Redundancy in cost functions for Byzantine fault-tolerant federated learning

From distributed optimization to federated learning

Shuo Liu [1]   Nirupam Gupta [2]   Nitin Vaidya [1]

[1] Georgetown University

[2] EPFL

ResilientFL 2021

Redundancy in cost functions for Byzantine fault-tolerant federated learning

From distributed optimization to federated learning

Shuo Liu [1]   Nirupam Gupta [2]   Nitin Vaidya [1]

[1] Georgetown University

[2] EPFL

ResilientFL 2021

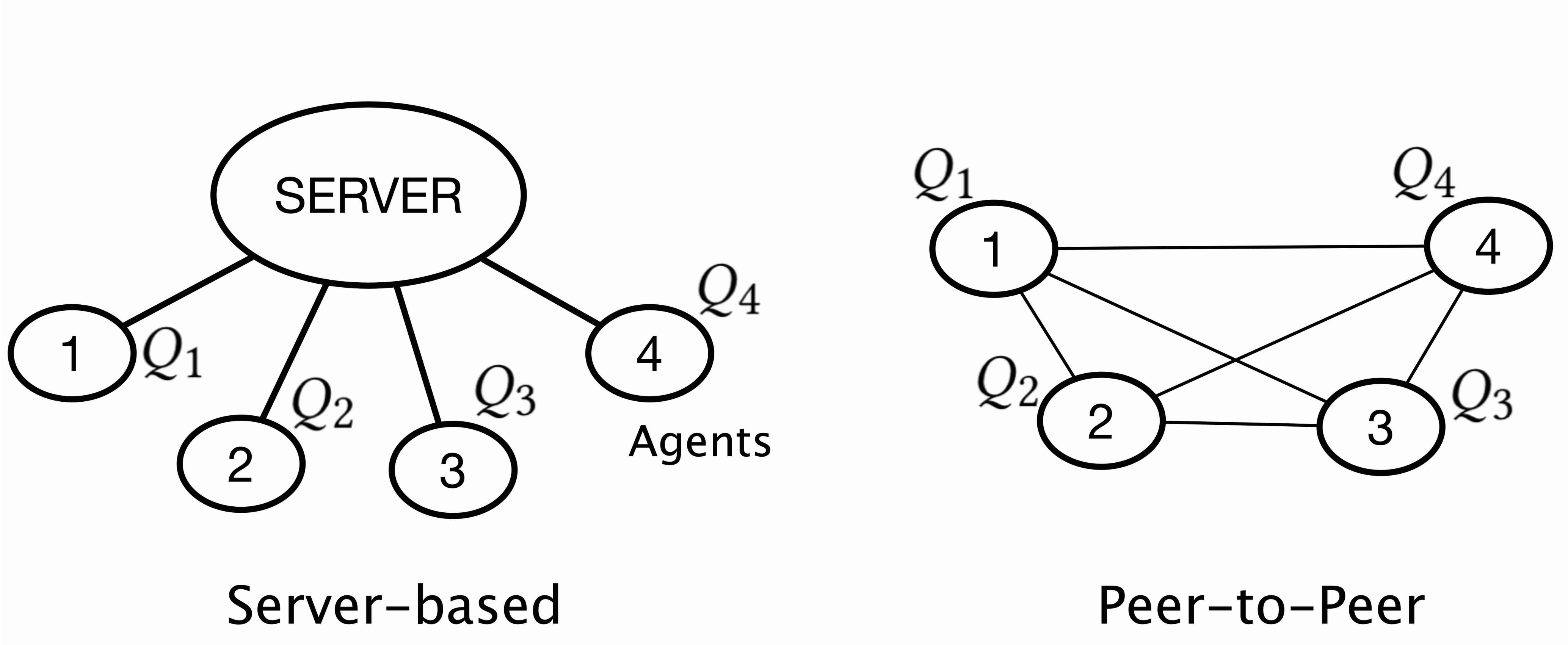
Introduction

Introduction

Fault-tolerance in distributed optimization

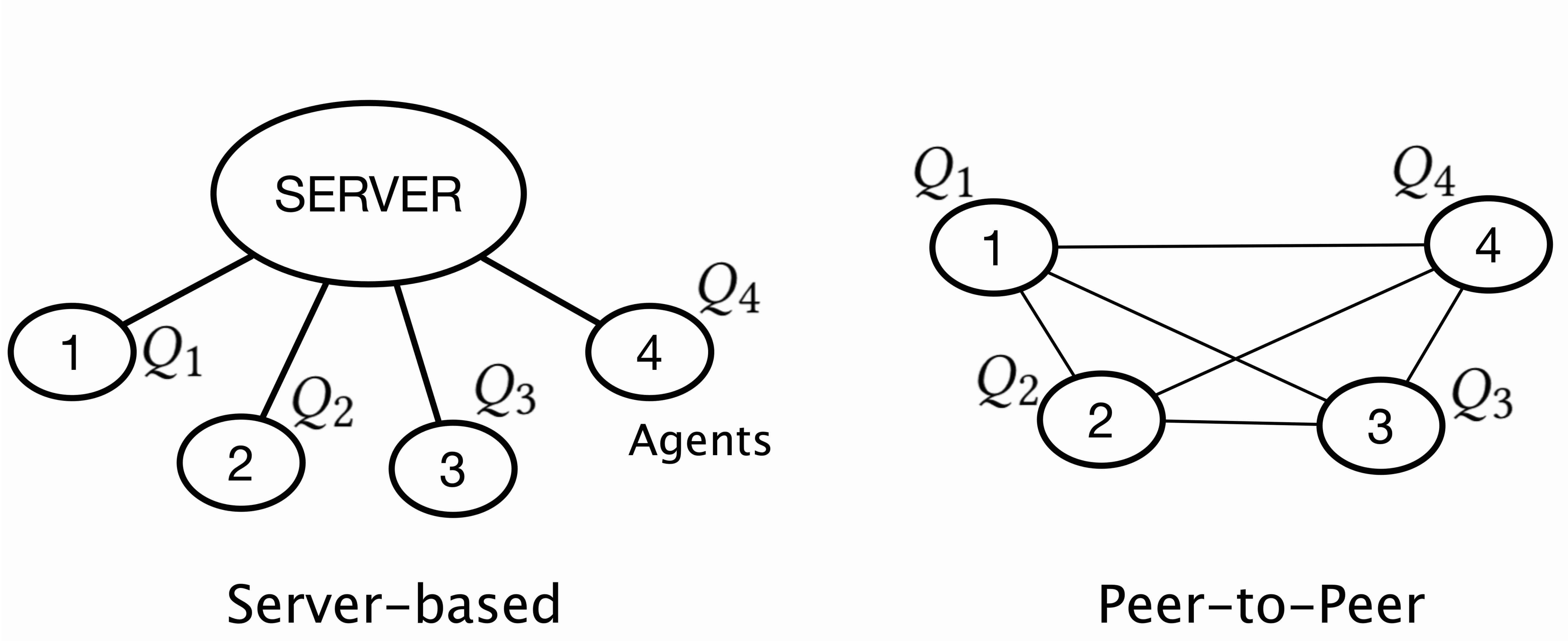
Introduction

Fault-tolerance in distributed optimization



Introduction

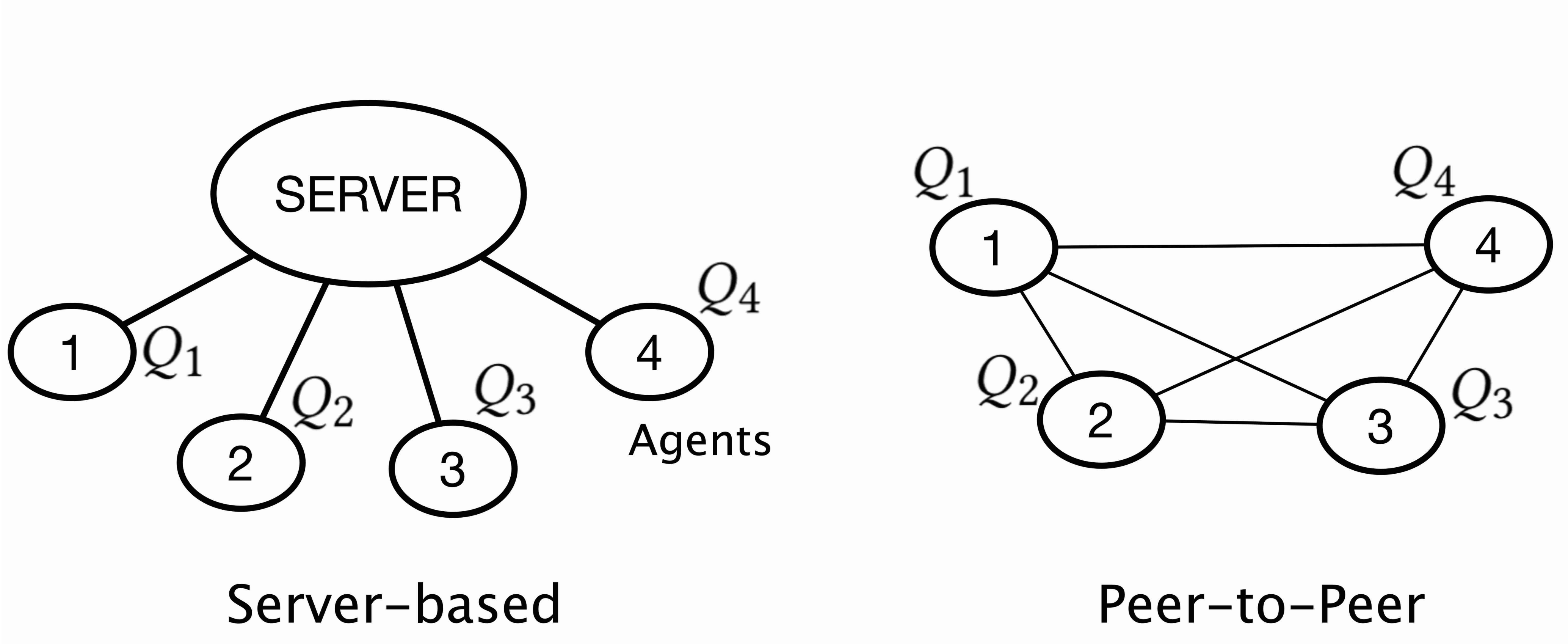
Fault-tolerance in distributed optimization



