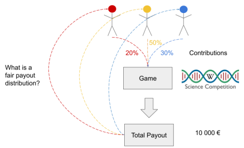
Interpretable Machine Learning

Shapley Values



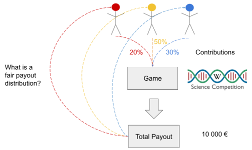
Learning goals

- Learn cooperative games and value functions
- Define the marginal contribution of a player
- Study Shapley value as a fair payout solution
- Compare order and set definitions

Interpretable Machine Learning

Shapley

Shapley Values

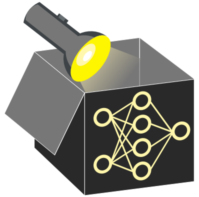


Learning goals

- Learn cooperative games and value functions
- Define the marginal contribution of a player
- Study Shapley value as a fair payout solution
- Compare order and set definitions

COOPERATIVE GAMES IN GAME THEORY

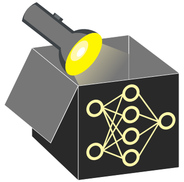
▸ Shapley (1951)



- **Game theory:** Studies strategic interactions among "players" (who act to maximize their utility), where outcomes depend on collective behavior
- **Cooperative games:** Any subset $S \subseteq P = \{1, \ldots, p\}$ can form a coalition to cooperate in a game, each achieving a payout $v(S)$

COOPERATIVE GAMES IN GAME THEORY

▸ SHAPLEY_1951

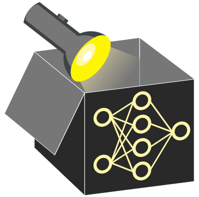


- **Game theory:** Studies strategic interactions among "players" (who act to maximize their utility), where outcomes depend on collective behavior
- **Cooperative games:** Any subset $S \subseteq P = \{1, \ldots, p\}$ can form a coalition to cooperate in a game, each achieving a payout $v(S)$

©

COOPERATIVE GAMES IN GAME THEORY

▸ Shapley (1951)

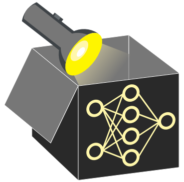


- **Game theory:** Studies strategic interactions among "players" (who act to maximize their utility), where outcomes depend on collective behavior
- **Cooperative games:** Any subset $S \subseteq P = \{1, \ldots, p\}$ can form a coalition to cooperate in a game, each achieving a payout $v(S)$
- **Value function:** $v : 2^P \to \mathbb{R}$ assigns each coalition S a payout $v(S)$
  - Convention: $v(\emptyset) = 0 \rightsquigarrow$ Empty coalitions generate no gain
  - $v(P)$: Total achievable payout when all players cooperate
    $\rightsquigarrow$ Forms the game's budget to be fairly distributed
- **Marginal contribution:** Measure how much value player j adds to coalition S by

$$\Delta(j, S) := v(S \cup \{j\}) - v(S) \qquad \text{(for all } j \in P\ S \subseteq P \setminus \{j\})$$

COOPERATIVE GAMES IN GAME THEORY

▸ SHAPLEY_1951



- **Game theory:** Studies strategic interactions among "players" (who act to maximize their utility), where outcomes depend on collective behavior
- **Cooperative games:** Any subset $S \subseteq P = \{1, \ldots, p\}$ can form a coalition to cooperate in a game, each achieving a payout $v(S)$
- **Value function:** $v : 2^P \to \mathbb{R}$ assigns each coalition S a payout $v(S)$
  - Convention: $v(\emptyset) = 0 \rightsquigarrow$ Empty coalitions generate no gain
  - $v(P)$: Total achievable payout when all players cooperate
    $\rightsquigarrow$ Forms the game's budget to be fairly distributed
- **Marginal contribution:** Measure how much value player j adds to coalition S by

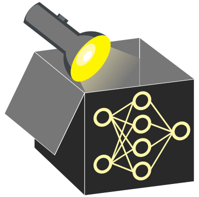
$$\Delta(j, S) := v(S \cup \{j\}) - v(S) \qquad \text{(for all } j \in P\ S \subseteq P \setminus \{j\})$$

©

Interpretable Machine Learning – 1 / 11

COOPERATIVE GAMES IN GAME THEORY

▸ Shapley (1951)



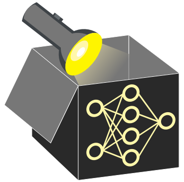
- **Game theory:** Studies strategic interactions among "players" (who act to maximize their utility), where outcomes depend on collective behavior
- **Cooperative games:** Any subset $S \subseteq P = \{1, \ldots, p\}$ can form a coalition to cooperate in a game, each achieving a payout $v(S)$
- **Value function:** $v : 2^P \to \mathbb{R}$ assigns each coalition S a payout $v(S)$
  - Convention: $v(\emptyset) = 0 \rightsquigarrow$ Empty coalitions generate no gain
  - $v(P)$: Total achievable payout when all players cooperate
    $\rightsquigarrow$ Forms the game's budget to be fairly distributed
- **Marginal contribution:** Measure how much value player j adds to coalition S by

$$\Delta(j, S) := v(S \cup \{j\}) - v(S) \qquad \text{(for all } j \in P \; S \subseteq P \setminus \{j\})$$

- **Challenge:** Players vary in their contributions & how they influence each other
- **Goal:** Fairly distribute $v(P)$ among players by accounting for player interactions
  $\rightsquigarrow$ Assign each player $j \in P$ a fair share ϕ_j (**Shapley value**)

