - \pi_\ell\big)\mu^+_\ell(f^o_\ell)\right| \\ &\overset{\text{(a)}}{\leq} 2\beta\sum_{\ell=1}^K \Big|[A^t]_{\ell k} - \pi_\ell\Big| \overset{\text{(b)}}{\leq} 2\beta K C(A, \sigma)\sigma^t\end{aligned} \tag{34}$$

where (a) comes from the bound of f_ℓ and the triangle inequality for the absolute values, and (b) follows from (31). This yields

$$\mathbb{E}_{\boldsymbol{h}^*}^{(+1)} \boldsymbol{\lambda}_{k,t}(\boldsymbol{h}^*) \geq \mu^+(f^o) - 2\beta K C(A,\sigma)\sigma^t > \delta - 2\beta K C(A,\sigma)\sigma^t > 0 \tag{35}$$

for all

$$t \geq \frac{\log \frac{\delta}{2\beta K C(A,\sigma)}}{\log \sigma} \tag{36}$$

when the δ-margin consistent training condition (7) holds. Accordingly, the probability in (32) can be bounded as

$$\begin{aligned} \mathbb{P}\big(\mathcal{M}_{k,t}|\boldsymbol{\gamma}^* = +1, \mathcal{C}_\delta\big) &\leq \mathbb{P}\Big(\boldsymbol{\lambda}_{k,t}(\boldsymbol{h}^*) - \mathbb{E}_{\boldsymbol{h}^*}^{(+1)}\boldsymbol{\lambda}_{k,t}(\boldsymbol{h}^*) \leq -\big(\delta - 2\beta K C(A,\sigma)\sigma^t\big)\Big) \\ &\leq \exp\left\{-\frac{\big(\delta - 2\beta K C(A,\sigma)\sigma^t\big)^2}{2\beta^2 \sum_{\ell=1}^K \big([A^t]_{\ell k}\big)^2}\right\} \end{aligned} \tag{37}$$

by using McDiarmid's inequality with the bounded difference condition established in (33). With similar techniques to those used in (32)–(37) for the case $\boldsymbol{\gamma}^* = -1$, we obtain

$$\mathbb{P}\big(\mathcal{M}_{k,t}|\boldsymbol{\gamma}^* = -1, \mathcal{C}_\delta\big) \leq \exp\left\{-\frac{\big(\delta - 2\beta K C(A,\sigma)\sigma^t\big)^2}{2\beta^2 \sum_{\ell=1}^K \big([A^t]_{\ell k}\big)^2}\right\} \tag{38}$$

for all t satisfying (36). Together with (37), (38) proves the bound in (14).

Similar to (30), a full characterization of the classification error within t rounds of communication is obtained by combining (14) in Theorem 2 with the upper bound for $\mathbb{P}(\overline{\mathcal{C}_\delta})$ established in Lemma 1:

$$P_{k,t} \triangleq \mathbb{P}(\mathcal{M}_{k,t}) \leq 2\exp\left\{-\frac{8N_{\max}}{\alpha^2\beta^2}\Big(\mathcal{E}(\mathsf{R}^o,\delta) - \rho\Big)^2\right\} + \exp\left\{-\frac{\big(\delta - 2\beta K C(A,\sigma)\sigma^t\big)^2}{2\beta^2 \sum_{\ell=1}^K \big([A^t]_{\ell k}\big)^2}\right\}. \tag{39}$$

A.4 Additional experimental details and evaluations

Experimental setup on the FashionMNIST dataset. Given the training set size N_0, we randomly sample $\frac{N_0}{2}$ images for each category (T-shirts or trousers) from this dataset. Each image is then divided into 9 patches, which are assigned to the 9 agents. For every agent k, the local training set consists of the patches taken from the same location across all images. As shown in Fig. 1c, the patches in the k-th row constitute the training set for agent k. Moreover, the 9 patches in the first column (marked by the red box) collectively form the first T-shirt image in FashionMNIST, shown in Fig. 1b. Likewise, the 9 patches contained in the blue box reconstruct the first trouser image in FashionMNIST. In the training phase of our framework, each agent trains a feedforward neural network with one hidden layer of 15 neurons and a $\tanh$ activation function. Training is run using mini-batches of 10 samples over 30 epochs, with the logistic loss function and the Adam optimizer (learning rate $= 0.0001$). During the testing phase, the simple averaging rule is used for constructing the combination policy A. That is,

$$a_{\ell k} = \frac{1}{|\mathcal{N}_k|},\ \forall \ell \in \mathcal{N}_k, \quad \text{where } |\mathcal{N}_k| \text{ denotes the cardinality of } \mathcal{N}_k. \tag{40}$$

Each agent is assumed to have a self-loop in the communication network, which is omitted from Fig. 1a for clarity. For the AdaBoost strategy studied in Fig. 2b, the 9 local classifiers are trained sequentially, and the final predictions are made by combining the hard decisions of each classifier. All results presented in Fig. 2 for a fixed training set size N_0 are averaged over 200 randomly generated training sets of size N_0. In Fig. 2b, the proposed collaborative classification framework outperforms AdaBoost when N_0 is small. This can be partially attributed to the heterogeneity of feature spaces across agents, as illustrated in Fig. 1b, which limits the benefits of sequential cooperative training.

Additional experiments on the CIRAR10 dataset. We also conducted experiments on the CIFAR10 dataset [30], where we constructed a binary classification task of distinguishing cats from dogs. For this task, each agent employs a convolutional neural network composed of two convolutional layers followed by three fully-connected layers [20]. The construction of local feature vectors and

Figure 3: (a) An example of local training datasets. (b) The cat image with observation patches from the red box of (a). (c) The dog image with observation patches from the blue box of (a).

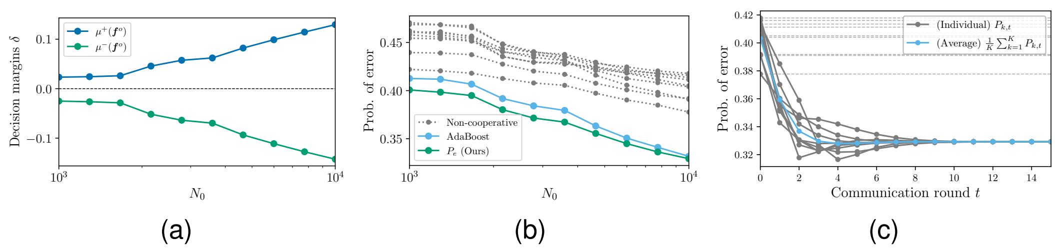

Figure 4: (a) Decision margins achieved under different training set sizes N_0. (b) Probabilities of error for non-cooperative agents (dotted lines), the proposed collaborative classification framework (green line), and the AdaBoost strategy (blue line) under different N_0. (c) Evolution of prediction errors versus the communication round for $N_0 = 10000$. All results correspond to the CIFAR10 dataset.

training datasets follows the same procedure as in the FashionMNIST case. An example of the local training datasets is presented in Fig. 3. Given the local dataset at each agent, the training is carried out with mini-batches of 128 samples over 100 epochs. The resulting decision margins and classification errors for different training set sizes N_0, along with the evolution of the error across communication rounds, are presented in Fig. 4. Similar behaviors to those observed in Fig. 2 for the FashionMNIST dataset are demonstrated in Fig. 4. In particular, Figs. 4b and 4c clearly illustrate the performance gains achieved through test-time collective classification.