$$Q_i : \mathbb{R}^d \to \mathbb{R}$$

Introduction

Fault-tolerance in distributed optimization

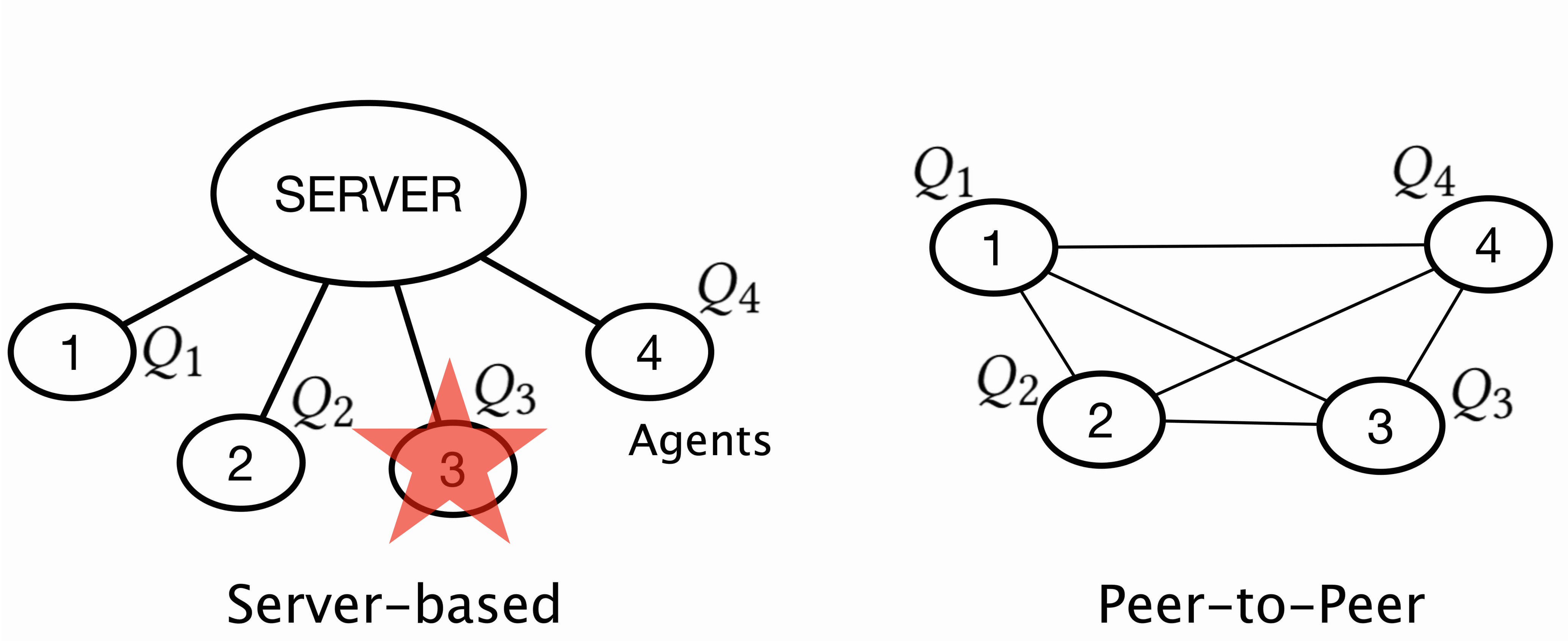


$$Q_i : \mathbb{R}^d \to \mathbb{R}$$

$$x^* \in \arg\min_{x \in \mathbb{R}^d} \sum_i Q_i(x)$$

Introduction

Fault-tolerance in distributed optimization

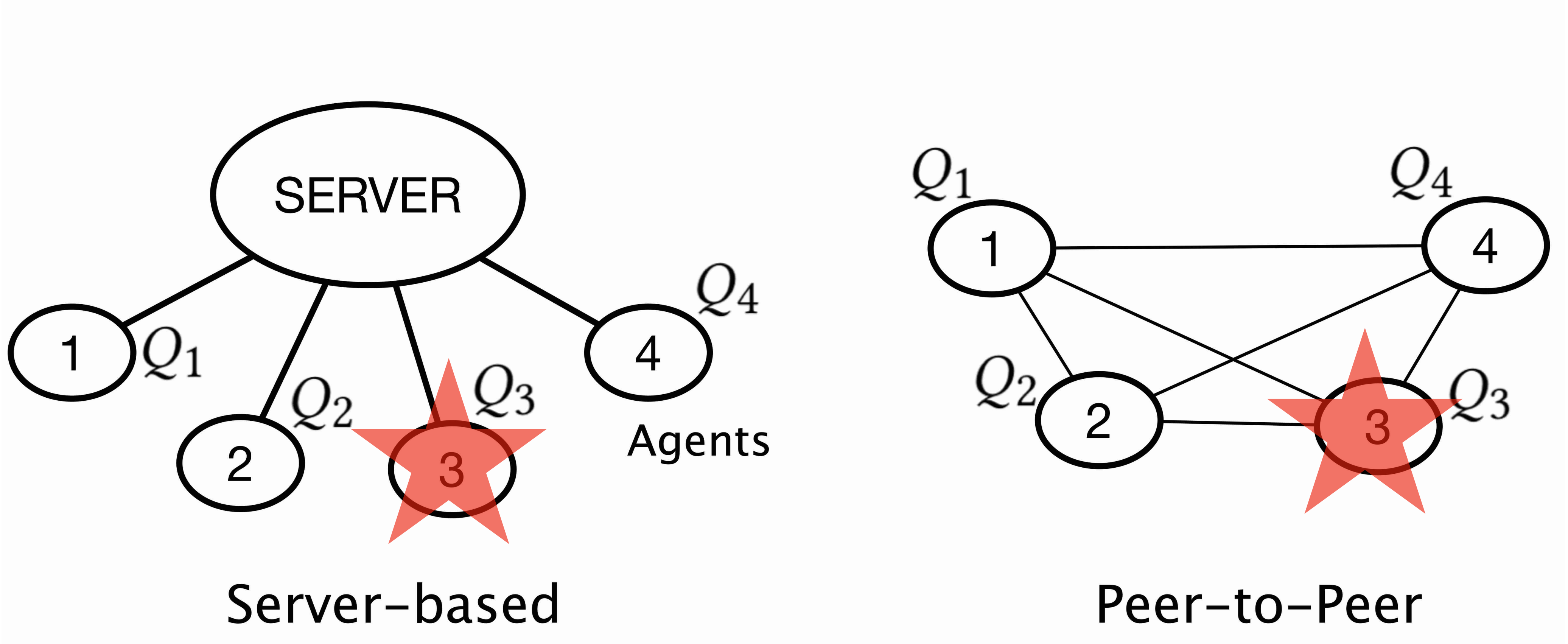


$$Q_i : \mathbb{R}^d \to \mathbb{R}$$

$$x^* \in \arg\min_{x \in \mathbb{R}^d} \sum_i Q_i(x)$$

Introduction

Fault-tolerance in distributed optimization

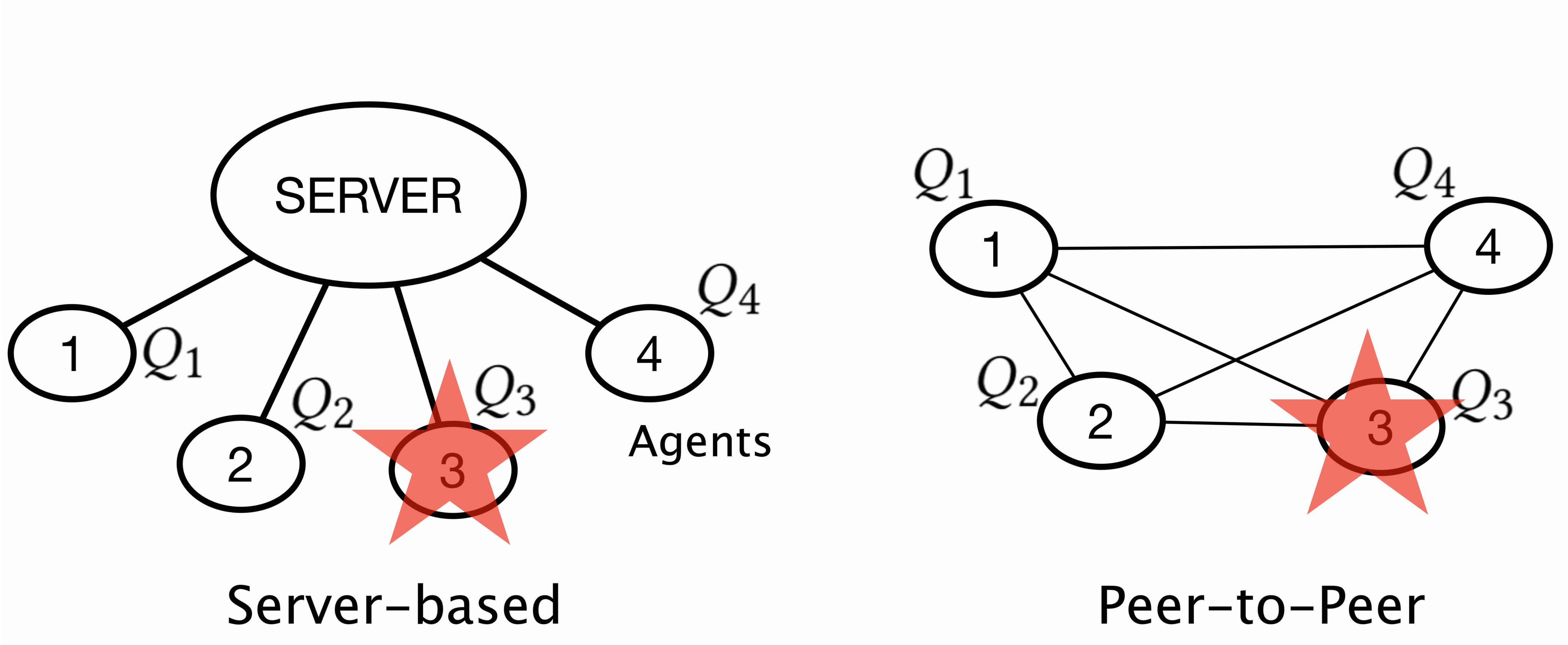


$$Q_i : \mathbb{R}^d \to \mathbb{R}$$

$$x^* \in \arg\min_{x \in \mathbb{R}^d} \sum_i Q_i(x)$$

Introduction

Fault-tolerance in distributed optimization



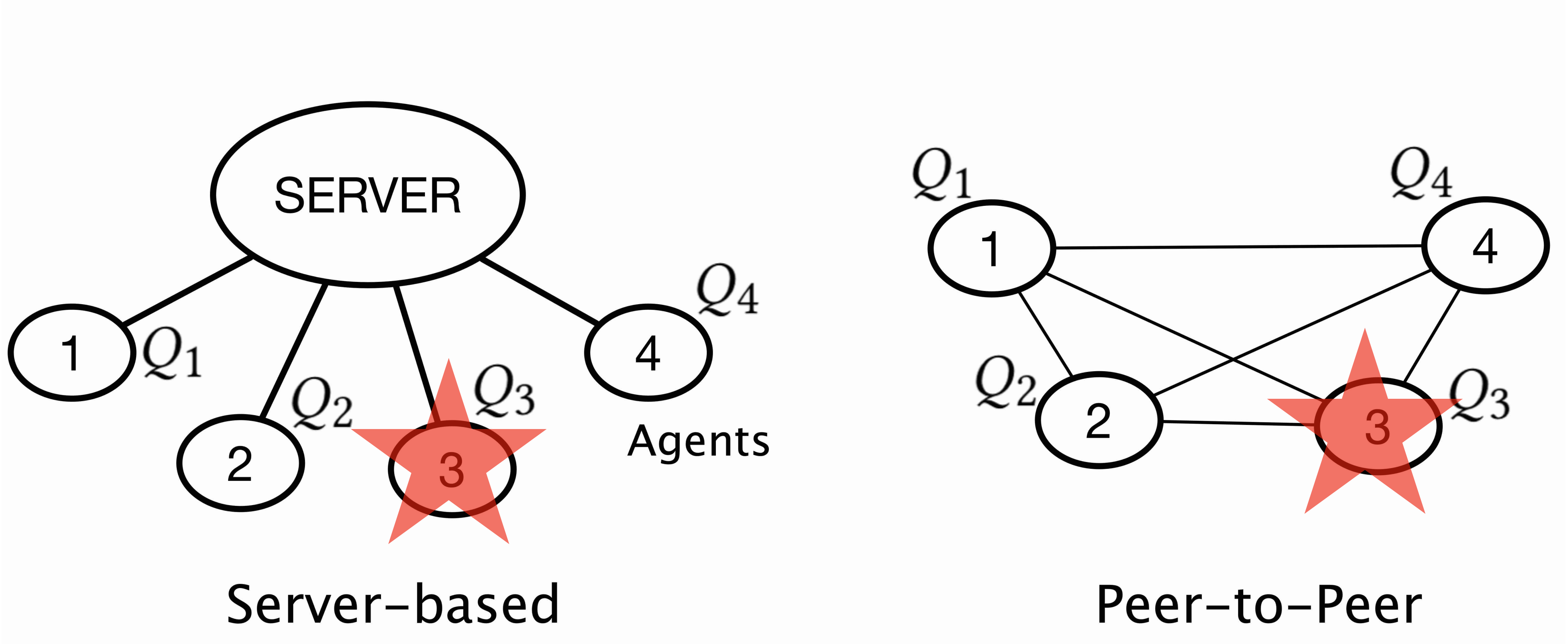
$$Q_i : \mathbb{R}^d \to \mathbb{R}$$

$$x^* \in \arg\min_{x \in \mathbb{R}^d} \sum_i Q_i(x)$$

Up to f out of n agents may be Byzantine faulty

Introduction

Fault-tolerance in distributed optimization

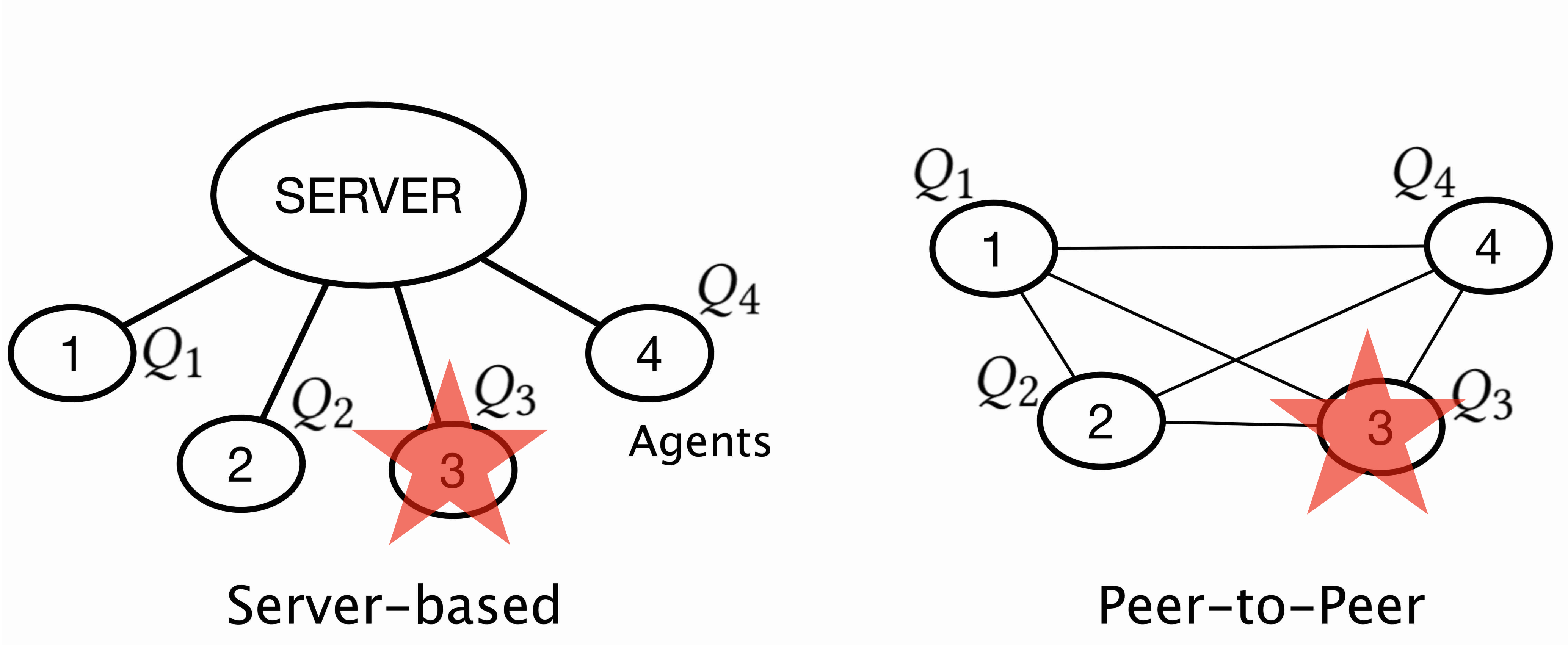


$Q_i : \mathbb{R}^d \to \mathbb{R}$

Up to f out of n agents may be Byzantine faulty

Introduction

Fault-tolerance in distributed optimization



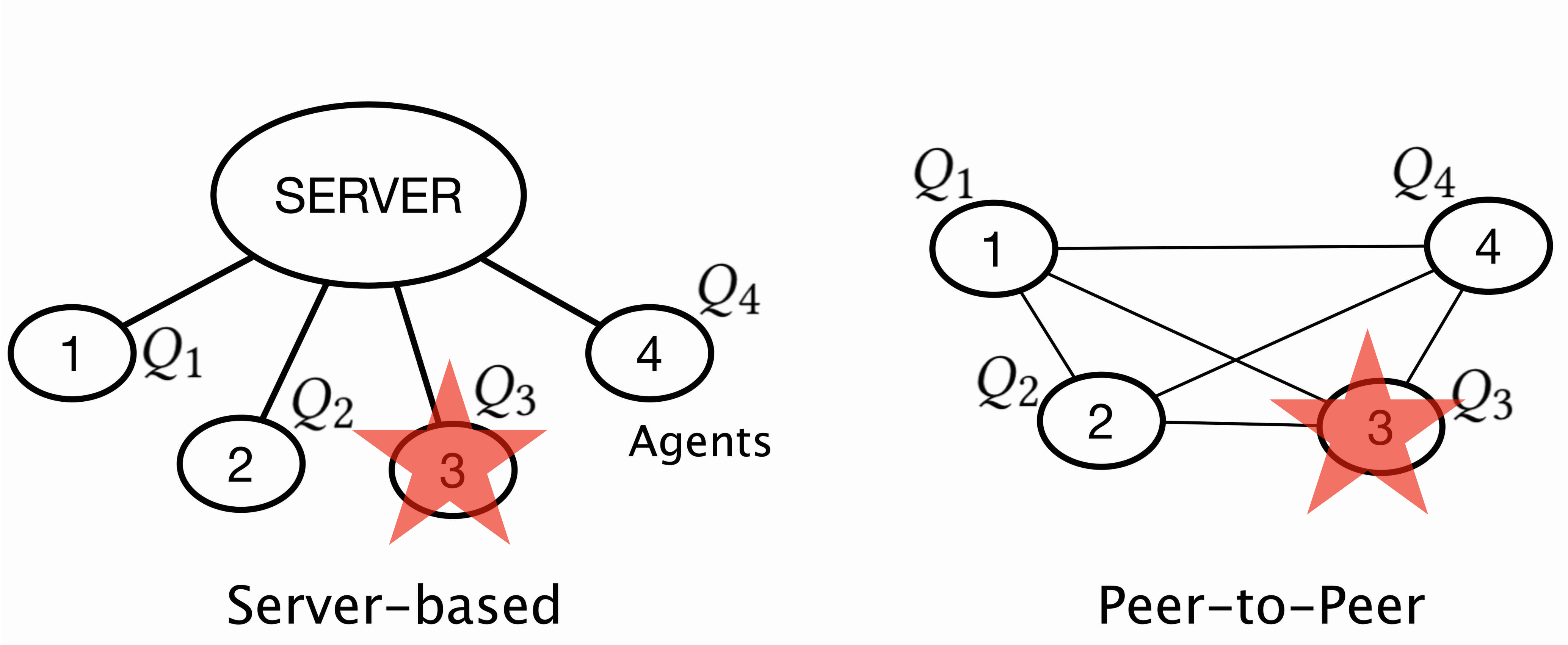
$Q_i : \mathbb{R}^d \to \mathbb{R}$

Up to f out of n agents may be Byzantine faulty

f-**resilient**: Output $\widehat{x}$ s.t. for each subset S of honest agents with $|S| = n - f$,

Introduction

Fault-tolerance in distributed optimization



$Q_i : \mathbb{R}^d \to \mathbb{R}$

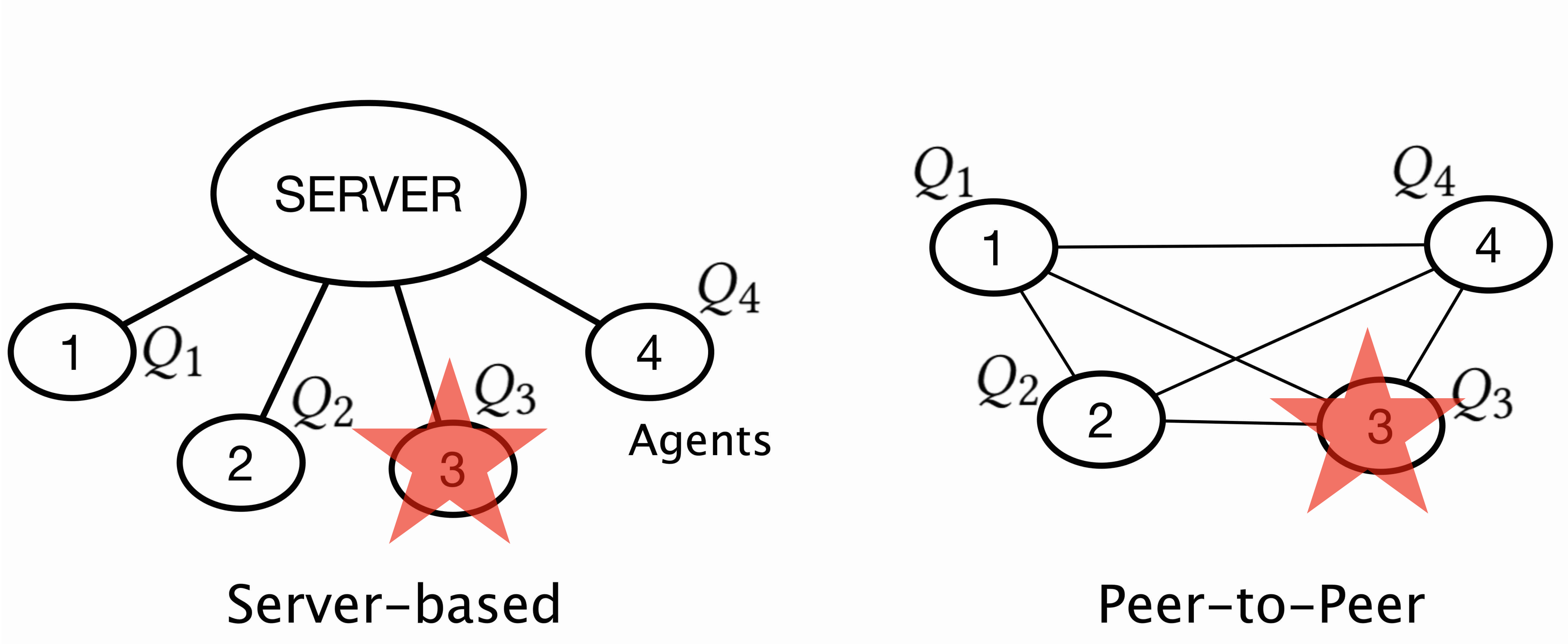
Up to f out of n agents may be Byzantine faulty

f-**resilient**: Output $\widehat{x}$ s.t. for each subset S of honest agents with $|S| = n - f$,

$$\widehat{x} \in \arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x)$$

Introduction

Fault-tolerance in distributed optimization



$Q_i : \mathbb{R}^d \to \mathbb{R}$

Up to f out of n agents may be Byzantine faulty

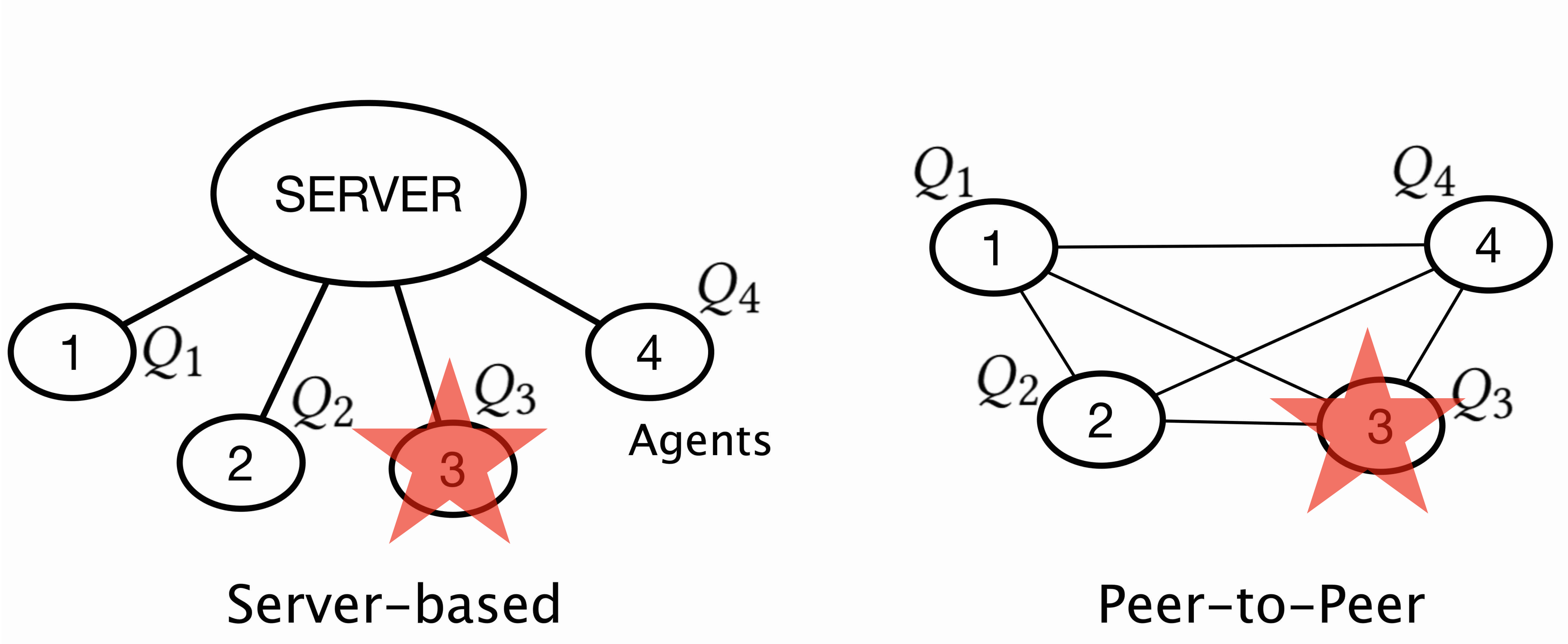
f-**resilient**: Output $\widehat{x}$ s.t. for each subset S of honest agents with $|S| = n - f$,

Exact fault-tolerance

$$\widehat{x} \in \arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x)$$

Introduction

Fault-tolerance in distributed optimization



$$Q_i : \mathbb{R}^d \to \mathbb{R}$$

Up to f out of n agents may be Byzantine faulty

f-**resilient**: Output $\hat{x}$ s.t. for each subset S of honest agents with $|S| = n - f$,

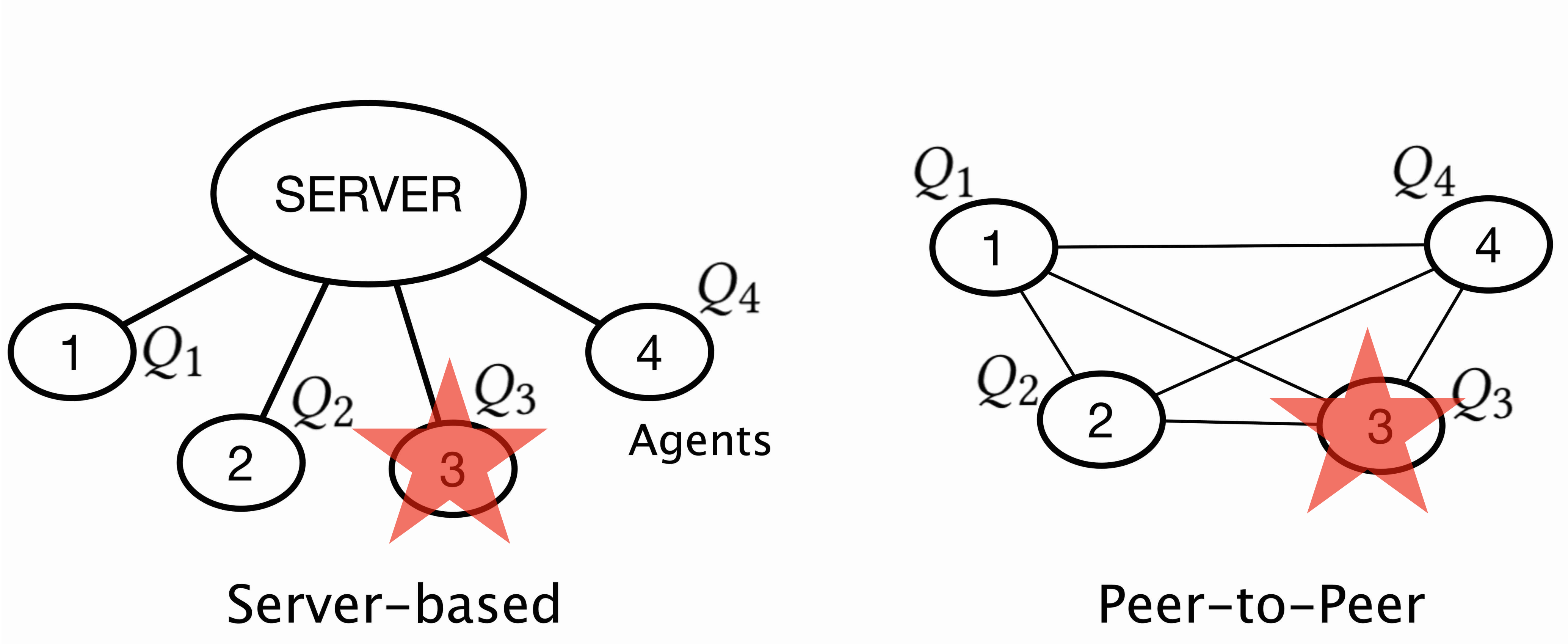
$$\hat{x} \in \arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x)$$