COOPERATIVE GAMES IN GAME THEORY

▸ SHAPLEY_1951

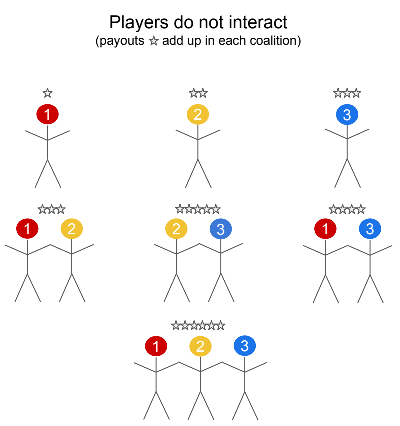


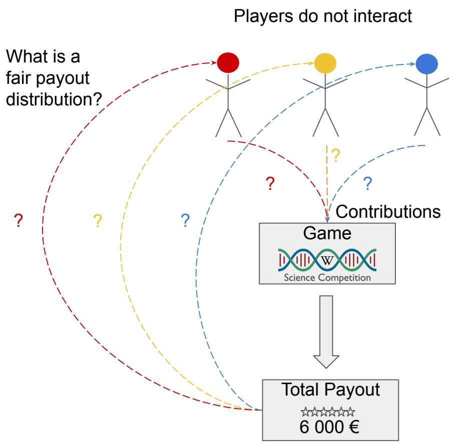
- **Game theory:** Studies strategic interactions among "players" (who act to maximize their utility), where outcomes depend on collective behavior
- **Cooperative games:** Any subset $S \subseteq P = \{1, \ldots, p\}$ can form a coalition to cooperate in a game, each achieving a payout $v(S)$
- **Value function:** $v : 2^P \to \mathbb{R}$ assigns each coalition S a payout $v(S)$
  - Convention: $v(\emptyset) = 0 \rightsquigarrow$ Empty coalitions generate no gain
  - $v(P)$: Total achievable payout when all players cooperate
    $\rightsquigarrow$ Forms the game's budget to be fairly distributed
- **Marginal contribution:** Measure how much value player j adds to coalition S by

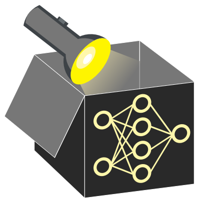
$$\Delta(j, S) := v(S \cup \{j\}) - v(S) \qquad \text{(for all } j \in P \; S \subseteq P \setminus \{j\})$$

- **Challenge:** Players vary in their contrib. & how they influence each other
- **Goal:** Distribute $v(P)$ among players by considering player interactions
  $\rightsquigarrow$ Assign each player $j \in P$ a fair share ϕ_j (**Shapley value**)

COOPERATIVE GAMES - NO INTERACTIONS

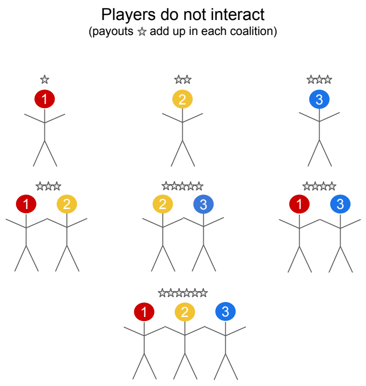


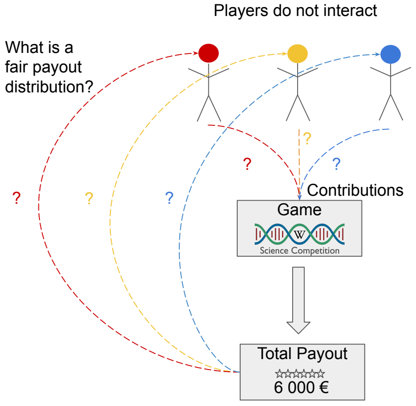


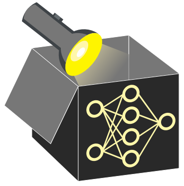


Question: What are the individual marginal contributions and what is a fair payout?

COOPERATIVE GAMES - NO INTERACTIONS





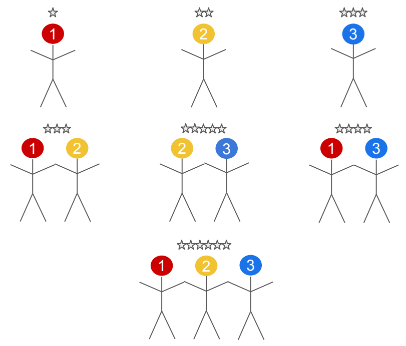


Question: What are individual marginal contributions and what's a fair payout?

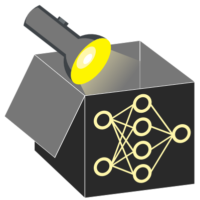
©

COOPERATIVE GAMES - NO INTERACTIONS

Players do not interact
(payouts ☆ add up in each coalition)

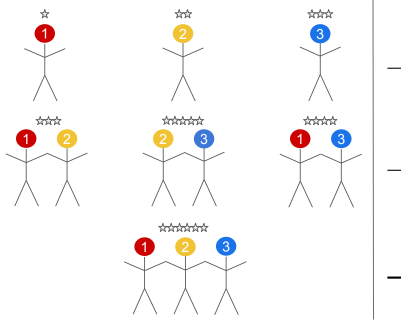


Player	Coalition S	$v(S \cup \{j\})$	$v(S)$	$\Delta(j, S)$
1	$\emptyset$	1000	0	1000
1	$\{2\}$	3000	2000	1000
1	$\{3\}$	4000	3000	1000
1	$\{2, 3\}$	6000	5000	1000
2	$\emptyset$	2000	0	2000
2	$\{1\}$	3000	1000	2000
2	$\{3\}$	5000	3000	2000
2	$\{1, 3\}$	6000	4000	2000
3	$\emptyset$	3000	0	3000
3	$\{1\}$	4000	1000	3000
3	$\{2\}$	5000	2000	3000
3	$\{1, 2\}$	6000	3000	3000

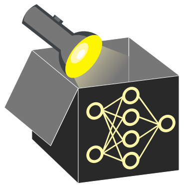


COOPERATIVE GAMES - NO INTERACTIONS

Players do not interact
(payouts ☆ add up in each coalition)



Player	Coalition S	$v(S \cup \{j\})$	$v(S)$	$\Delta(j, S)$
1	$\emptyset$	1000	0	1000
1	$\{2\}$	3000	2000	1000
1	$\{3\}$	4000	3000	1000
1	$\{2, 3\}$	6000	5000	1000
2	$\emptyset$	2000	0	2000
2	$\{1\}$	3000	1000	2000
2	$\{3\}$	5000	3000	2000
2	$\{1, 3\}$	6000	4000	2000
3	$\emptyset$	3000	0	3000
3	$\{1\}$	4000	1000	3000
3	$\{2\}$	5000	2000	3000
3	$\{1, 2\}$	6000	3000	3000

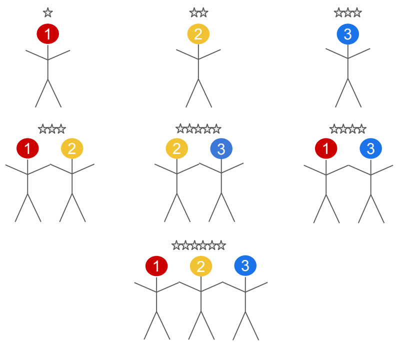


©

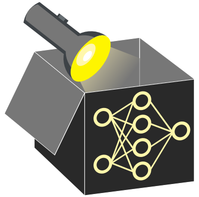
Interpretable Machine Learning – 2 / 11

COOPERATIVE GAMES - NO INTERACTIONS

Players do not interact
(payouts ☆ add up in each coalition)



Player	Coalition S	$v(S \cup \{j\})$	$v(S)$	$\Delta(j, S)$
❶	$\emptyset$	1000	0	1000
❶	{❷}	3000	2000	1000
❶	{❸}	4000	3000	1000
❶	{❷, ❸}	6000	5000	1000
❷	$\emptyset$	2000	0	2000
❷	{❶}	3000	1000	2000
❷	{❸}	5000	3000	2000
❷	{❶, ❸}	6000	4000	2000
❸	$\emptyset$	3000	0	3000
❸	{❶}	4000	1000	3000
❸	{❷}	5000	2000	3000
❸	{❶, ❷}	6000	3000	3000

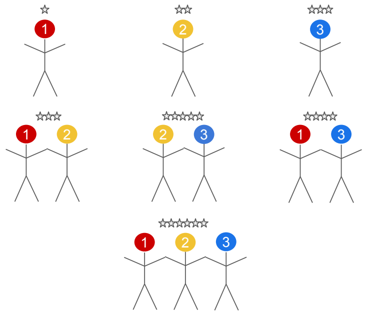


- **No interactions:** Each player contributes the same fixed value to each coalition
  - ⇝ Player ❶ always adds 1000, ❷ adds 2000, and ❸ adds 3000
  - ⇝ Marginal contributions are constant across all coalitions S
- **Conclusion:** Fair payout $=$ average marginal contribution across all S
  - ⇝ Total value $v(P) = 6000$ splits proportionally by individual contributions:

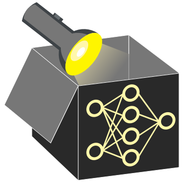
$$❶ = \tfrac{1}{6}, \quad ❷ = \tfrac{1}{3}, \quad ❸ = \tfrac{1}{2}$$

COOPERATIVE GAMES - NO INTERACTIONS

Players do not interact
(payouts ☆ add up in each coalition)



Player	Coalition S	$v(S \cup \{j\})$	$v(S)$	$\Delta(j, S)$
❶	$\emptyset$	1000	0	1000
❶	{❷}	3000	2000	1000
❶	{❸}	4000	3000	1000
❶	{❷, ❸}	6000	5000	1000
❷	$\emptyset$	2000	0	2000
❷	{❶}	3000	1000	2000
❷	{❸}	5000	3000	2000
❷	{❶, ❸}	6000	4000	2000
❸	$\emptyset$	3000	0	3000
❸	{❶}	4000	1000	3000
❸	{❷}	5000	2000	3000
❸	{❶, ❷}	6000	3000	3000