Exact fault-tolerance

Su & Vaidya, PODC'16

Introduction

Fault-tolerance in distributed optimization



$Q_i : \mathbb{R}^d \to \mathbb{R}$

Up to f out of n agents may be Byzantine faulty

f-**resilient**: Output $\widehat{x}$ s.t. for each subset S of honest agents with $|S| = n - f$,

$$\widehat{x} \in \arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x)$$

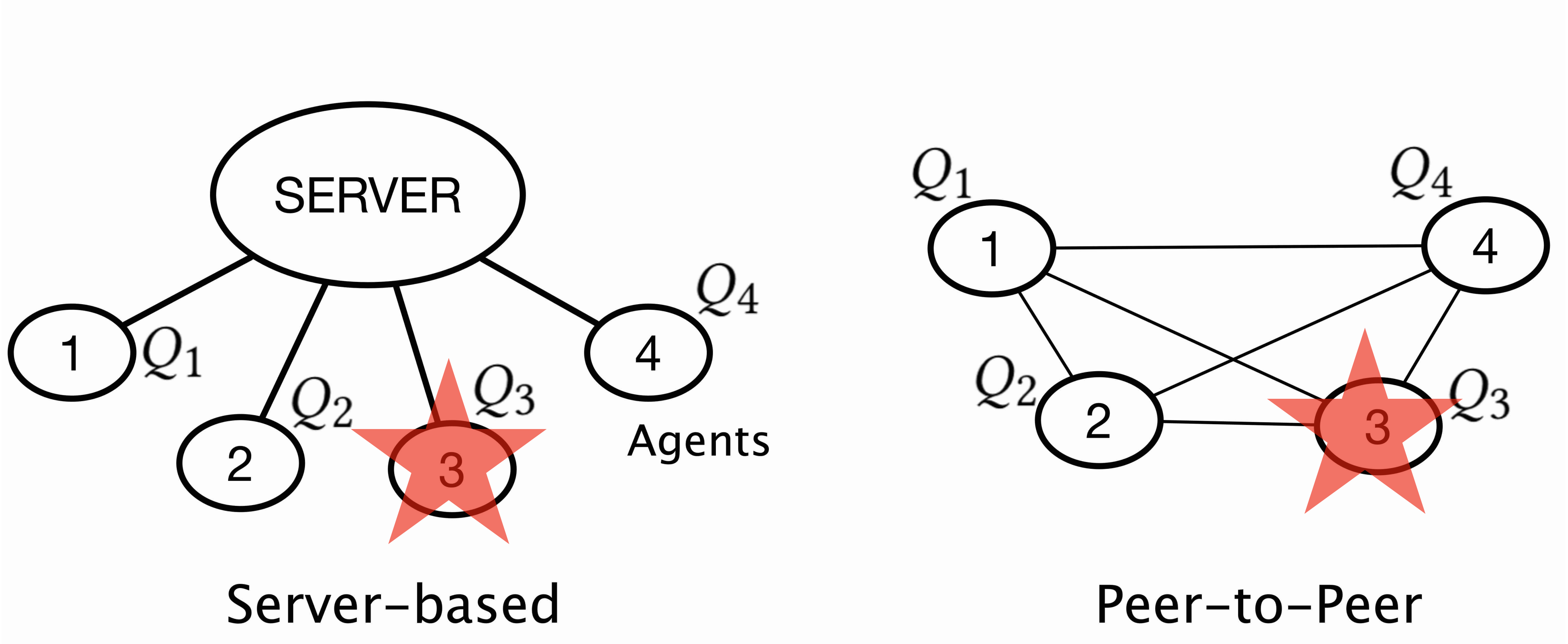
Exact fault-tolerance

Su & Vaidya, PODC'16

When f = 0 this is the standard distributed optimization goal

Introduction

Fault-tolerance in distributed optimization



$Q_i : \mathbb{R}^d \to \mathbb{R}$

Up to f out of n agents may be Byzantine faulty

f-**resilient**: Output $\widehat{x}$ s.t. for each subset S of honest agents with $|S| = n - f$,

$$\widehat{x} \in \arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x)$$

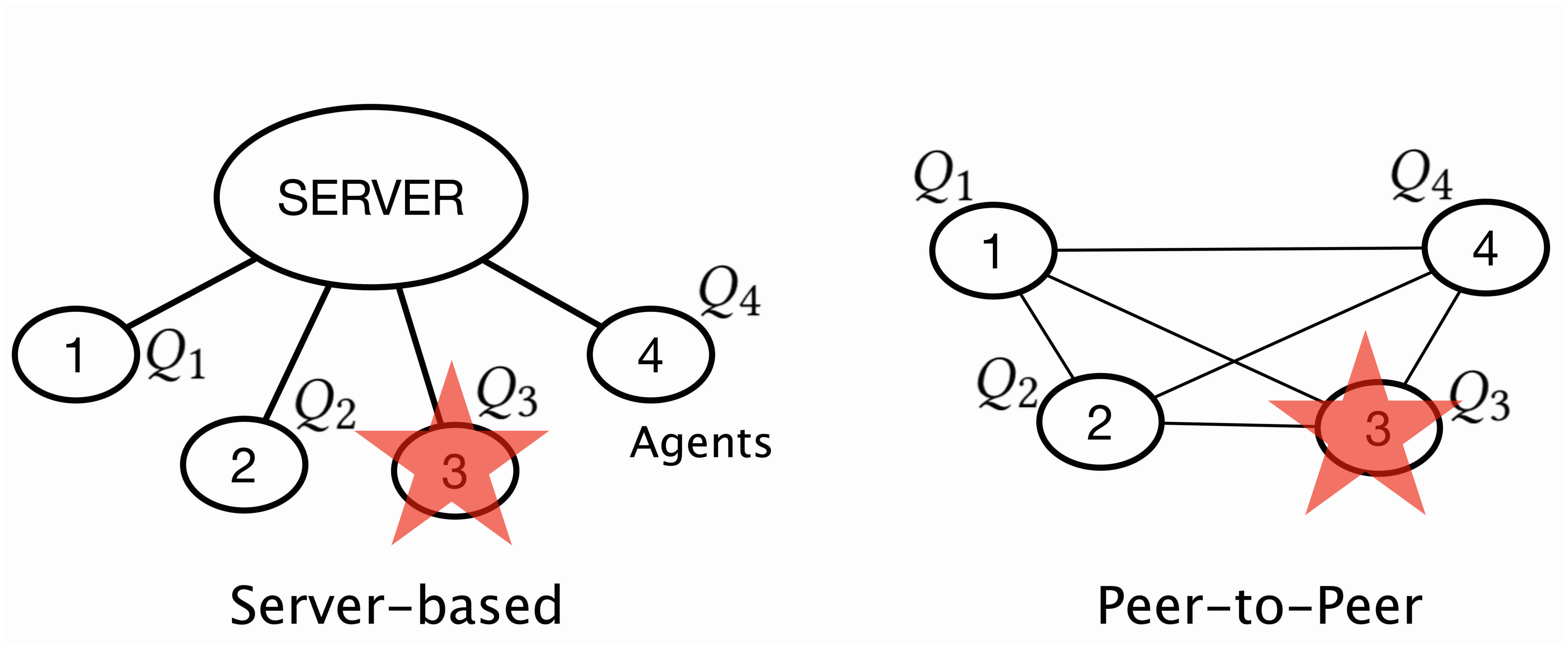
Exact fault-tolerance

Su & Vaidya, PODC'16

When f = 0 this is the standard distributed optimization goal

Introduction

Fault-tolerance in distributed optimization



$$Q_i : \mathbb{R}^d \to \mathbb{R}$$

$$x^* \in \arg\min_{x \in \mathbb{R}^d} \sum_i Q_i(x)$$

Up to f out of n agents may be Byzantine faulty

f-**resilient**: Output $\widehat{x}$ s.t. for each subset S of honest agents with $|S| = n - f$,

$$\widehat{x} \in \arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x)$$

Exact fault-tolerance

Su & Vaidya, PODC'16

When f = 0 this is the standard distributed optimization goal

Exact Fault-Tolerance *needs* Strong Redundancy

Exact Fault-Tolerance *needs* Strong Redundancy

Theorem 1

Exact fault-tolerance achievable iff $2f$-*redundancy* holds true

Exact Fault-Tolerance *needs* Strong Redundancy

Theorem 1

Exact fault-tolerance achievable iff $2f$-*redundancy* holds true

Gupta & Vaidya, PODC'20

Exact Fault-Tolerance *needs* Strong Redundancy

Theorem 1

Exact fault-tolerance achievable iff $2f$-*redundancy* holds true

Gupta & Vaidya, PODC'20

$2f$-**redundancy**: Subsets $S, \widehat{S} \subseteq \{1,\ldots,n\}$ with $|S| = n - f$, $|\widehat{S}| \geq n - 2f$, and $\widehat{S} \subseteq S$,

$$\arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x) = \arg\min_{x \in \mathbb{R}^d} \sum_{i \in \widehat{S}} Q_i(x)$$

Exact Fault-Tolerance *needs* Strong Redundancy

Theorem 1

Exact fault-tolerance achievable iff $2f$-*redundancy* holds true

Gupta & Vaidya, PODC'20

$2f$-**redundancy**: Subsets $S, \widehat{S} \subseteq \{1,\ldots,n\}$ with $|S| = n - f$, $|\widehat{S}| \geq n - 2f$, and $\widehat{S} \subseteq S$,

$$\arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x) = \arg\min_{x \in \mathbb{R}^d} \sum_{i \in \widehat{S}} Q_i(x)$$

Minimizer of the agg. of $n - 2f$ honest costs

Exact Fault-Tolerance *needs* Strong Redundancy

Theorem 1

Exact fault-tolerance achievable iff $2f$-*redundancy* holds true

Gupta & Vaidya, PODC'20

$2f$-**redundancy**: Subsets $S, \widehat{S} \subseteq \{1,\ldots,n\}$ with $|S| = n - f$, $|\widehat{S}| \geq n - 2f$, and $\widehat{S} \subseteq S$,

$$\arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x) = \arg\min_{x \in \mathbb{R}^d} \sum_{i \in \widehat{S}} Q_i(x)$$

Minimizer of the agg. of $n - 2f$ honest costs $\longleftrightarrow$

Exact Fault-Tolerance *needs* Strong Redundancy

Theorem 1

Exact fault-tolerance achievable iff $2f$-*redundancy* holds true

Gupta & Vaidya, PODC'20

$2f$-**redundancy**: Subsets $S, \widehat{S} \subseteq \{1,\ldots,n\}$ with $|S| = n - f$, $|\widehat{S}| \geq n - 2f$, and $\widehat{S} \subseteq S$,

$$\arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x) = \arg\min_{x \in \mathbb{R}^d} \sum_{i \in \widehat{S}} Q_i(x)$$

Minimizer of the agg. of $n - 2f$ honest costs ⟷ Minimizer of the agg. of *all* honest costs

Exact Fault-Tolerance *needs* Strong Redundancy

Theorem 1

Exact fault-tolerance achievable iff $2f$-*redundancy* holds true

Gupta & Vaidya, PODC'20

$2f$-**redundancy**: Subsets $S, \widehat{S} \subseteq \{1,\ldots,n\}$ with $|S| = n - f$, $|\widehat{S}| \geq n - 2f$, and $\widehat{S} \subseteq S$,

$$\arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x) = \arg\min_{x \in \mathbb{R}^d} \sum_{i \in \widehat{S}} Q_i(x)$$

Minimizer of the agg. of $n - 2f$ honest costs $\longleftrightarrow$ Minimizer of the agg. of *all* honest costs

$2f$-redundancy applicable in many scenarios: **learning, swarm robotics, etc**

Exact Fault-Tolerance *needs* Strong Redundancy

Theorem 1

Exact fault-tolerance achievable iff $2f$-*redundancy* holds true

Gupta & Vaidya, PODC'20

$2f$-**redundancy**: Subsets S, $\widehat{S} \subseteq \{1,\ldots,n\}$ with $|S| = n - f$, $|\widehat{S}| \geq n - 2f$, and $\widehat{S} \subseteq S$,

$$\arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x) = \arg\min_{x \in \mathbb{R}^d} \sum_{i \in \widehat{S}} Q_i(x)$$

Minimizer of the agg. of $n - 2f$ honest costs $\Longleftrightarrow$ Minimizer of the agg. of *all* honest costs

Exact Fault-Tolerance *needs* Strong Redundancy

Theorem 1

Exact fault-tolerance achievable iff $2f$-*redundancy* holds true

Gupta & Vaidya, PODC'20

$2f$-**redundancy**: Subsets $S, \widehat{S} \subseteq \{1,\ldots,n\}$ with $|S| = n - f$, $|\widehat{S}| \geq n - 2f$, and $\widehat{S} \subseteq S$,

$$\arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x) = \arg\min_{x \in \mathbb{R}^d} \sum_{i \in \widehat{S}} Q_i(x)$$

Minimizer of the agg. of $n - 2f$ honest costs $\longleftrightarrow$ Minimizer of the agg. of *all* honest costs

$2f$-**redundancy is difficult in practical settings**; noise, uncertainties, etc.

Exact Fault-Tolerance *needs* Strong Redundancy

Theorem 1

Exact fault-tolerance achievable iff $2f$-*redundancy* holds true

Gupta & Vaidya, PODC'20

$2f$-**redundancy**: Subsets $S, \widehat{S} \subseteq \{1,\ldots,n\}$ with $|S| = n - f$, $|\widehat{S}| \geq n - 2f$, and $\widehat{S} \subseteq S$,

$$\arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x) = \arg\min_{x \in \mathbb{R}^d} \sum_{i \in \widehat{S}} Q_i(x)$$

Minimizer of the agg. of $n - 2f$ honest costs ⟷ Minimizer of the agg. of *all* honest costs

$2f$-**redundancy is difficult in practical settings**; noise, uncertainties, etc.

Inadequate to characterize relationship between redundancy and resilience!

Approximate Fault-Tolerance

Liu et al., PODC'21

Approximate Fault-Tolerance

Liu et al., PODC'21

(f, ϵ)-**resilient**: Output $\widehat{x}$ s.t. for each subset S of honest agents with $|S| = n - f$,

$$\text{dist}\left(\widehat{x},\ \arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x) \right) \leq \epsilon$$

Approximate Fault-Tolerance

Liu et al., PODC'21

(f, ϵ)-**resilient**: Output $\widehat{x}$ s.t. for each subset S of honest agents with $|S| = n - f$,

$$\mathrm{dist}\left(\widehat{x}, \arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x)\right) \le \epsilon$$

Approximate a minimizer of the aggregate honest cost with ϵ-accuracy

Approximate Fault-Tolerance

Liu et al., PODC'21

(f, ϵ)-**resilient**: Output $\widehat{x}$ s.t. for each subset S of honest agents with $|S| = n - f$,

$$\mathrm{dist}\left(\widehat{x},\ \arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x)\right) \leq \epsilon$$

Approximate a minimizer of the aggregate honest cost with ϵ-accuracy

Relaxing 2f-redundancy:

Approximate Fault-Tolerance

Liu et al., PODC'21

(f, ϵ)-**resilient**: Output $\widehat{x}$ s.t. for each subset S of honest agents with $|S| = n - f$,

$$\text{dist}\left(\widehat{x},\ \arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x)\right) \leq \epsilon$$

Approximate a minimizer of the aggregate honest cost with ϵ-accuracy

Relaxing 2f-redundancy:

$\left(2f, \epsilon\right)$-**redundancy**: Subsets $S, \widehat{S} \subseteq \{1,\ldots,n\}$ with $|S| = n - f$, $|\widehat{S}| \geq n - 2f$, and $\widehat{S} \subseteq S$,

$$\text{dist}\left(\arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x),\ \ \arg\min_{x \in \mathbb{R}^d} \sum_{i \in \widehat{S}} Q_i(x)\right) \leq \epsilon$$

Approximate Fault-Tolerance

Liu et al., PODC'21

(f, ϵ)**-resilient**: Output $\widehat{x}$ s.t. for each subset S of honest agents with $|S| = n - f$,

$$\mathrm{dist}\left(\widehat{x},\ \arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x)\right) \leq \epsilon$$

Approximate a minimizer of the aggregate honest cost with ϵ-accuracy

Relaxing 2f-redundancy:

$\left(2f, \epsilon\right)$**-redundancy**: Subsets $S, \widehat{S} \subseteq \{1,\ldots,n\}$ with $|S| = n - f$, $|\widehat{S}| \geq n - 2f$, and $\widehat{S} \subseteq S$,

$$\mathrm{dist}\left(\arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x),\ \arg\min_{x \in \mathbb{R}^d} \sum_{i \in \widehat{S}} Q_i(x)\right) \leq \epsilon$$

Minimiser of the agg. of any $n - 2f$ honest costs is ϵ-close to that of *all* the honest costs

More on $(2f, \epsilon)$-redundancy

More on $(2f, \epsilon)$-redundancy

ϵ quantifies the loss of redundancy

More on $(2f, \epsilon)$-redundancy

ϵ quantifies the loss of redundancy

Satisfied by every system for varied value of ϵ

More on $(2f, \epsilon)$-redundancy

ϵ quantifies the loss of redundancy

Satisfied by every system for varied value of ϵ

Enables derivation of lower and upper bounds on Byzantine resilience*

More on $(2f, \epsilon)$-redundancy

ϵ quantifies the loss of redundancy

Satisfied by every system for varied value of ϵ

Enables derivation of lower and upper bounds on Byzantine resilience*

* Generalize prior results on resilience in distributed optimization, learning, state estimation, and swarm robotics.

More on $(2f, \epsilon)$-redundancy

ϵ quantifies the loss of redundancy

Satisfied by every system for varied value of ϵ

Enables derivation of lower and upper bounds on Byzantine resilience*

In distributed learning: we can characterize resilience versus heterogeneity

* Generalize prior results on resilience in distributed optimization, learning, state estimation, and swarm robotics.

We Show that … *

Liu et al., PODC'21

* In *deterministic* setting.

We Show that … *

Liu et al., PODC'21

Lower Bound

* In *deterministic* setting.

We Show that … *

Liu et al., PODC'21

Lower Bound

$$\boxed{\begin{array}{c} (f, \epsilon)\text{-resilience} \\ \textit{only if } \ (2f, \epsilon)\text{-redundancy} \end{array}}$$

* In *deterministic* setting.

We Show that … *

Liu et al., PODC'21

Lower Bound

(f, ϵ)-resilience

only if $(2f, \epsilon)$-redundancy

Upper Bound

* In *deterministic* setting.

We Show that … *

Liu et al., PODC'21

Lower Bound

(f, ϵ)-resilience

only if $(2f, \epsilon)$-redundancy

Upper Bound

If $(2f, \epsilon)$-redundancy

then $(f, 2\epsilon)$-resilience

* In *deterministic* setting.

We Show that … *

Liu et al., PODC'21

Lower Bound

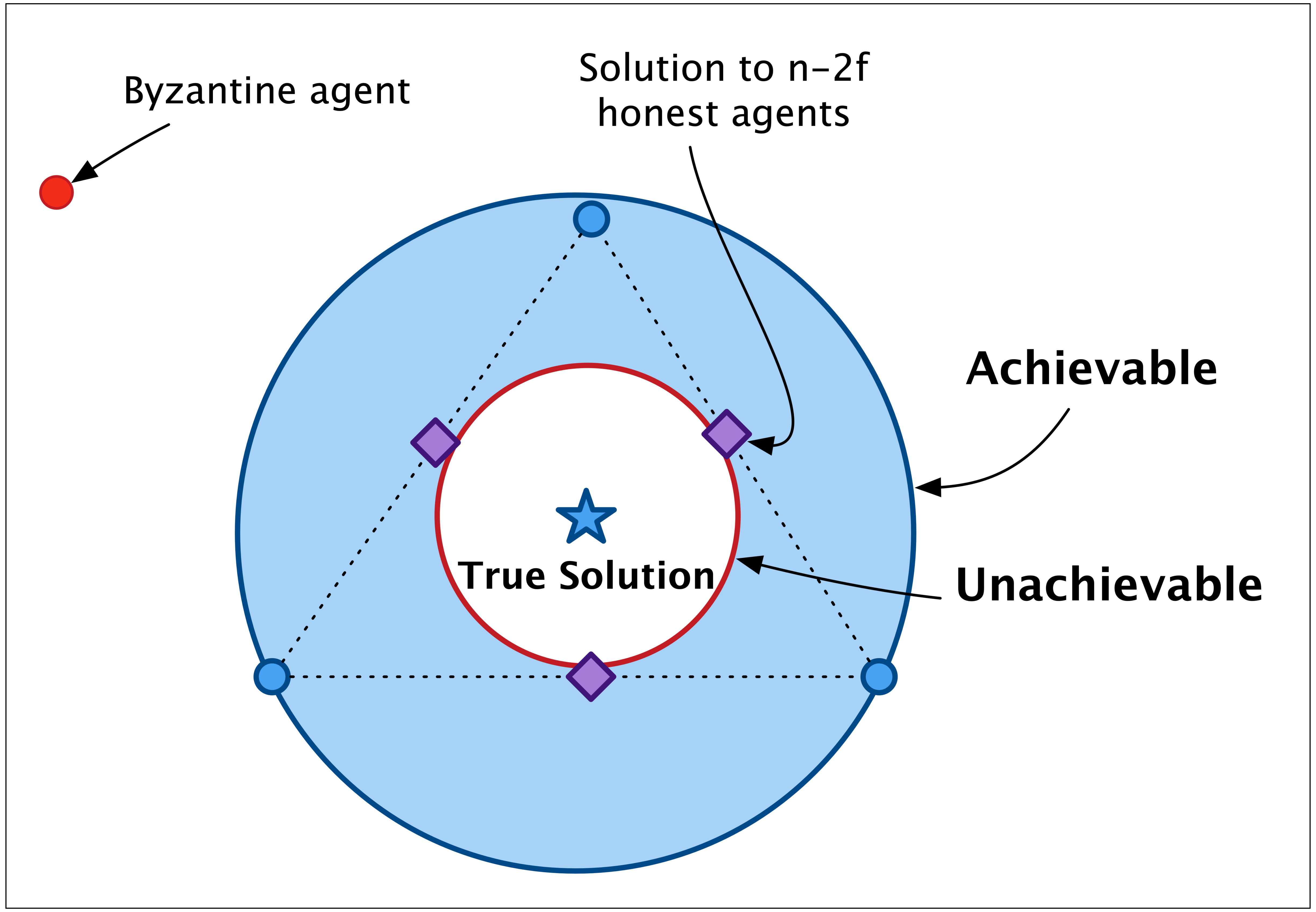
(f, ϵ)-resilience

only if $(2f, \epsilon)$-redundancy

Upper Bound

If $(2f, \epsilon)$-redundancy

then $(f, 2\epsilon)$-resilience



* In *deterministic* setting.