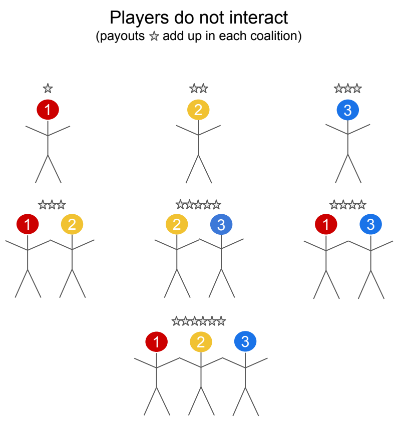


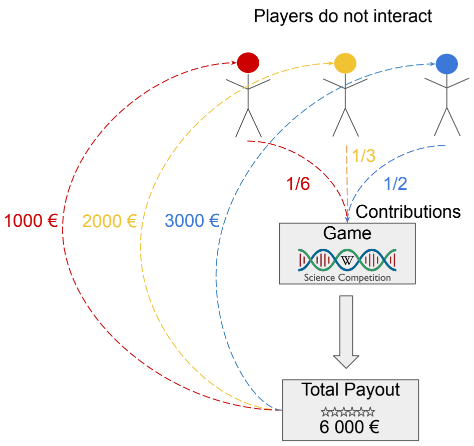
- **No interactions:** Each player contrib.s same fixed value to each coalition
  - ⇝ Player ❶ always adds 1000, ❷ adds 2000, and ❸ adds 3000
  - ⇝ Marginal contributions are constant across all coalitions S
- **Conclusion:** Fair payout $=$ average marginal contribution across all S
  - ⇝ Total value $v(P) = 6000$ splits proportionally by individual contribs:

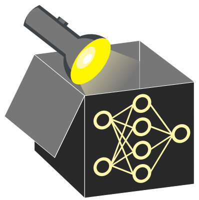
$$❶ = \tfrac{1}{6}, \quad ❷ = \tfrac{1}{3}, \quad ❸ = \tfrac{1}{2}$$

©

COOPERATIVE GAMES - NO INTERACTIONS

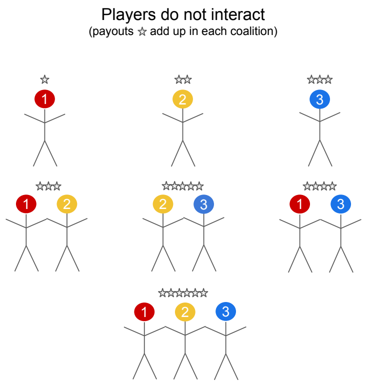


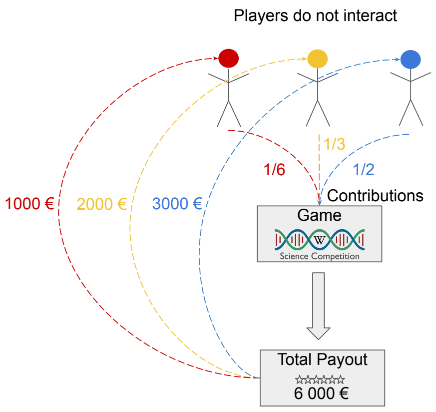


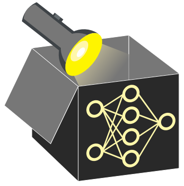


⇒ Fair payouts are trivial without interactions

COOPERATIVE GAMES - NO INTERACTIONS



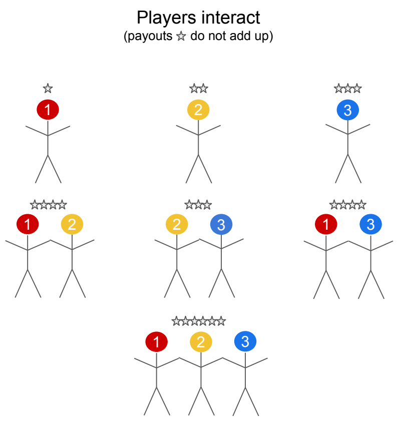


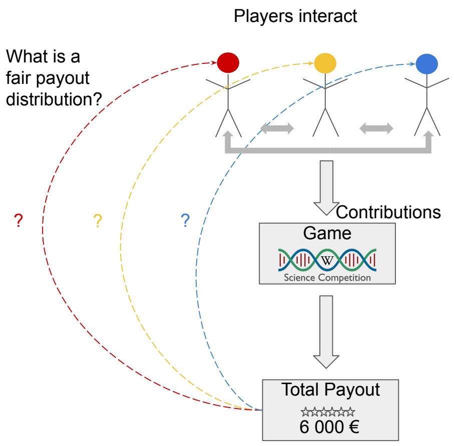


⇒ Fair payouts are trivial without interactions

©

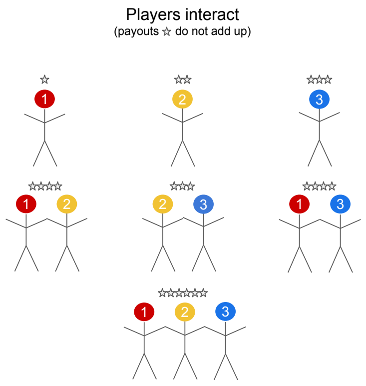
COOPERATIVE GAMES - INTERACTIONS

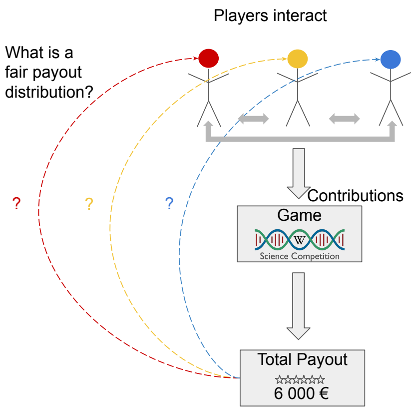




⇒ Unclear how to fairly distribute payouts when players interact

COOPERATIVE GAMES - INTERACTIONS





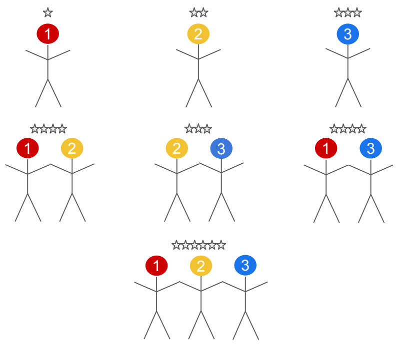
⇒ Unclear how to fairly distribute payouts when players interact

©

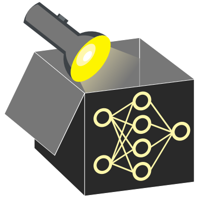
Interpretable Machine Learning – 3 / 11

COOPERATIVE GAMES - INTERACTIONS

Players interact
(payouts ☆ do not add up)

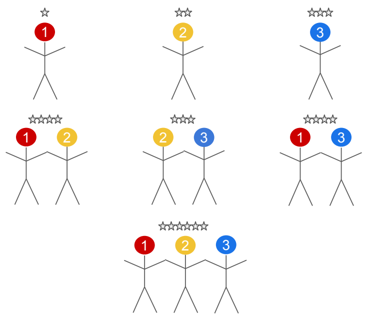


Player	Coalition S	$v(S \cup \{j\})$	$v(S)$	$\Delta(j, S)$
1	$\emptyset$	1000	0	1000
1	{2}	4000	2000	2000
1	{3}	4000	3000	1000
1	{2, 3}	6000	3000	3000
2	$\emptyset$	2000	0	2000
2	{1}	4000	1000	3000
2	{3}	3000	3000	0
2	{1, 3}	6000	4000	2000
3	$\emptyset$	3000	0	3000
3	{1}	4000	1000	3000
3	{2}	3000	2000	1000
3	{1, 2}	6000	4000	2000

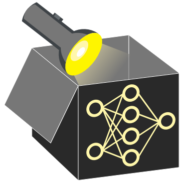


COOPERATIVE GAMES - INTERACTIONS

Players interact
(payouts ☆ do not add up)



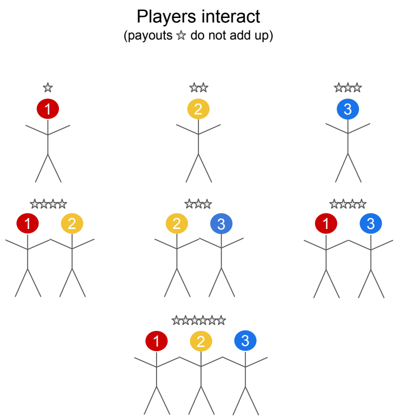
Player	Coalition S	$v(S \cup \{j\})$	$v(S)$	$\Delta(j, S)$
1	$\emptyset$	1000	0	1000
1	{2}	4000	2000	2000
1	{3}	4000	3000	1000
1	{2, 3}	6000	3000	3000
2	$\emptyset$	2000	0	2000
2	{1}	4000	1000	3000
2	{3}	3000	3000	0
2	{1, 3}	6000	4000	2000
3	$\emptyset$	3000	0	3000
3	{1}	4000	1000	3000
3	{2}	3000	2000	1000
3	{1, 2}	6000	4000	2000



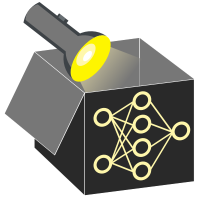
©

Interpretable Machine Learning – 4 / 11

COOPERATIVE GAMES - INTERACTIONS

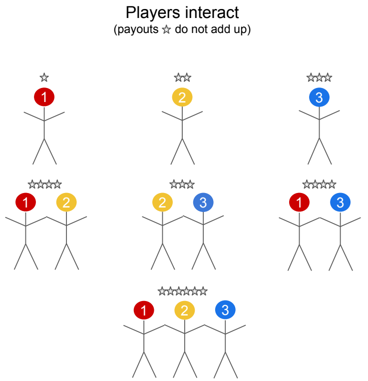


Player	Coalition S	$v(S \cup \{j\})$	$v(S)$	$\Delta(j, S)$
❶	$\emptyset$	1000	0	1000
❶	{❷}	4000	2000	2000
❶	{❸}	4000	3000	1000
❶	{❷, ❸}	6000	3000	3000
❷	$\emptyset$	2000	0	2000
❷	{❶}	4000	1000	3000
❷	{❸}	3000	3000	0
❷	{❶, ❸}	6000	4000	2000
❸	$\emptyset$	3000	0	3000
❸	{❶}	4000	1000	3000
❸	{❷}	3000	2000	1000
❸	{❶, ❷}	6000	4000	2000

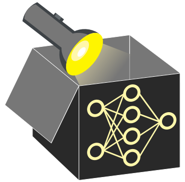


- **With interactions:** Players contribute different amounts depending on coalition
  $\rightsquigarrow$ Marginal contributions vary across coalitions S (e.g., due to overlap, synergy)
- Averaging over subsets does not recover total payout $v(P)$ $\rightsquigarrow$ unfair payout distr.
  $\rightsquigarrow$ average contrib. ❶ = 1750, ❷ = 1750, ❸ = 2250 do not sum to $v(P) = 6000$
- Value a player adds depends on joining order, not just who else is in the coalition
  $\rightsquigarrow$ Shapley values fairly average over all possible joining orders