We Show that … *

Liu et al., PODC'21

Lower Bound

(f, ϵ)-resilience

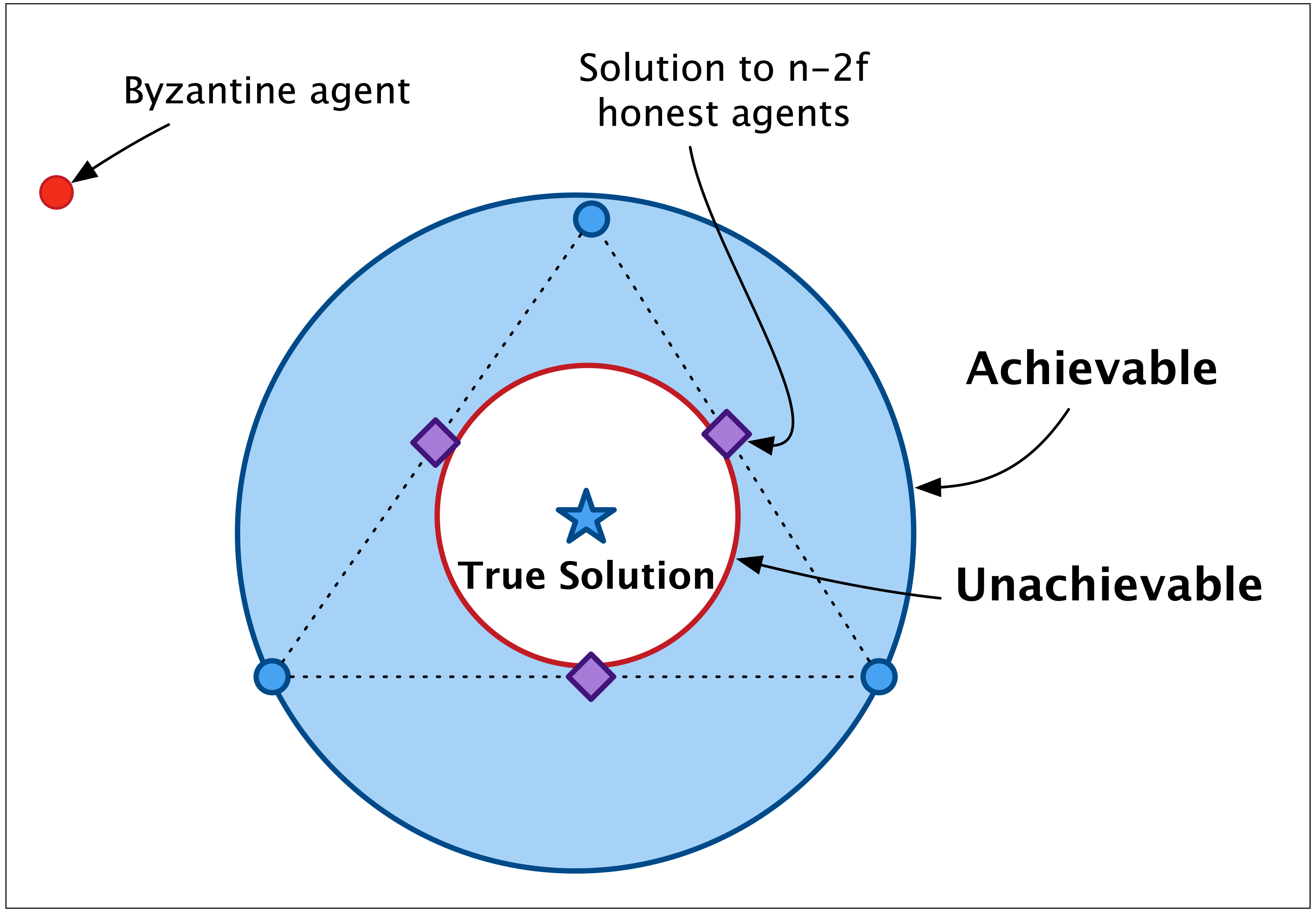
only if $(2f, \epsilon)$-redundancy

Theorem 2

Upper Bound

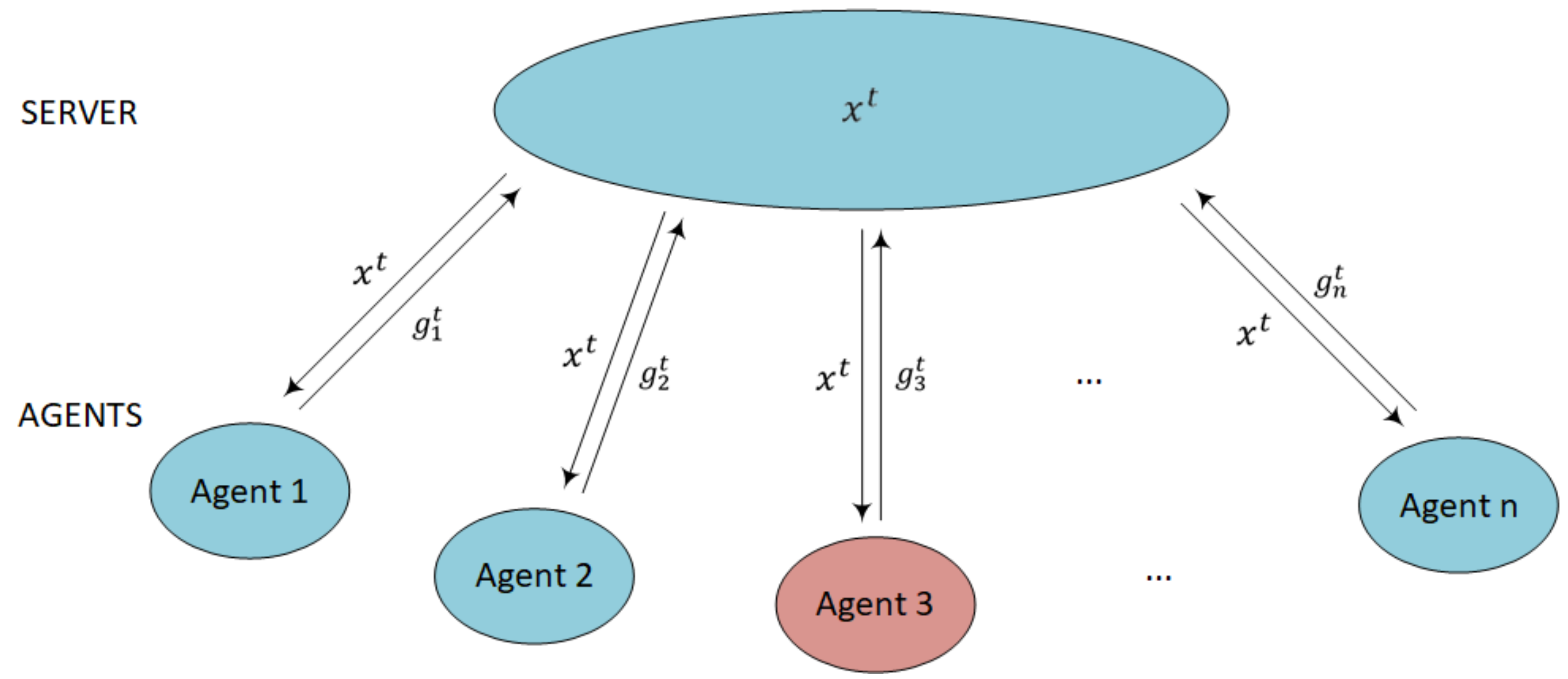
If $(2f, \epsilon)$-redundancy

then $(f, 2\epsilon)$-resilience



* In *deterministic* setting.

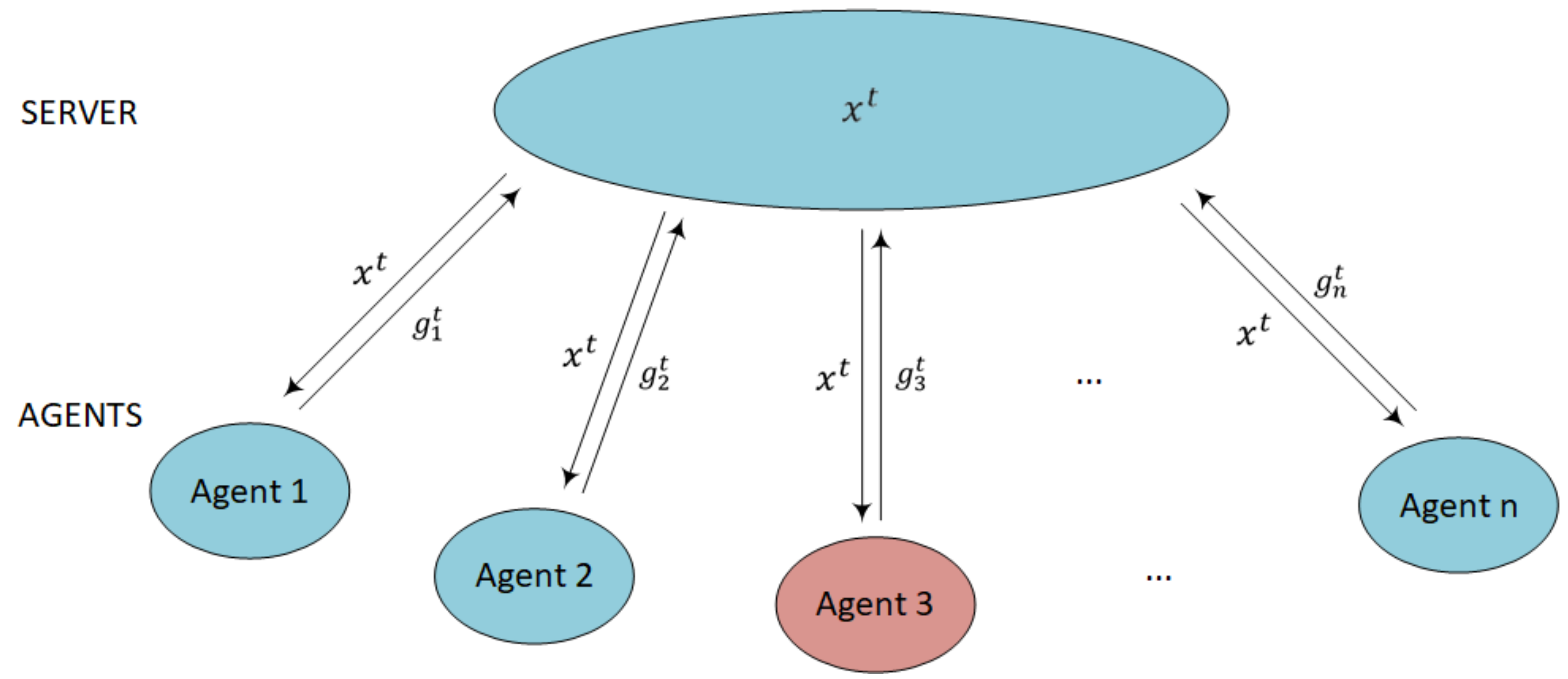
Fault-tolerance in DGD



* a.k.a., Byzantine robust **gradient aggregation rule (GAR)**

Fault-tolerance in DGD

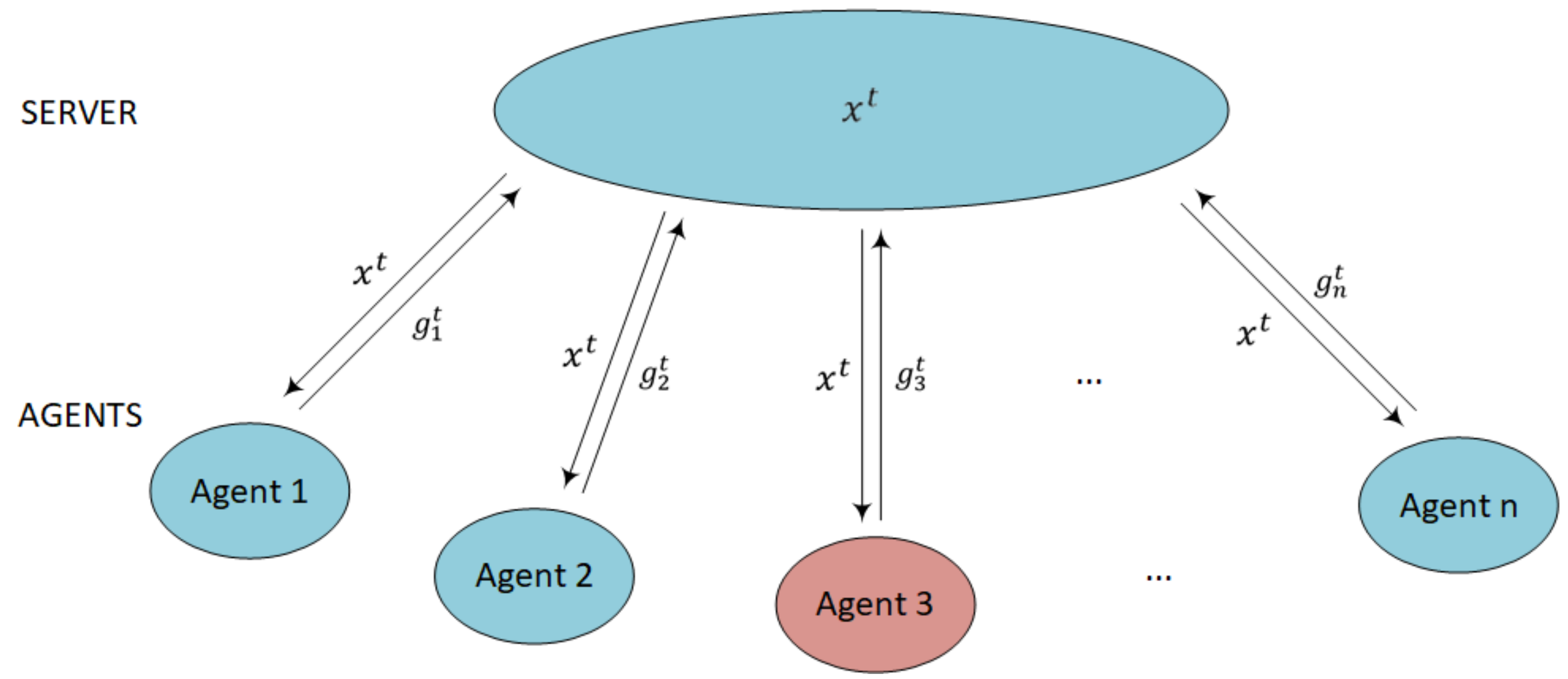
Byzantine robust gradient-filters



* a.k.a., Byzantine robust **gradient aggregation rule (GAR)**

Fault-tolerance in DGD

Byzantine robust gradient-filters

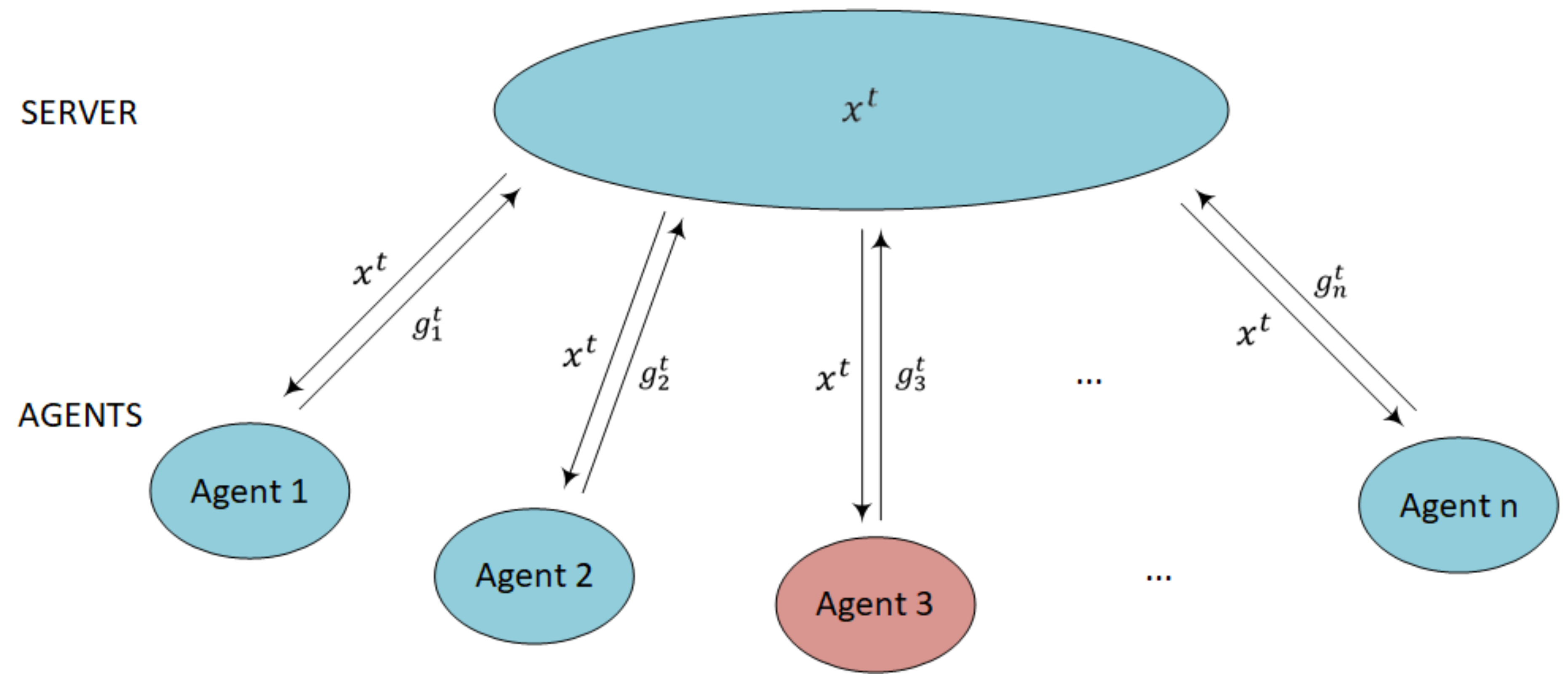


- Presence of Byzantine gradients warrants **gradient filtering***

* a.k.a., Byzantine robust **gradient aggregation rule (GAR)**

Fault-tolerance in DGD

Byzantine robust gradient-filters

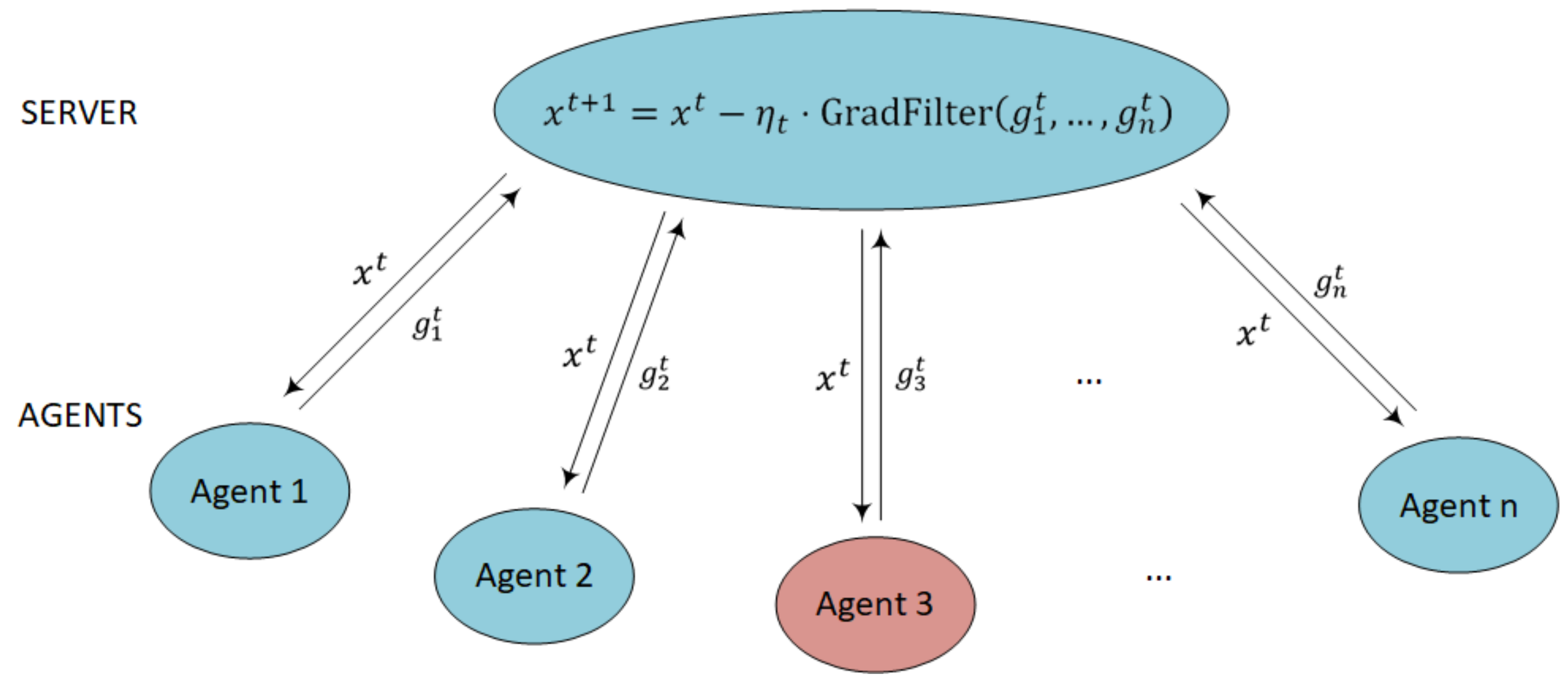


- Presence of Byzantine gradients warrants **gradient filtering***
- GradFilter$(g_1^t, \ldots, g_n^t)$ *robustly* aggregates gradients, instead of simple averaging