COOPERATIVE GAMES - INTERACTIONS



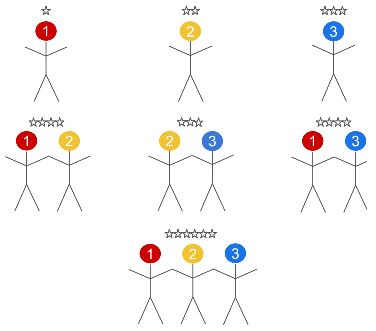
Player	Coalition S	$v(S \cup \{j\})$	$v(S)$	$\Delta(j, S)$
❶	$\emptyset$	1000	0	1000
❶	{❷}	4000	2000	2000
❶	{❸}	4000	3000	1000
❶	{❷, ❸}	6000	3000	3000
❷	$\emptyset$	2000	0	2000
❷	{❶}	4000	1000	3000
❷	{❸}	3000	3000	0
❷	{❶, ❸}	6000	4000	2000
❸	$\emptyset$	3000	0	3000
❸	{❶}	4000	1000	3000
❸	{❷}	3000	2000	1000
❸	{❶, ❷}	6000	4000	2000

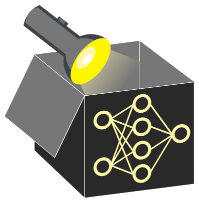


- **With interactions:** Players contribute differently depending on coalition
  $\rightsquigarrow$ Marginal contribs vary across coalitions S (e.g. overlap, synergy)
- Averaging over subsets does not recover total payout $v(P)$
  $\rightsquigarrow$ unfair payout distribution
  $\rightsquigarrow$ avg. contrib. ❶ = 1750 ❷ = 1750 ❸ = 2250 don't sum to $v(P) = 6000$
- Value a player adds depends on joining order, not just who's in coalition
  $\rightsquigarrow$ Shapley values fairly average over all possible joining orders

©

COOPERATIVE GAMES - INTERACTIONS





Ordering 1: ❸ → ❷ → ❶

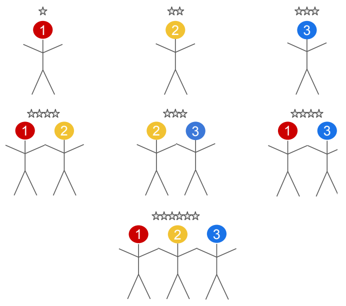
- ❸ joins alone: 3 ☆
- ❷ joins: total = 3 ☆, marginal = 0
- ❶ joins: total = 6 ☆, marginal = +3

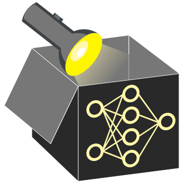
But what if ❶ joins before ❷?

Ordering 2: ❸ → ❶ → ❷

- ❸ joins alone: 3 ☆
- ❶ joins: total = 4 ☆, marginal = +1
- ❷ joins: total = 6 ☆, marginal = +2

COOPERATIVE GAMES - INTERACTIONS





Ordering 1: ❸ → ❷ → ❶

- ❸ joins alone: 3 ☆
- ❷ joins: total = 3 ☆, marginal = 0
- ❶ joins: total = 6 ☆, marginal = +3

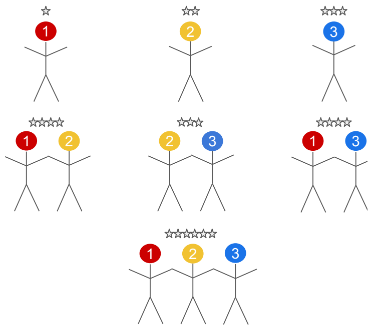
But what if ❶ joins before ❷?

Ordering 2: ❸ → ❶ → ❷

- ❸ joins alone: 3 ☆
- ❶ joins: total = 4 ☆, marginal = +1
- ❷ joins: total = 6 ☆, marginal = +2

©

COOPERATIVE GAMES - INTERACTIONS



Ordering 1: ❸ → ❷ → ❶

- ❸ joins alone: 3 ☆
- ❷ joins: total = 3 ☆, marginal = 0
- ❶ joins: total = 6 ☆, marginal = +3

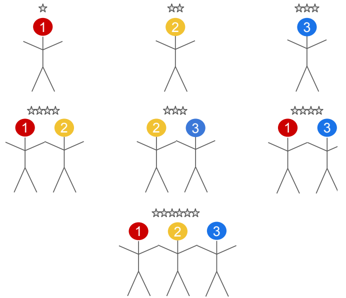
But what if ❶ joins before ❷?

Ordering 2: ❸ → ❶ → ❷

- ❸ joins alone: 3 ☆
- ❶ joins: total = 4 ☆, marginal = +1
- ❷ joins: total = 6 ☆, marginal = +2

- **Order sensitivity:** A player's marginal contribution depends on when they join S
- **Shapley value:** Averages each player's contribution over all possible join orders
  - ⇝ Resolves redundancy (e.g., ❸'s contribution/skill overlaps with ❷'s)
  - ⇝ Accounts for order sensitivity (e.g., ❶ brings more value if added last)
  - ⇝ Ensures fairness (no player is advantaged or penalized by order of joining)

COOPERATIVE GAMES - INTERACTIONS



Ordering 1: ❸ → ❷ → ❶

- ❸ joins alone: 3 ☆
- ❷ joins: total = 3 ☆, marginal = 0
- ❶ joins: total = 6 ☆, marginal = +3

But what if ❶ joins before ❷?

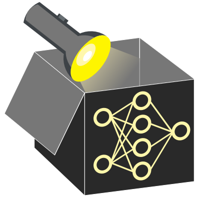
Ordering 2: ❸ → ❶ → ❷

- ❸ joins alone: 3 ☆
- ❶ joins: total = 4 ☆, marginal = +1
- ❷ joins: total = 6 ☆, marginal = +2

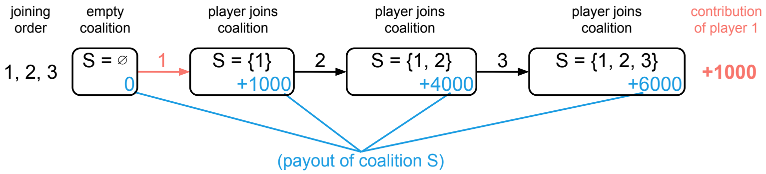
- **Order sensitivity:** A player's marginal contribution depends on when they join S
- **Shapley value:** Averages each player's contribution over all possible join orders
  - ⇝ Resolves redundancy (e.g., ❸'s contribution/skill overlaps with ❷'s)
  - ⇝ Accounts for order sensitivity (e.g., ❶ brings more value if added last)
  - ⇝ Ensures fairness (order of joining gives no advantage/disadvantage)

©

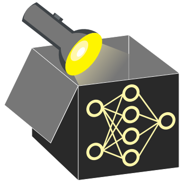
SHAPLEY VALUES - ILLUSTRATION



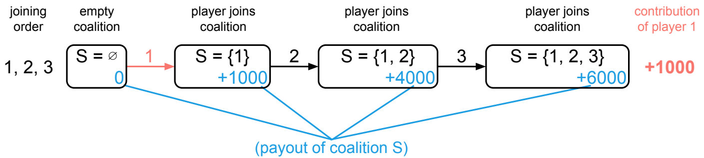
- Generate all possible joining orders of players (all permutations of full set *P*)
- For each order: track player *j*-th marginal contribution when *j* joins a coalition



SHAPLEY VALUES - ILLUSTRATION



- Generate all possible joining orders (all permutations of full set *P*)
- For each order: track player *j*-th marginal contrib when *j* joins a coalition



©

SHAPLEY VALUES - ILLUSTRATION

- Generate all possible joining orders of players (all permutations of full set P)
- For each order: track player j-th marginal contribution when j joins a coalition
- Shapley value of j: Average this marginal contribution over all joining orders
- **Example:** Compute payout difference after player 1 enters coalition ⇝ average

joining order	empty coalition		player joins coalition		player joins coalition		player joins coalition	contribution of player 1
1, 2, 3	S = ∅ 0	1	S = {1} +1000	2	S = {1, 2} +4000	3	S = {1, 2, 3} +6000	+1000
1, 3, 2	S = ∅ 0	1	S = {1} +1000	3	S = {1, 3} +4000	2	S = {1, 2, 3} +6000	+1000
2, 1, 3	S = ∅ 0	2	S = {2} +2000	1	S = {1, 2} +4000	3	S = {1, 2, 3} +6000	+2000
2, 3, 1	S = ∅ 0	2	S = {2} +2000	3	S = {2, 3} +3000	1	S = {1, 2, 3} +6000	+3000
3, 1, 2	S = ∅ 0	3	S = {3} +3000	1	S = {1, 3} +4000	2	S = {1, 2, 3} +6000	+1000
3, 2, 1	S = ∅ 0	3	S = {3} +3000	2	S = {2, 3} +3000	1	S = {1, 2, 3} +6000	+3000

Shapley value of player 1: +1833.33

SHAPLEY VALUES - ILLUSTRATION

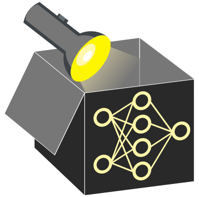
- Generate all possible joining orders (all permutations of full set P)
- For each order: track player j-th marginal contrib when j joins a coalition
- Shapley value of j: Average this marginal contrib over all joining orders
- **Example:** Compute payout diff. after player 1 enters coalition ⇝ average

joining order	empty coalition		player joins coalition		player joins coalition		player joins coalition	contribution of player 1
1, 2, 3	S = ∅ 0	1	S = {1} +1000	2	S = {1, 2} +4000	3	S = {1, 2, 3} +6000	+1000
1, 3, 2	S = ∅ 0	1	S = {1} +1000	3	S = {1, 3} +4000	2	S = {1, 2, 3} +6000	+1000
2, 1, 3	S = ∅ 0	2	S = {2} +2000	1	S = {1, 2} +4000	3	S = {1, 2, 3} +6000	+2000
2, 3, 1	S = ∅ 0	2	S = {2} +2000	3	S = {2, 3} +3000	1	S = {1, 2, 3} +6000	+3000
3, 1, 2	S = ∅ 0	3	S = {3} +3000	1	S = {1, 3} +4000	2	S = {1, 2, 3} +6000	+1000
3, 2, 1	S = ∅ 0	3	S = {3} +3000	2	S = {2, 3} +3000	1	S = {1, 2, 3} +6000	+3000