* a.k.a., Byzantine robust **gradient aggregation rule (GAR)**

Fault-tolerance in DGD

Byzantine robust gradient-filters

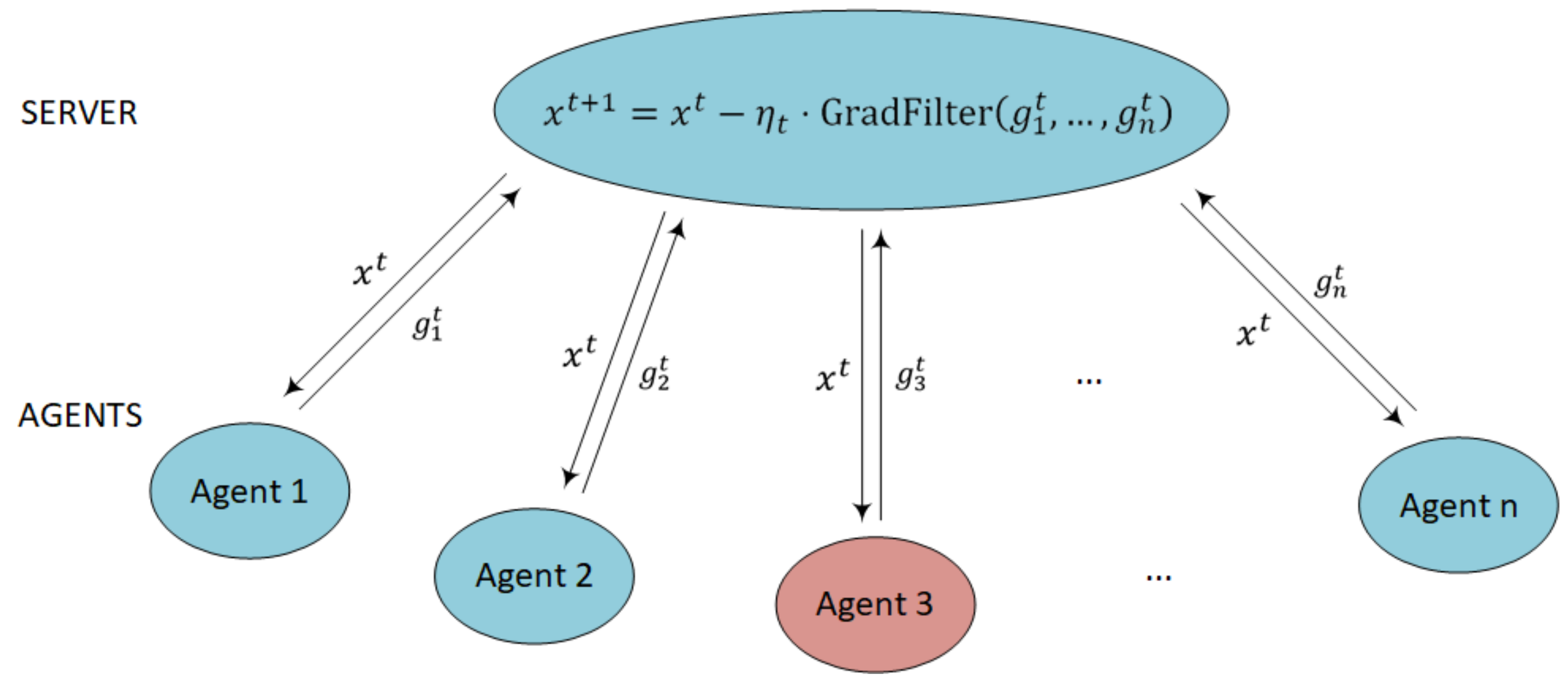


- Presence of Byzantine gradients warrants **gradient filtering***
- GradFilter$(g_1^t, \dots, g_n^t)$ *robustly* aggregates gradients, instead of simple averaging

* a.k.a., Byzantine robust **gradient aggregation rule (GAR)**

Fault-tolerance in DGD

Byzantine robust gradient-filters



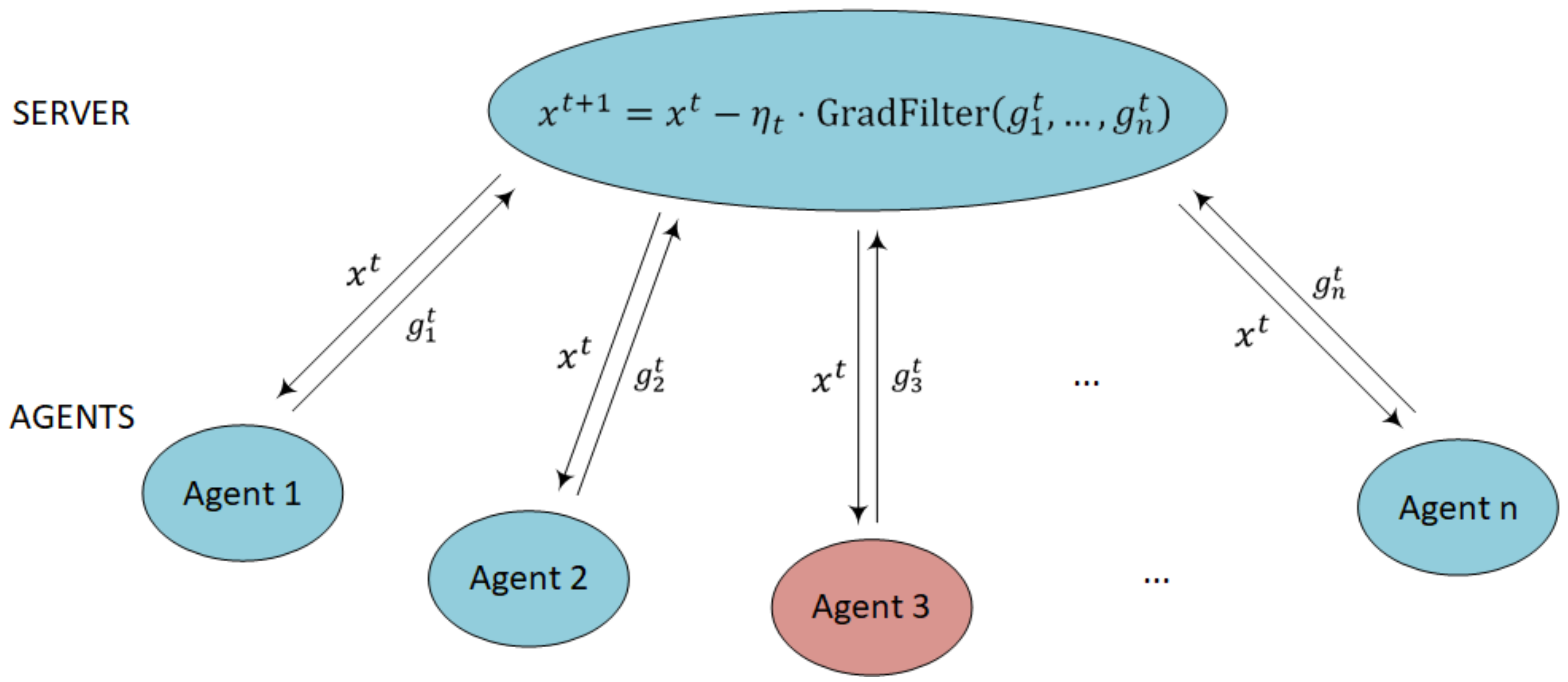
- Presence of Byzantine gradients warrants **gradient filtering***
- GradFilter$(g_1^t, \dots, g_n^t)$ *robustly* aggregates gradients, instead of simple averaging

$$\boxed{x^{t+1} = x^t - \eta_t \cdot \text{GradFilter}(g_1^t, \dots, g_n^t)}$$

* a.k.a., Byzantine robust **gradient aggregation rule (GAR)**

Fault-tolerance in DGD

Byzantine robust gradient-filters



- Presence of Byzantine gradients warrants **gradient filtering***
- GradFilter$(g_1^t, \ldots, g_n^t)$ *robustly* aggregates gradients, instead of simple averaging

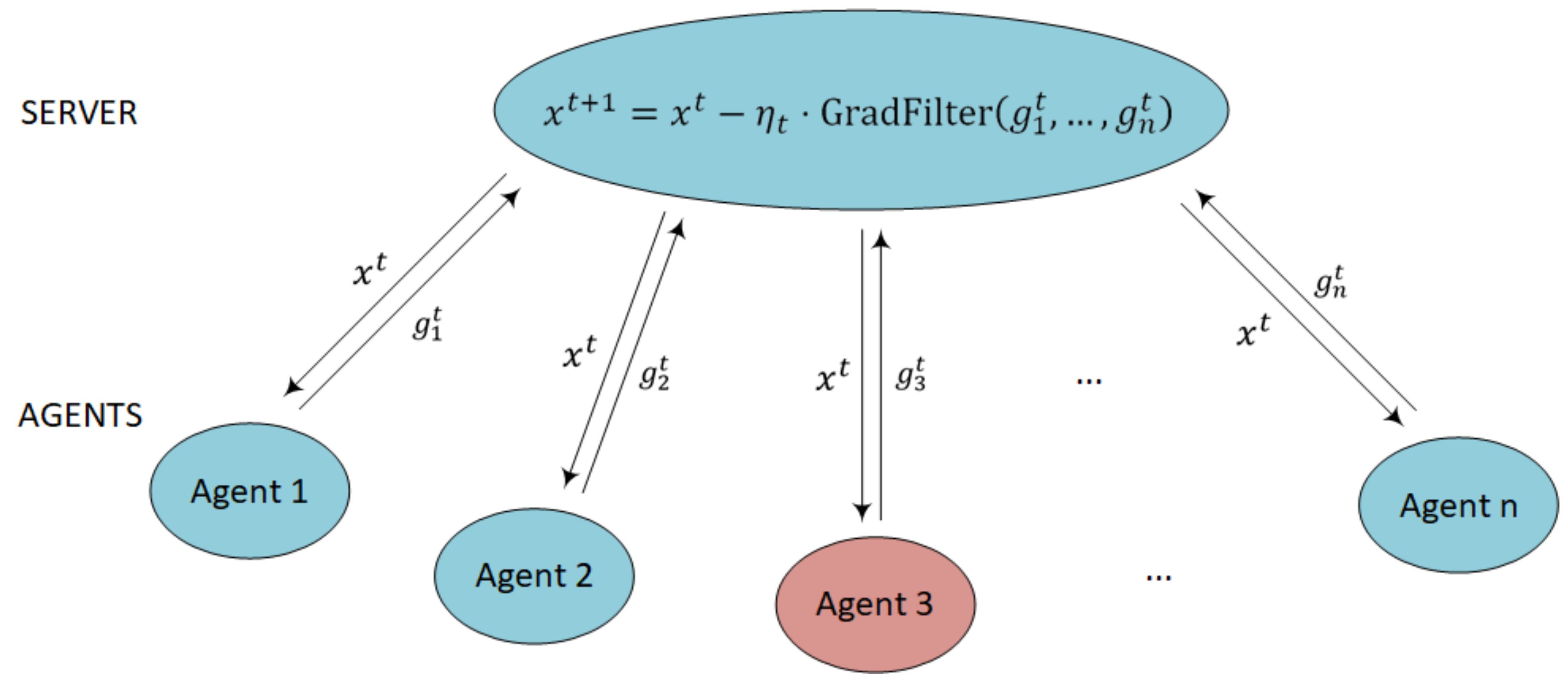
$$x^{t+1} = x^t - \eta_t \cdot \text{GradFilter}(g_1^t, \ldots, g_n^t)$$

- Prominent gradient-filters -

* a.k.a., Byzantine robust **gradient aggregation rule (GAR)**

Fault-tolerance in DGD

Byzantine robust gradient-filters



- Presence of Byzantine gradients warrants **gradient filtering***
- GradFilter$(g_1^t, \ldots, g_n^t)$ *robustly* aggregates gradients, instead of simple averaging

$$x^{t+1} = x^t - \eta_t \cdot \textbf{GradFilter}(g_1^t, \ldots, g_n^t)$$

- Prominent gradient-filters -

KRUM [Blanchard et al., NIPS'17], **GMoM** [Chen et al., SIGMETRICS'18];
Bulyan [El-Mhamdi et al., ICML'18], **CWTM** [Yin et al., ICML'18];
CGE [Gupta & Vaidya, PODC'20]

* a.k.a., Byzantine robust **gradient aggregation rule (GAR)**

Examples

Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

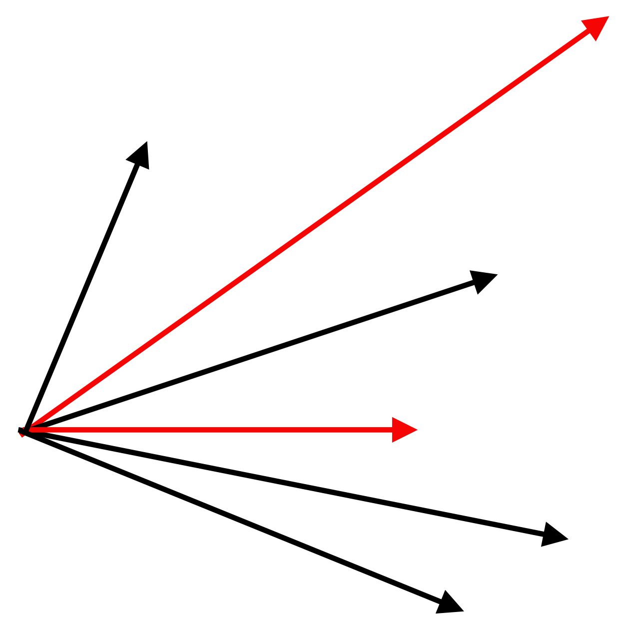
Remove gradients with f-largest norms

Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

Remove gradients with f-largest norms

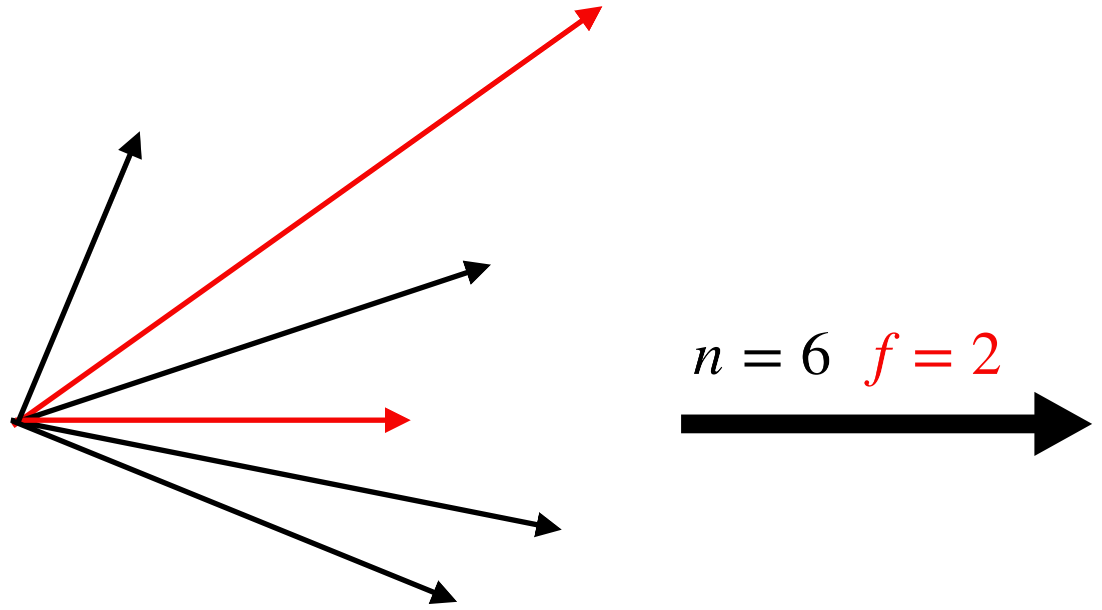


Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

Remove gradients with f-largest norms

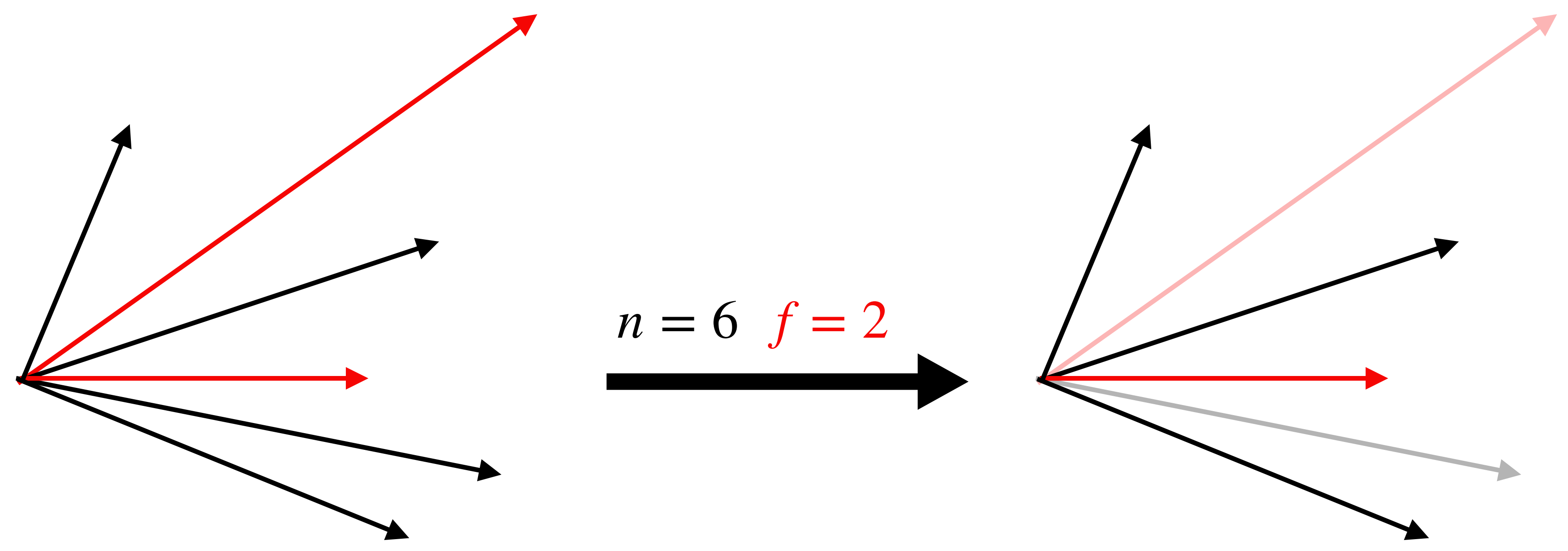


Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

Remove gradients with f-largest norms

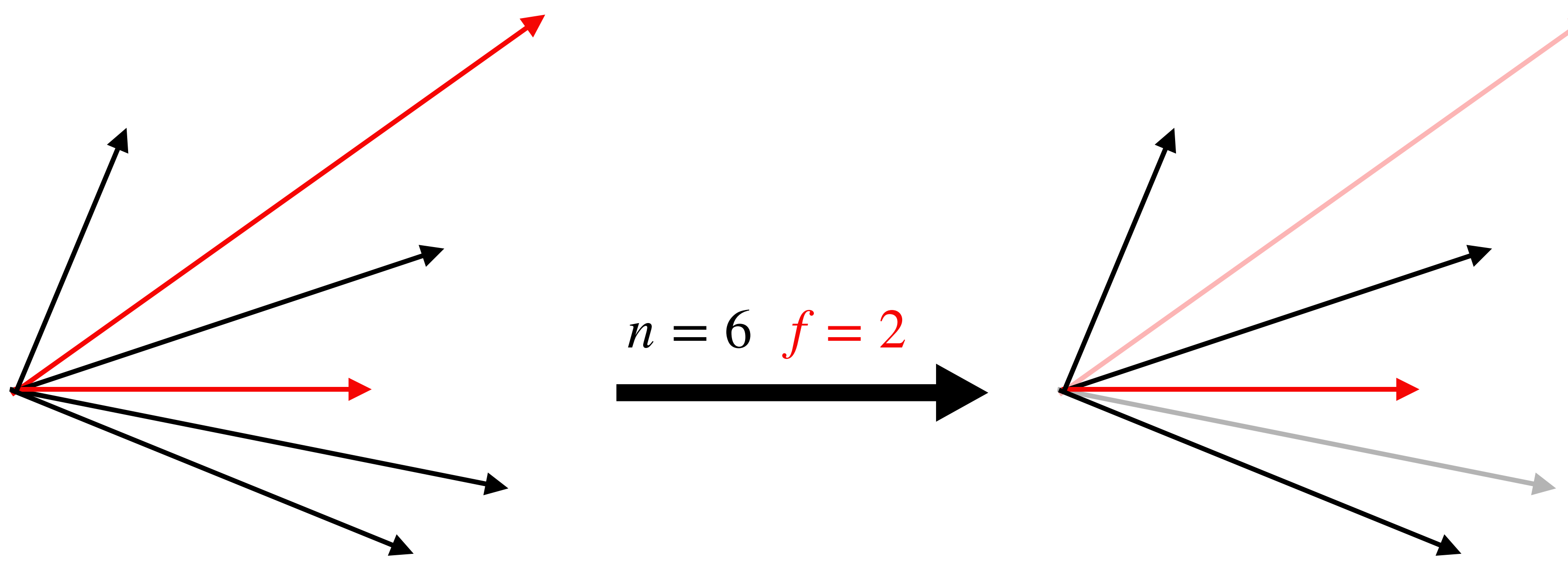


Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

Remove gradients with f-largest norms



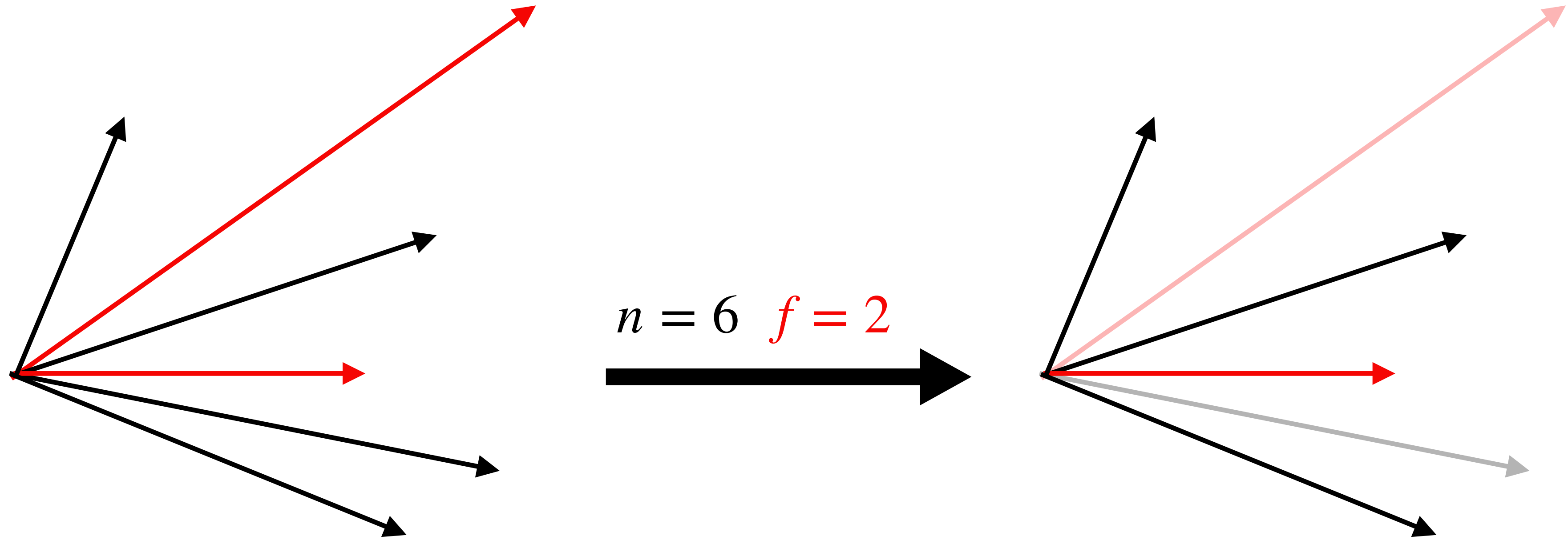
- **C**oordinate-**w**ise **t**rimmed **m**ean (**CWTM**)

Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

Remove gradients with f-largest norms



- **C**oordinate-**w**ise **t**rimmed **m**ean (**CWTM**)

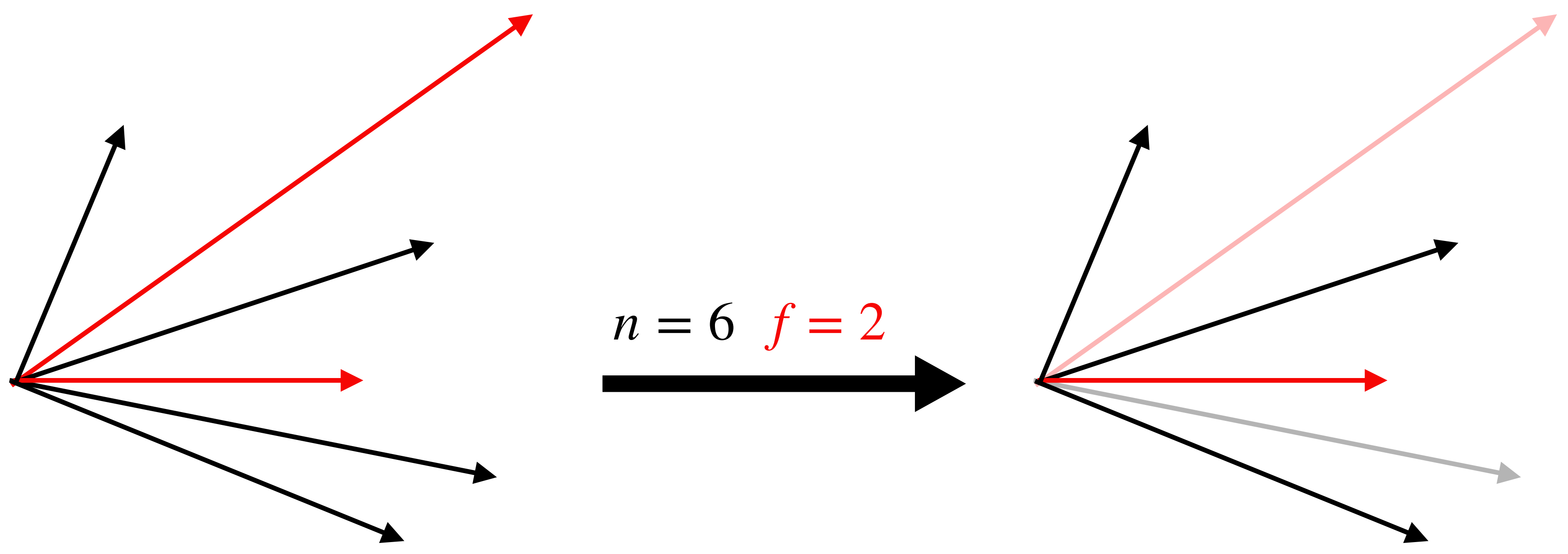
Yin et al., ICML’18; Su & Shahrampour, TRACON’19

Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

Remove gradients with f-largest norms



- **C**oordinate-**w**ise **t**rimmed **m**ean (**CWTM**)

Yin et al., ICML'18; Su & Shahrampour, TRACON'19

For each coord., remove f largest/smallest values

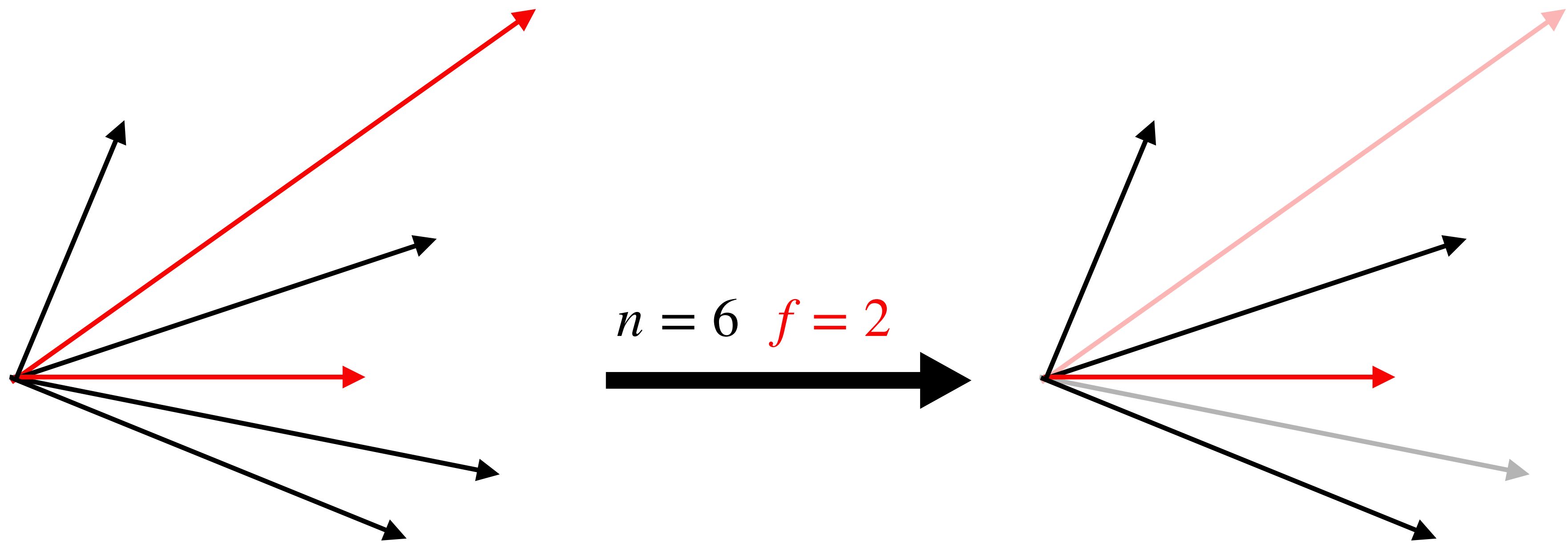
then calculate the mean

Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

Remove gradients with f-largest norms



- **C**oordinate-**w**ise **t**rimmed **m**ean (**CWTM**)

Yin et al., ICML’18; Su & Shahrampour, TRACON’19

-5	-2	1	4	7	9

For each coord., remove f largest/smallest values

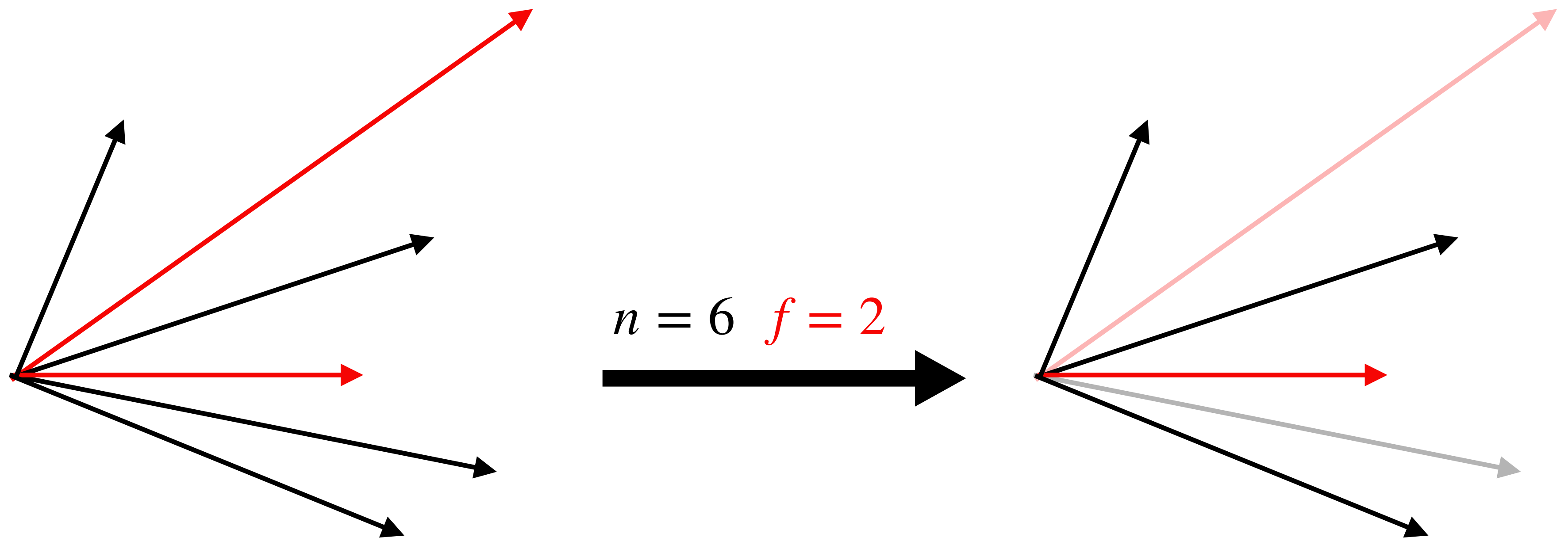
then calculate the mean

Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

Remove gradients with f-largest norms



- **C**oordinate-**w**ise **t**rimmed **m**ean (**CWTM**)

Yin et al., ICML'18; Su & Shahrampour, TRACON'19

For each coord., remove f largest/smallest values

then calculate the mean

-5	-2	1	4	7	9

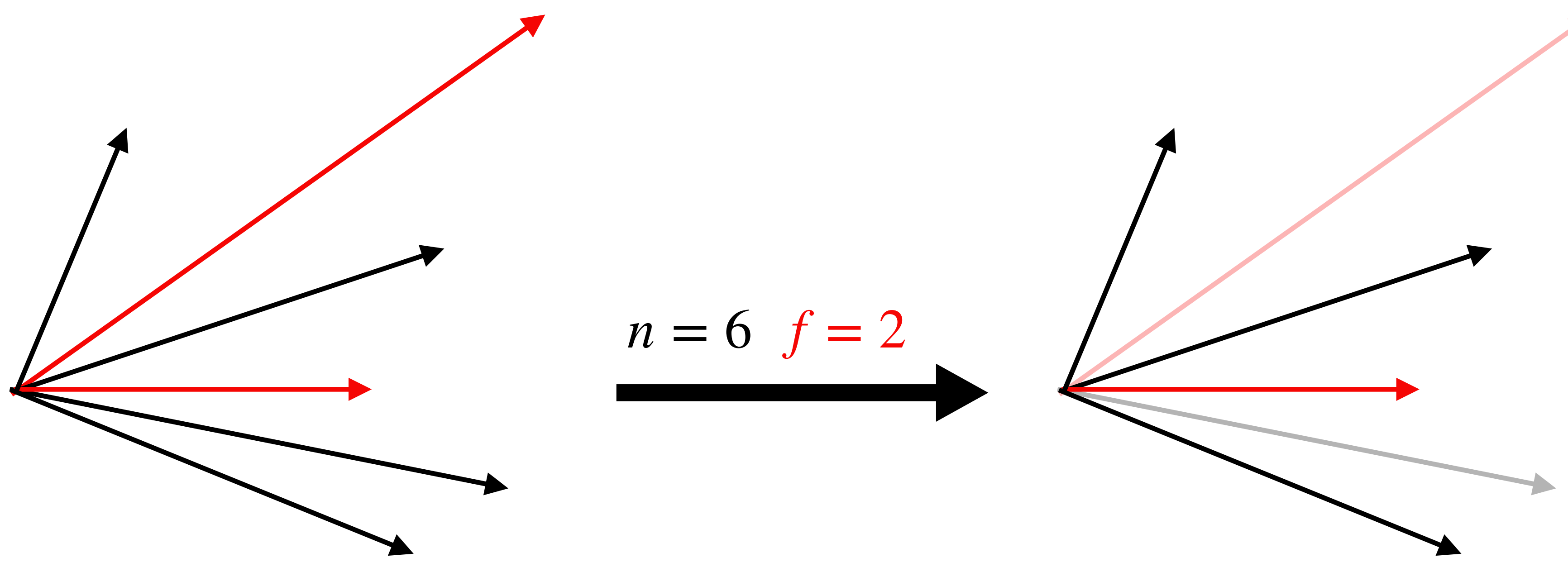
↓ Trim ($n = 6$ $f = 2$)

Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

Remove gradients with f-largest norms

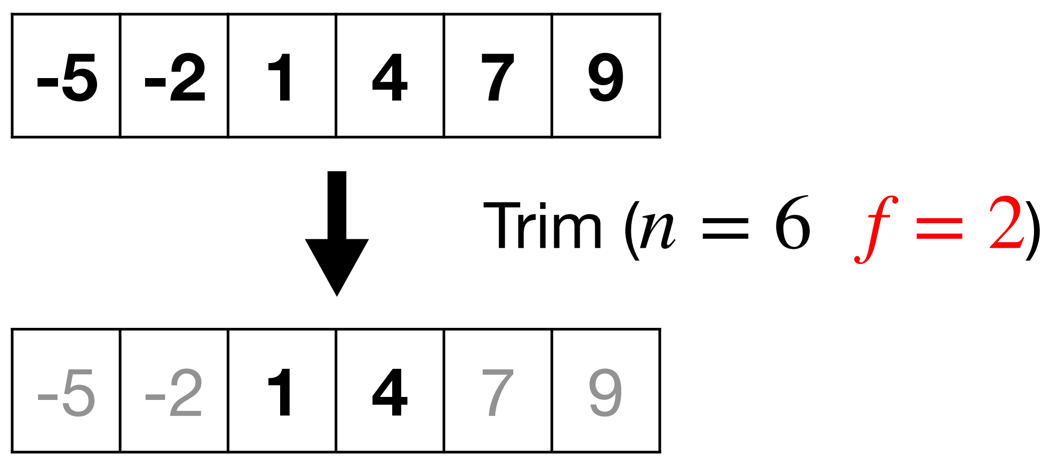


- **C**oordinate-**wise** **t**rimmed **m**ean (**CWTM**)

Yin et al., ICML'18; Su & Shahrampour, TRACON'19

For each coord., remove f largest/smallest values

then calculate the mean

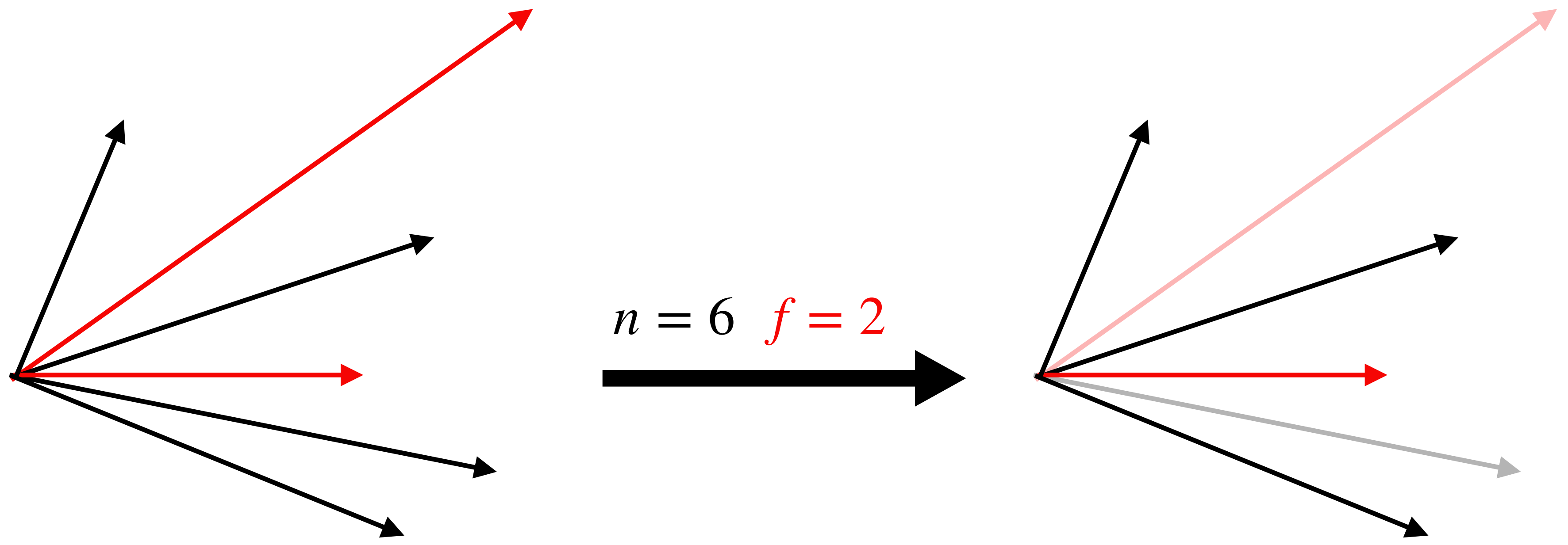


Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

Remove gradients with f-largest norms

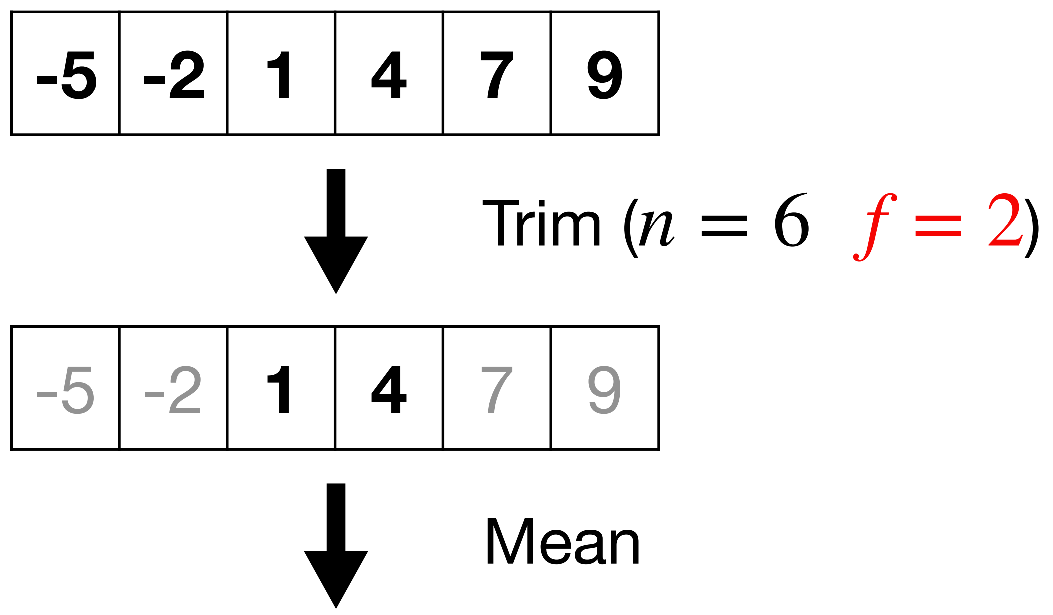


- **C**oordinate-**w**ise **t**rimmed **m**ean (**CWTM**)

Yin et al., ICML'18; Su & Shahrampour, TRACON'19

For each coord., remove f largest/smallest values

then calculate the mean

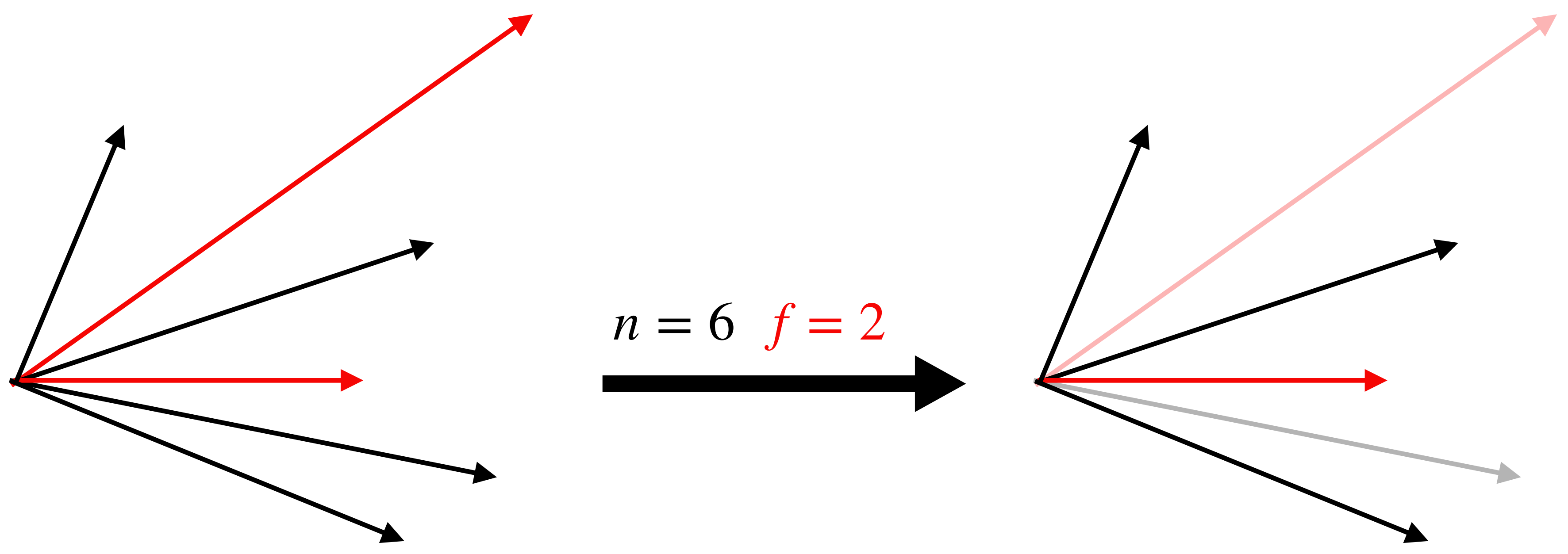


Examples

- **C**omparative **g**radient **e**limination (**CGE**)

Gupta & Vaidya, 2019

Remove gradients with f-largest norms

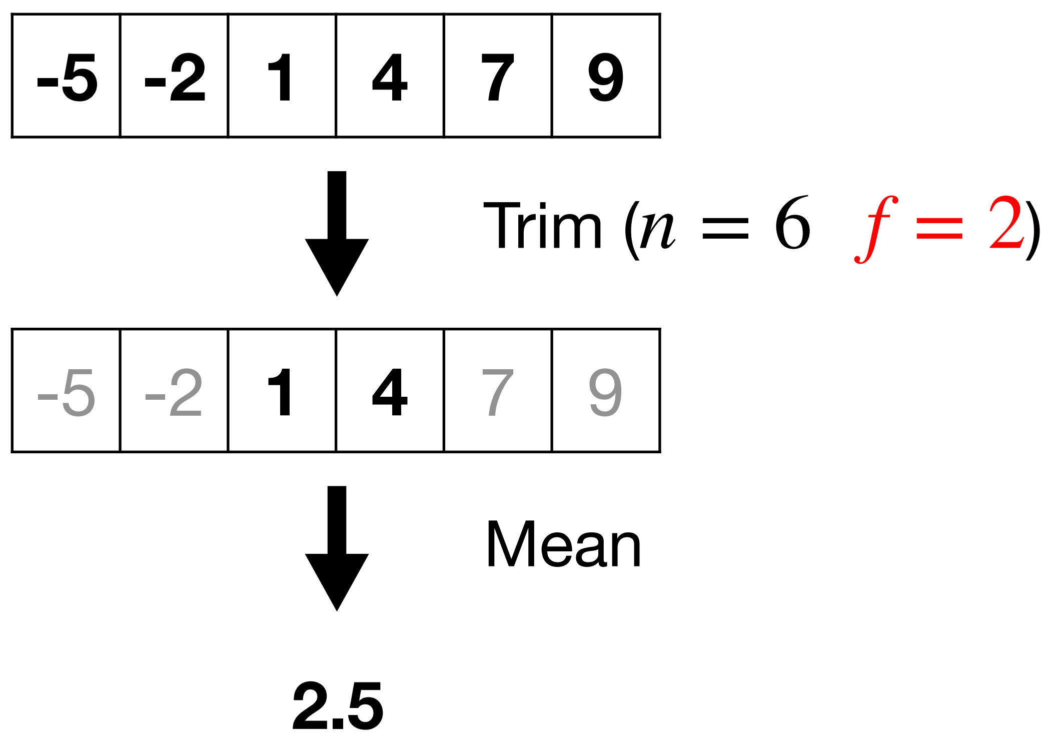


- **C**oordinate-**w**ise **t**rimmed **mean** (**CWTM**)

Yin et al., ICML’18; Su & Shahrampour, TRACON’19

For each coord., remove f largest/smallest values

then calculate the mean



Accuracy under $(2f, \epsilon)$-redundancy

Accuracy under $(2f, \epsilon)$-redundancy

With $(2f, \epsilon)$-redundancy, DGD with gradient filters is $(f, \mathcal{O}(\epsilon))$-resilient

Liu et al., PODC '21

Accuracy under $(2f, \epsilon)$-redundancy

With $(2f, \epsilon)$-redundancy, DGD with gradient filters is $(f, \mathcal{O}(\epsilon))$-resilient

Liu et al., PODC '21

CGE requires $f < n/3$

Resilience of **CWTM** is independent of f *

Accuracy under $(2f, \epsilon)$-redundancy

With $(2f, \epsilon)$-redundancy, DGD with gradient filters is $(f, \mathcal{O}(\epsilon))$-resilient

Liu et al., PODC '21

CGE requires $f < n/3$

Resilience of **CWTM** is independent of f *

* Relies on additional assumption, besides $(2f, \epsilon)$-redundancy

Accuracy under $(2f, \epsilon)$-redundancy

With $(2f, \epsilon)$-redundancy, DGD with gradient filters is $(f, \mathcal{O}(\epsilon))$-resilient

Liu et al., PODC ’21

CGE requires $f < n/3$

Resilience of **CWTM** is independent of f *

Resilience of **CGE** independent of ***d***

Resilience of **CWTM** dependent on ***d***

* Relies on additional assumption, besides $(2f, \epsilon)$-redundancy

Implications on Byzantine fault-tolerant FL

Implications on Byzantine fault-tolerant FL

Distributed ML can be formulated in the same way

Implications on Byzantine fault-tolerant FL

Distributed ML can be formulated in the same way

Each agent has a data distribution $\mathscr{D}_i$

Implications on Byzantine fault-tolerant FL

Distributed ML can be formulated in the same way

Each agent has a data distribution $\mathcal{D}_i$

Loss function for each data point z: $\ell(x; z) : \mathbb{R}^d \mapsto \mathbb{R}$

Implications on Byzantine fault-tolerant FL

Distributed ML can be formulated in the same way

Each agent has a data distribution $\mathscr{D}_i$

Loss function for each data point z: $\ell(x; z) : \mathbb{R}^d \mapsto \mathbb{R}$

$$\boxed{Q_i(x) = \mathbb{E}_{z^t \in \mathscr{D}_i} \ell(x; z_i^t)}$$

Implications on Byzantine fault-tolerant FL

Distributed ML can be formulated in the same way

Each agent has a data distribution $\mathscr{D}_i$

Loss function for each data point z: $\ell(x; z) : \mathbb{R}^d \mapsto \mathbb{R}$

$$\boxed{Q_i(x) = \mathbb{E}_{z^t \in \mathscr{D}_i} \ell(x; z_i^t)}$$

SGD instead of GD

Implications on Byzantine fault-tolerant FL

Distributed ML can be formulated in the same way

Each agent has a data distribution $\mathscr{D}_i$

Loss function for each data point z: $\ell(x; z) : \mathbb{R}^d \mapsto \mathbb{R}$

$$\boxed{Q_i(x) = \mathbb{E}_{z^t \in \mathscr{D}_i} \ell(x; z_i^t)}$$

SGD instead of GD

$$\boxed{g_i^t = \frac{1}{k} \sum_{z_i^t \in z^t} \nabla \ell(x^t; z_i^t)}$$

Implications on Byzantine fault-tolerant FL

Distributed ML can be formulated in the same way

Each agent has a data distribution $\mathscr{D}_i$

Loss function for each data point z: $\ell(x; z) : \mathbb{R}^d \mapsto \mathbb{R}$

$$\boxed{Q_i(x) = \mathbb{E}_{z^t \in \mathscr{D}_i} \ell(x; z_i^t)}$$

SGD instead of GD

$$\boxed{g_i^t = \frac{1}{k} \sum_{z_i^t \in z^t} \nabla \ell(x^t; z_i^t)}$$

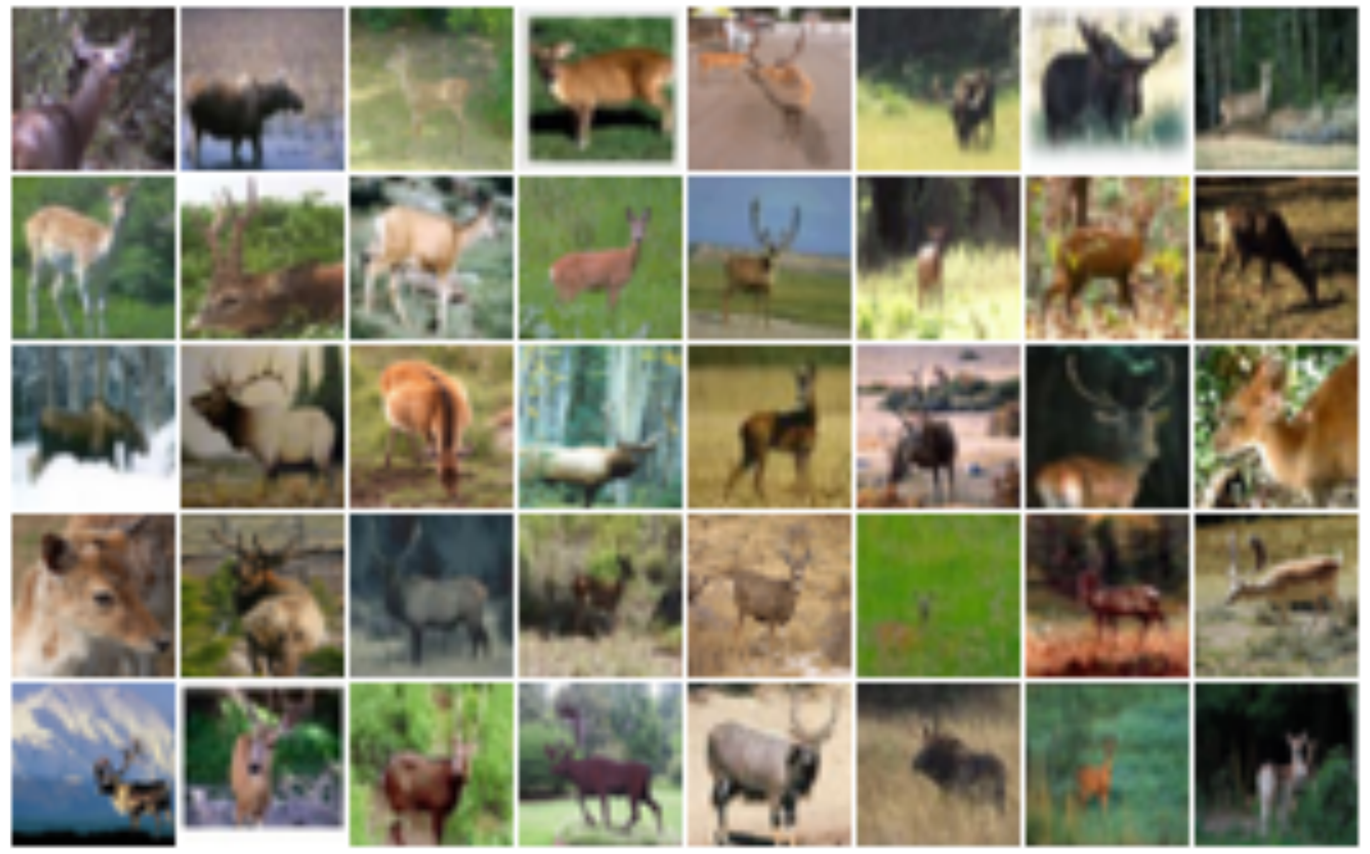
$$\mathbb{E}\left[g_i^t\right] = \nabla Q_i(x^t)$$

Example

Example

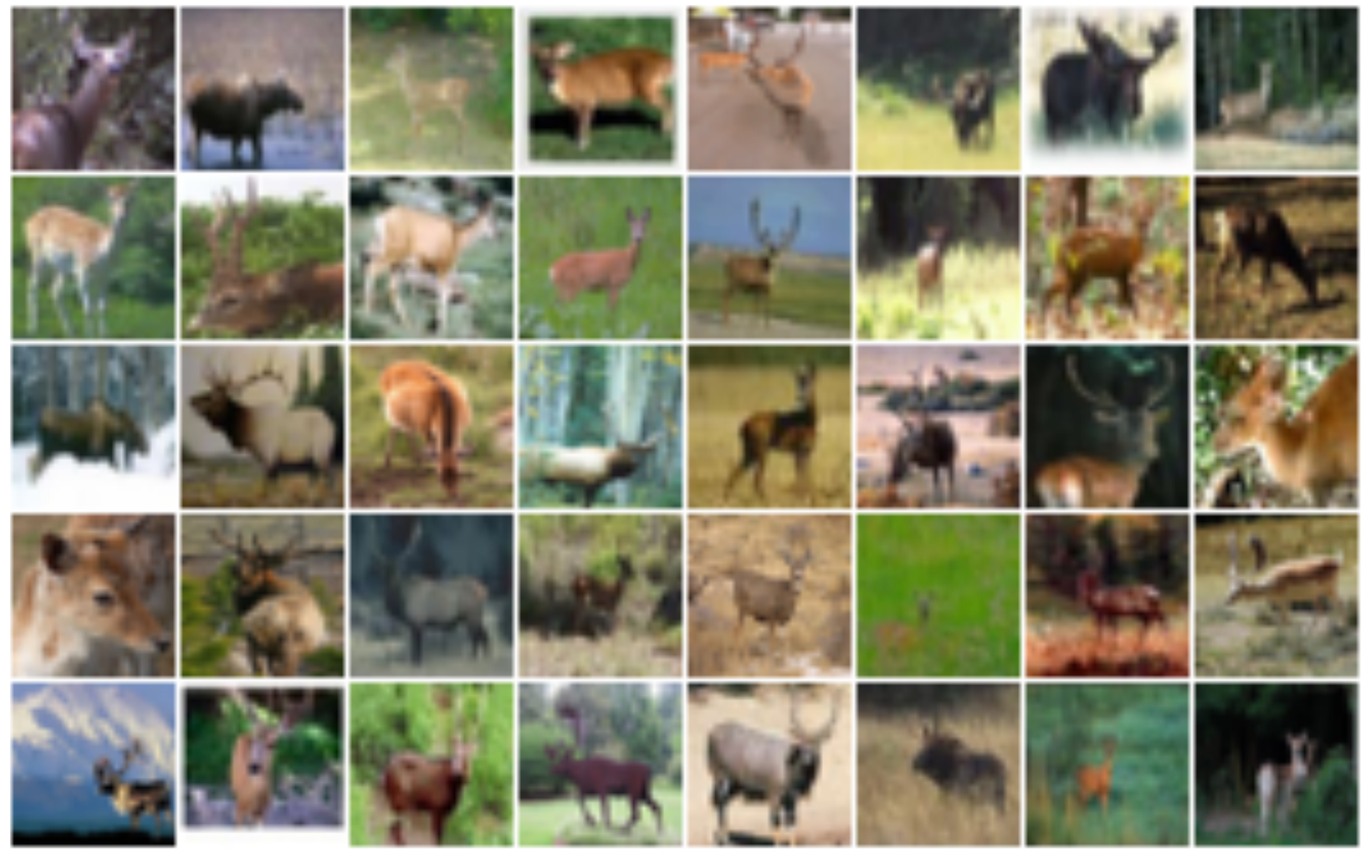
Distributed Image Classifiers

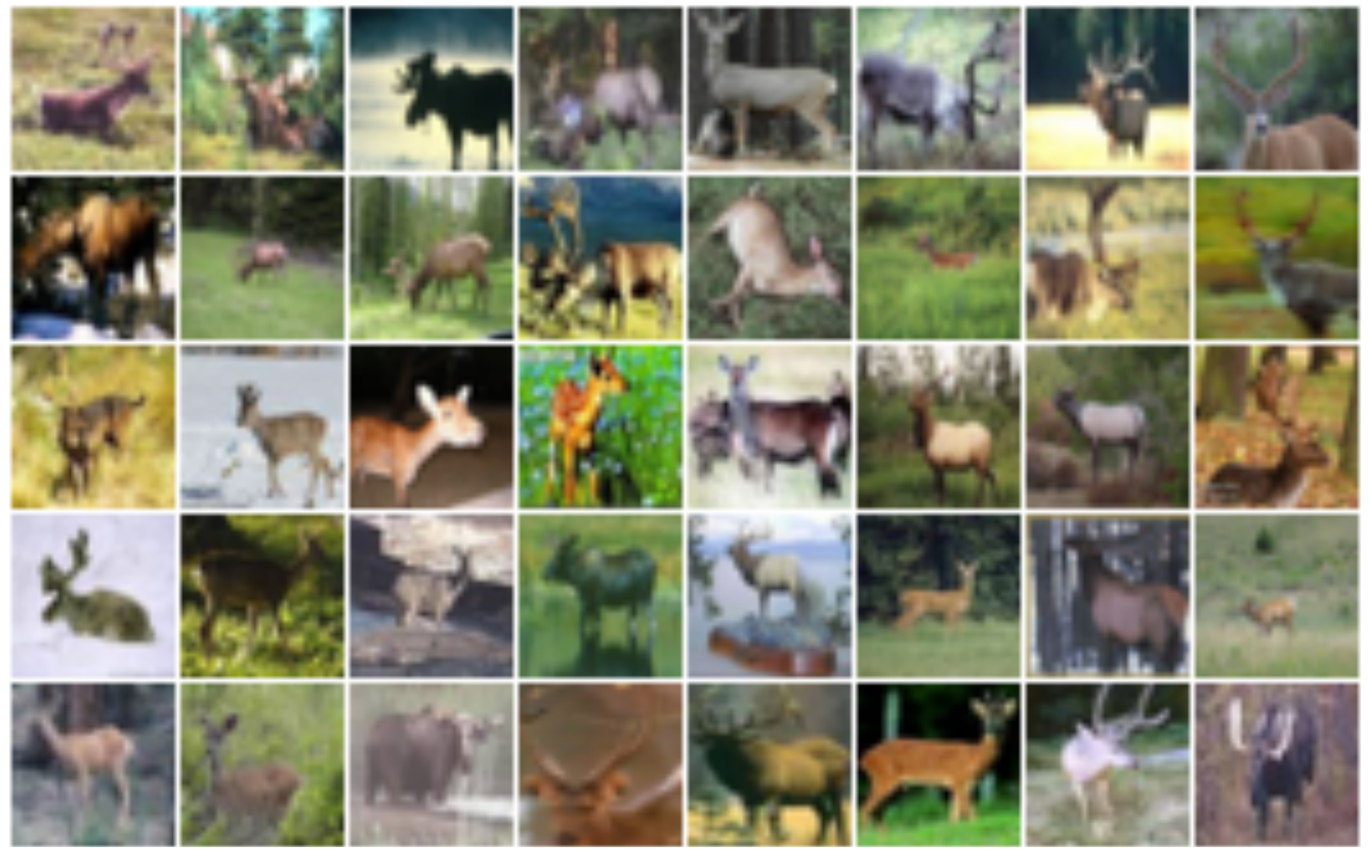
Example



Distributed Image Classifiers

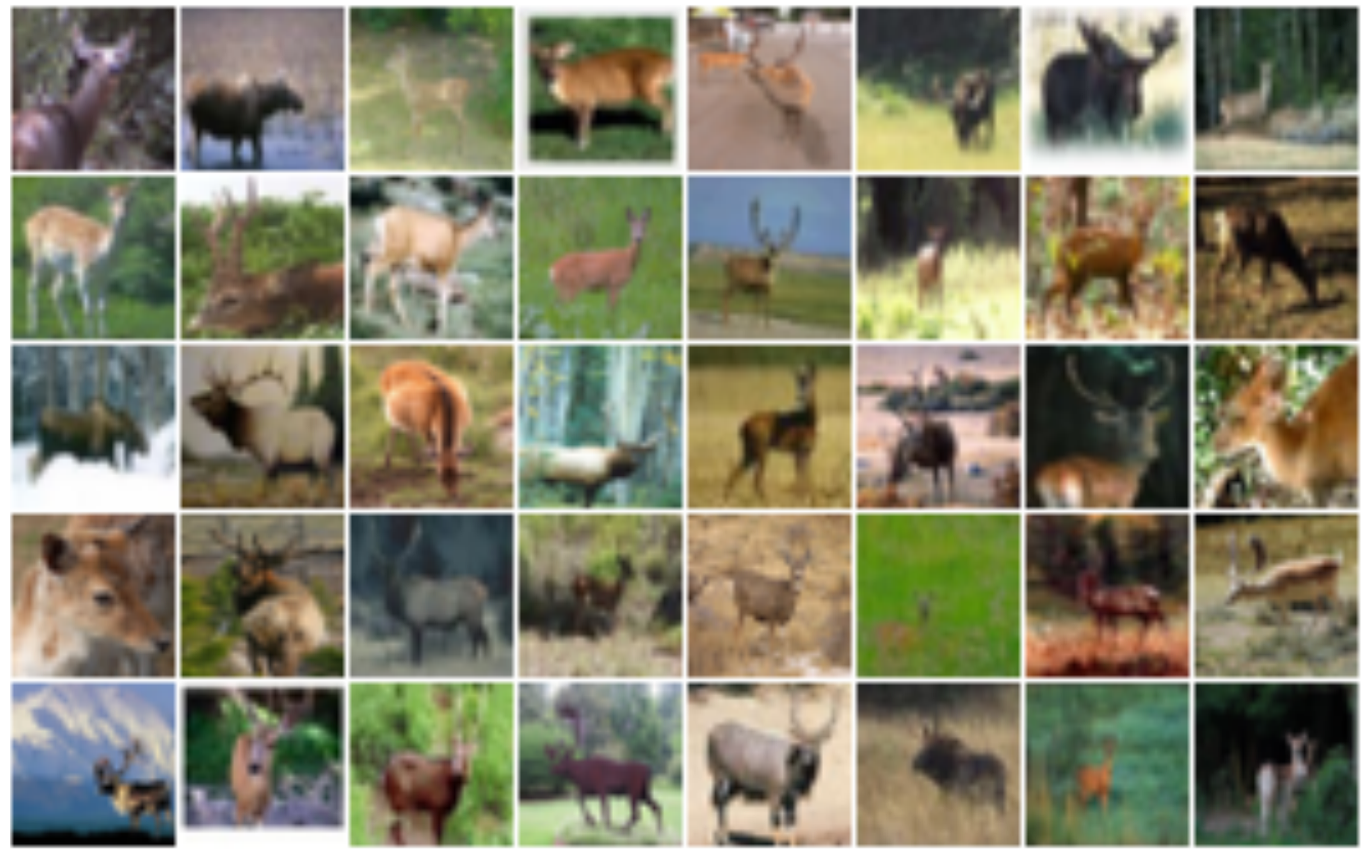
Example

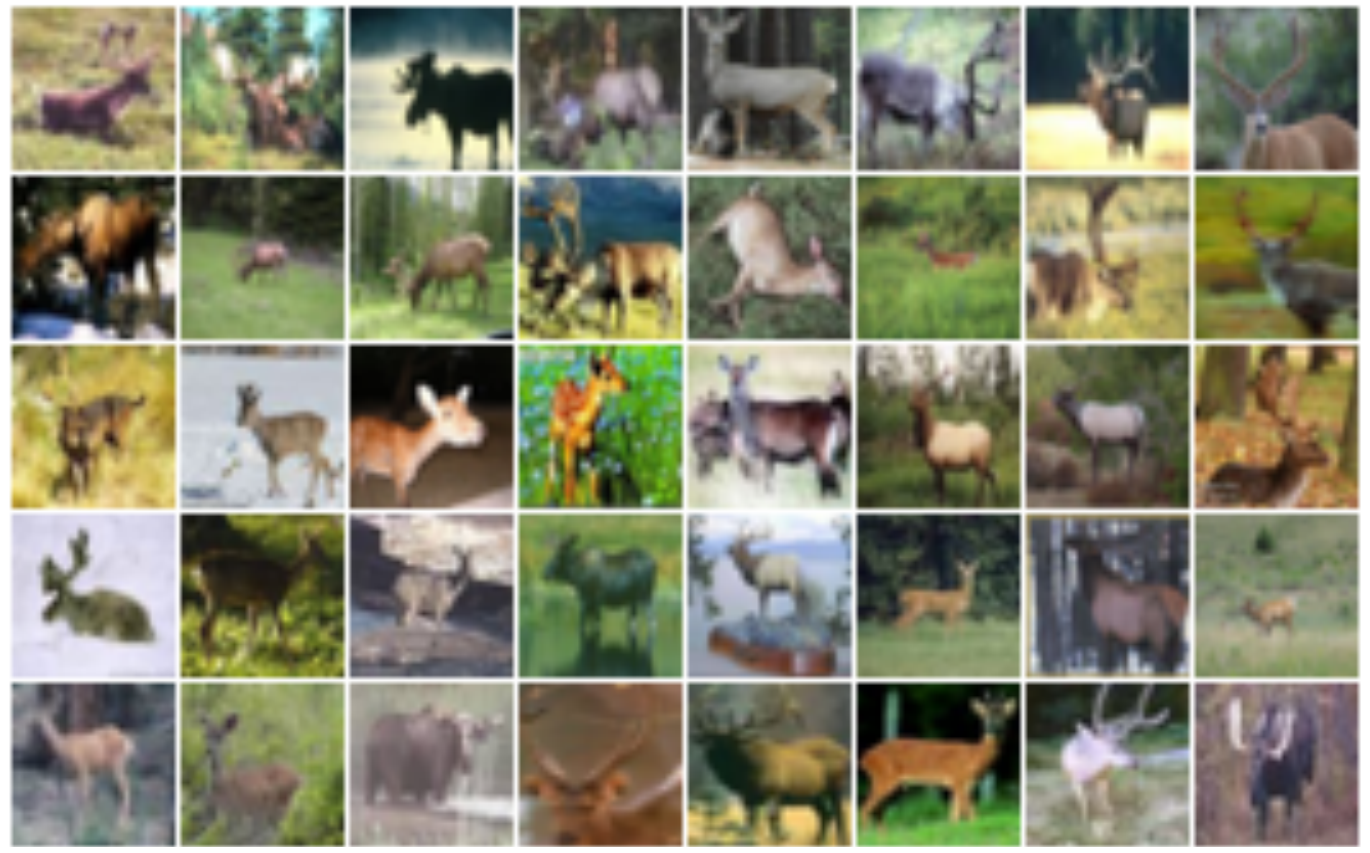




Distributed Image Classifiers

Example





Distributed Image Classifiers

Example

Distributed Image Classifiers

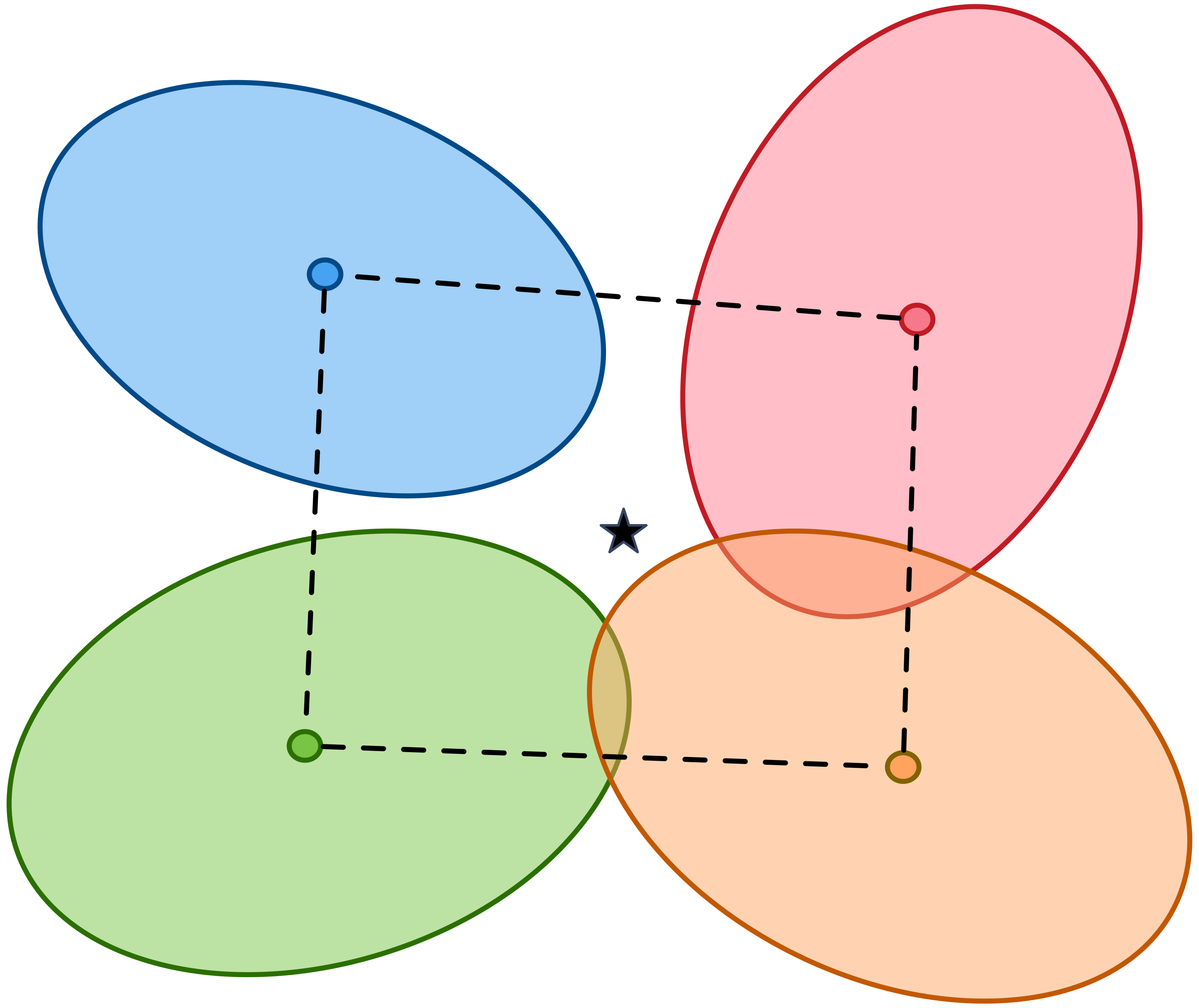
Example

They are all deers.*

Distributed Image Classifiers

* CIFAR -10

Example

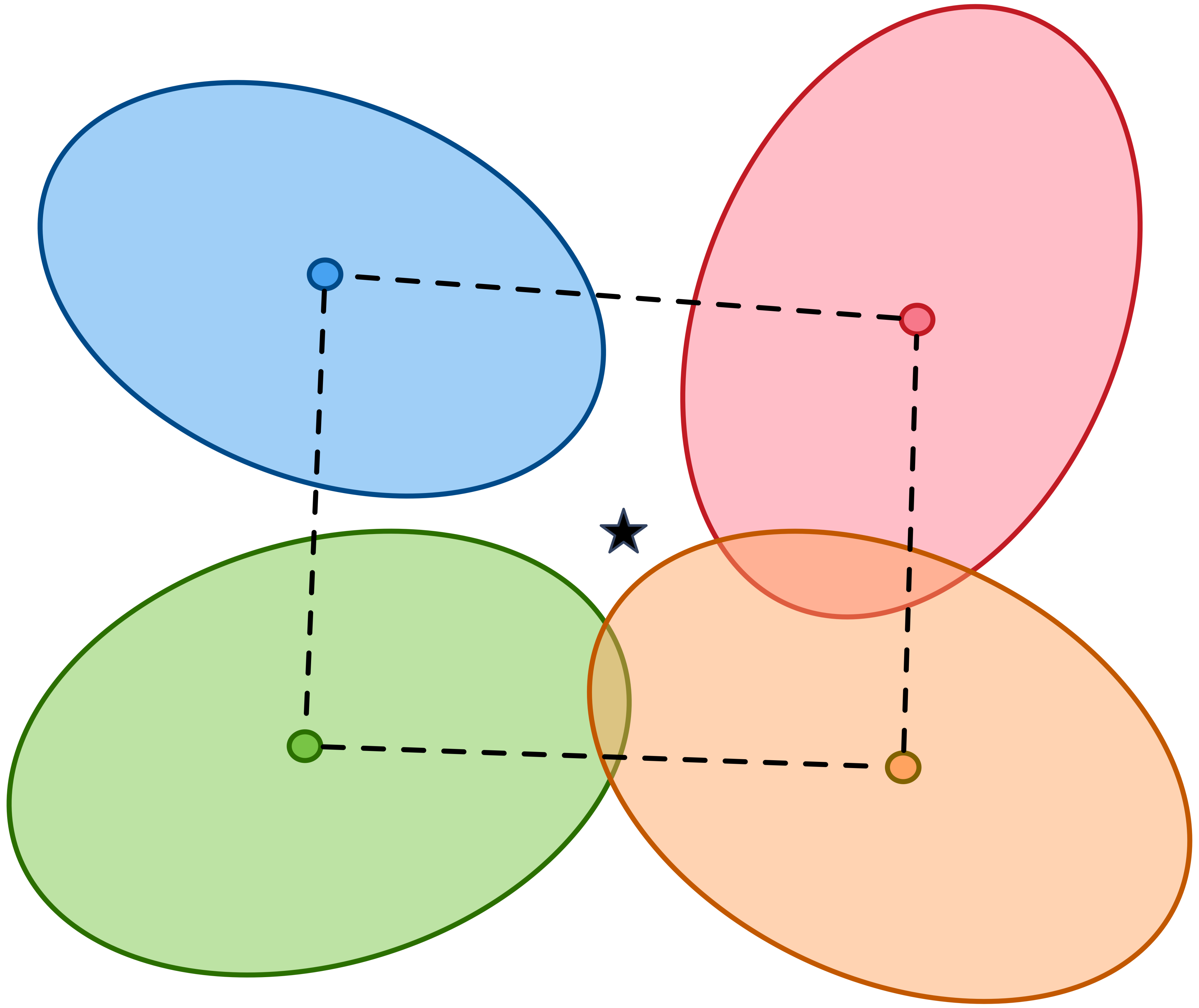


Distributed Image Classifiers

They are all deers.*

* CIFAR -10

Example



Distributed Image Classifiers

Data diversity <-> Redundancy

They are all deers.*

* CIFAR -10

Implications on Byzantine fault-tolerant FL

$(2f, \epsilon)$-redundancy always exists

σ is a bound over variance of stochastic gradients

Implications on Byzantine fault-tolerant FL

$(2f, \epsilon)$-redundancy always exists

Recall that redundancy *describes* the cost functions

σ is a bound over variance of stochastic gradients

Implications on Byzantine fault-tolerant FL

$(2f, \epsilon)$-redundancy always exists

Redundancy *describes* the **datasets**

σ is a bound over variance of stochastic gradients

Implications on Byzantine fault-tolerant FL

$(2f, \epsilon)$-redundancy always exists

Redundancy *describes* the **datasets**

With $(2f, \epsilon)$-redundancy
D-SGD with CGE is $(f, \mathcal{O}(\epsilon) + \mathcal{O}(\sigma))$-resilient

Liu et al., arXiv '21

σ **is a bound over variance of stochastic gradients**

Implications on Resilient FL

From $(2f, \epsilon)$-redundancy to $(f, r; \epsilon)$-redundancy

σ is a bound over variance of stochastic gradients

Implications on Resilient FL

From $(2f, \epsilon)$-redundancy to $(f, r; \epsilon)$-redundancy

We can further involve asynchrony (resilient against stragglers)

σ is a bound over variance of stochastic gradients

Implications on Resilient FL

From $(2f, \epsilon)$-redundancy to $(f, r; \epsilon)$-redundancy

We can further involve asynchrony (resilient against stragglers)

Up to f Byzantine agents and r stragglers

σ is a bound over variance of stochastic gradients

Implications on Resilient FL

From $(2f, \epsilon)$-redundancy to $(f, r; \epsilon)$-redundancy

We can further involve asynchrony (resilient against stragglers)

Up to f Byzantine agents and r stragglers

$\left(f, r; \epsilon\right)$**-redundancy**: Subsets $S, \widehat{S} \subseteq \{1,\ldots,n\}$ with $|S| = n - f$, $|\widehat{S}| \geq n - 2f - r$, and $\widehat{S} \subseteq S$,

$$\mathrm{dist}\left(\arg\min_{x \in \mathbb{R}^d} \sum_{i \in S} Q_i(x),\ \ \arg\min_{x \in \mathbb{R}^d} \sum_{i \in \widehat{S}} Q_i(x)\right) \leq \epsilon$$

σ is a bound over variance of stochastic gradients

Implications on Resilient FL

From $(2f, \epsilon)$-redundancy to $(f, r; \epsilon)$-redundancy

We can further involve asynchrony (resilient against stragglers)

Up to f Byzantine agents and r stragglers

σ is a bound over variance of stochastic gradients

Implications on Resilient FL

From $(2f, \epsilon)$-redundancy to $(f, r; \epsilon)$-redundancy

We can further involve asynchrony (resilient against stragglers)

Up to f Byzantine agents and r stragglers

With $(f, r; \epsilon)$-redundancy
D-SGD with CGE is $(f, \mathcal{O}(\epsilon) + \mathcal{O}(\sigma))$-resilient

Liu et al., arXiv '21

σ is a bound over variance of stochastic gradients

Implications on Resilient FL

From $(2f, \epsilon)$-redundancy to $(f, r; \epsilon)$-redundancy

We can further involve asynchrony (resilient against stragglers)

Up to f Byzantine agents and r stragglers

With $(f, r; \epsilon)$-redundancy
D-SGD with CGE is $(f, \mathcal{O}(\epsilon) + \mathcal{O}(\sigma))$-resilient

Liu et al., arXiv '21

D-SGD with gradient filter can tolerate Byzantine agents and stragglers at the same time given redundancy

σ is a bound over variance of stochastic gradients

Details of Presented Results

* Some of the slides come from our PODC’21 presentation

Details of Presented Results

Liu, Shuo, Nirupam Gupta, and Nitin H. Vaidya. **"Approximate Byzantine Fault-Tolerance in Distributed Optimization."** *arXiv preprint arXiv:2101.09337* (2021).

*** Some of the slides come from our PODC'21 presentation**

Details of Presented Results

Liu, Shuo, Nirupam Gupta, and Nitin H. Vaidya. "**Approximate Byzantine Fault-Tolerance in Distributed Optimization.**" *arXiv preprint arXiv:2101.09337* (2021).

Gupta, Nirupam, Shuo Liu, and Nitin H. Vaidya. "**Byzantine Fault-Tolerant Distributed Machine Learning Using Stochastic Gradient Descent (SGD) and Norm-Based Comparative Gradient Elimination (CGE).**" *arXiv preprint arXiv:2008.04699* (2020).

*** Some of the slides come from our PODC'21 presentation**

Details of Presented Results

Liu, Shuo, Nirupam Gupta, and Nitin H. Vaidya. "**Approximate Byzantine Fault-Tolerance in Distributed Optimization.**" *arXiv preprint arXiv:2101.09337* (2021).

Gupta, Nirupam, Shuo Liu, and Nitin H. Vaidya. "**Byzantine Fault-Tolerant Distributed Machine Learning Using Stochastic Gradient Descent (SGD) and Norm-Based Comparative Gradient Elimination (CGE).**" *arXiv preprint arXiv:2008.04699* (2020).

Liu, Shuo, Nirupam Gupta, and Nitin H. Vaidya. "**Utilizing Redundancy in Cost Functions for Resilience in Distributed Optimization and Learning.**" *arXiv preprint arXiv:2110.10858* (2021).

*** Some of the slides come from our PODC'21 presentation**

Details of Presented Results

Liu, Shuo, Nirupam Gupta, and Nitin H. Vaidya. **"Approximate Byzantine Fault-Tolerance in Distributed Optimization."** *arXiv preprint arXiv:2101.09337* (2021).

Gupta, Nirupam, Shuo Liu, and Nitin H. Vaidya. **"Byzantine Fault-Tolerant Distributed Machine Learning Using Stochastic Gradient Descent (SGD) and Norm-Based Comparative Gradient Elimination (CGE)."** *arXiv preprint arXiv:2008.04699* (2020).

Liu, Shuo, Nirupam Gupta, and Nitin H. Vaidya. **"Utilizing Redundancy in Cost Functions for Resilience in Distributed Optimization and Learning."** *arXiv preprint arXiv:2110.10858* (2021).

See also

* Some of the slides come from our PODC'21 presentation

Details of Presented Results

Liu, Shuo, Nirupam Gupta, and Nitin H. Vaidya. **"Approximate Byzantine Fault-Tolerance in Distributed Optimization."** *arXiv preprint arXiv:2101.09337* (2021).

Gupta, Nirupam, Shuo Liu, and Nitin H. Vaidya. **"Byzantine Fault-Tolerant Distributed Machine Learning Using Stochastic Gradient Descent (SGD) and Norm-Based Comparative Gradient Elimination (CGE)."** *arXiv preprint arXiv:2008.04699* (2020).

Liu, Shuo, Nirupam Gupta, and Nitin H. Vaidya. **"Utilizing Redundancy in Cost Functions for Resilience in Distributed Optimization and Learning."** *arXiv preprint arXiv:2110.10858* (2021).

See also

Gupta, Nirupam, and Nitin H. Vaidya. **"Fault-tolerance in distributed optimization: The case of redundancy."** *Proceedings of the 39th Symposium on Principles of Distributed Computing*. 2020.

* Some of the slides come from our PODC'21 presentation

Details of Presented Results

Liu, Shuo, Nirupam Gupta, and Nitin H. Vaidya. "**Approximate Byzantine Fault-Tolerance in Distributed Optimization.**" *arXiv preprint arXiv:2101.09337* (2021).

Gupta, Nirupam, Shuo Liu, and Nitin H. Vaidya. "**Byzantine Fault-Tolerant Distributed Machine Learning Using Stochastic Gradient Descent (SGD) and Norm-Based Comparative Gradient Elimination (CGE).**" *arXiv preprint arXiv:2008.04699* (2020).

Liu, Shuo, Nirupam Gupta, and Nitin H. Vaidya. "**Utilizing Redundancy in Cost Functions for Resilience in Distributed Optimization and Learning.**" *arXiv preprint arXiv:2110.10858* (2021).

See also

Gupta, Nirupam, and Nitin H. Vaidya. "**Fault-tolerance in distributed optimization: The case of redundancy.**" *Proceedings of the 39th Symposium on Principles of Distributed Computing*. 2020.

Gupta, Nirupam, and Nitin H. Vaidya. "**Resilience in collaborative optimization: redundant and independent cost functions.**" *arXiv preprint arXiv:2003.09675* (2020).

* Some of the slides come from our PODC'21 presentation

Acknowledgements

Research reported in this paper was supported in part by the **Army Research Laboratory** under Cooperative Agreement W911NF- 17-2-0196, and by the **National Science Foundation** award 1842198.* Research reported in this paper is also supported in part by a **Fritz Fellowship** from Georgetown University.

* The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, National Science Foundation or the U.S. Government.

Thank You!

Shuo Liu: sl1539@georgetown.edu

Nirupam Gupta: nirupam.gupta@epfl.ch

Nitin Vaidya: nitin.vaidya@georgetown.edu