Shapley value of player 1: +1833.33

©

SHAPLEY VALUES - ILLUSTRATION

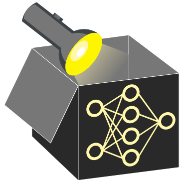


- Generate all possible joining orders of players (all permutations of full set P)
- For each order: track player j-th marginal contribution when j joins a coalition
- Shapley value of j: Average this marginal contribution over all joining orders
- **Example:** Compute payout difference after player 2 enters coalition $\rightsquigarrow$ average

joining order	empty coalition		player joins coalition		player joins coalition		player joins coalition	contribution of player 2
1, 2, 3	S = ∅ 0	1	S = {1} +1000	2	S = {1, 2} +4000	3	S = {1, 2, 3} +6000	+3000
1, 3, 2	S = ∅ 0	1	S = {1} +1000	3	S = {1, 3} +4000	2	S = {1, 2, 3} +6000	+2000
2, 1, 3	S = ∅ 0	2	S = {2} +2000	1	S = {1, 2} +4000	3	S = {1, 2, 3} +6000	+2000
2, 3, 1	S = ∅ 0	2	S = {2} +2000	3	S = {2, 3} +3000	1	S = {1, 2, 3} +6000	+2000
3, 1, 2	S = ∅ 0	3	S = {3} +3000	1	S = {1, 3} +4000	2	S = {1, 2, 3} +6000	+2000
3, 2, 1	S = ∅ 0	3	S = {3} +3000	2	S = {2, 3} +3000	1	S = {1, 2, 3} +6000	+0

Shapley value of player 2: **+1833.33**

SHAPLEY VALUES - ILLUSTRATION



- Generate all possible joining orders (all permutations of full set P)
- For each order: track player j-th marginal contrib when j joins a coalition
- Shapley value of j: Average this marginal contrib over all joining orders
- **Example:** Compute payout diff. after player 2 enters coalition $\rightsquigarrow$ average

joining order	empty coalition		player joins coalition		player joins coalition		player joins coalition	contribution of player 2
1, 2, 3	S = ∅ 0	1	S = {1} +1000	2	S = {1, 2} +4000	3	S = {1, 2, 3} +6000	+3000
1, 3, 2	S = ∅ 0	1	S = {1} +1000	3	S = {1, 3} +4000	2	S = {1, 2, 3} +6000	+2000
2, 1, 3	S = ∅ 0	2	S = {2} +2000	1	S = {1, 2} +4000	3	S = {1, 2, 3} +6000	+2000
2, 3, 1	S = ∅ 0	2	S = {2} +2000	3	S = {2, 3} +3000	1	S = {1, 2, 3} +6000	+2000
3, 1, 2	S = ∅ 0	3	S = {3} +3000	1	S = {1, 3} +4000	2	S = {1, 2, 3} +6000	+2000
3, 2, 1	S = ∅ 0	3	S = {3} +3000	2	S = {2, 3} +3000	1	S = {1, 2, 3} +6000	+0

Shapley value of player 2: **+1833.33**

©

SHAPLEY VALUES - ILLUSTRATION

- Generate all possible joining orders of players (all permutations of full set P)
- For each order: track player j-th marginal contribution when j joins a coalition
- Shapley value of j: Average this marginal contribution over all joining orders
- **Example:** Compute payout difference after player 3 enters coalition ⇝ average

joining order	empty coalition		player joins coalition		player joins coalition		player joins coalition	contribution of player 3
1, 2, 3	S = ∅ 0	1	S = {1} +1000	2	S = {1, 2} +4000	3	S = {1, 2, 3} +6000	+2000
1, 3, 2	S = ∅ 0	1	S = {1} +1000	3	S = {1, 3} +4000	2	S = {1, 2, 3} +6000	+3000
2, 1, 3	S = ∅ 0	2	S = {2} +2000	1	S = {1, 2} +4000	3	S = {1, 2, 3} +6000	+2000
2, 3, 1	S = ∅ 0	2	S = {2} +2000	3	S = {2, 3} +3000	1	S = {1, 2, 3} +6000	+1000
3, 1, 2	S = ∅ 0	3	S = {3} +3000	1	S = {1, 3} +4000	2	S = {1, 2, 3} +6000	+3000
3, 2, 1	S = ∅ 0	3	S = {3} +3000	2	S = {2, 3} +3000	1	S = {1, 2, 3} +6000	+3000

Shapley value of player 3: **+2333.33**

SHAPLEY VALUES - ILLUSTRATION

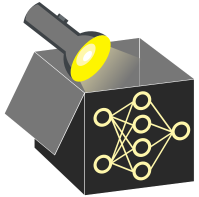
- Generate all possible joining orders (all permutations of full set P)
- For each order: track player j-th marginal contrib when j joins a coalition
- Shapley value of j: Average this marginal contrib over all joining orders
- **Example:** Compute payout diff. after player 3 enters coalition ⇝ average

joining order	empty coalition		player joins coalition		player joins coalition		player joins coalition	contribution of player 3
1, 2, 3	S = ∅ 0	1	S = {1} +1000	2	S = {1, 2} +4000	3	S = {1, 2, 3} +6000	+2000
1, 3, 2	S = ∅ 0	1	S = {1} +1000	3	S = {1, 3} +4000	2	S = {1, 2, 3} +6000	+3000
2, 1, 3	S = ∅ 0	2	S = {2} +2000	1	S = {1, 2} +4000	3	S = {1, 2, 3} +6000	+2000
2, 3, 1	S = ∅ 0	2	S = {2} +2000	3	S = {2, 3} +3000	1	S = {1, 2, 3} +6000	+1000
3, 1, 2	S = ∅ 0	3	S = {3} +3000	1	S = {1, 3} +4000	2	S = {1, 2, 3} +6000	+3000
3, 2, 1	S = ∅ 0	3	S = {3} +3000	2	S = {2, 3} +3000	1	S = {1, 2, 3} +6000	+3000

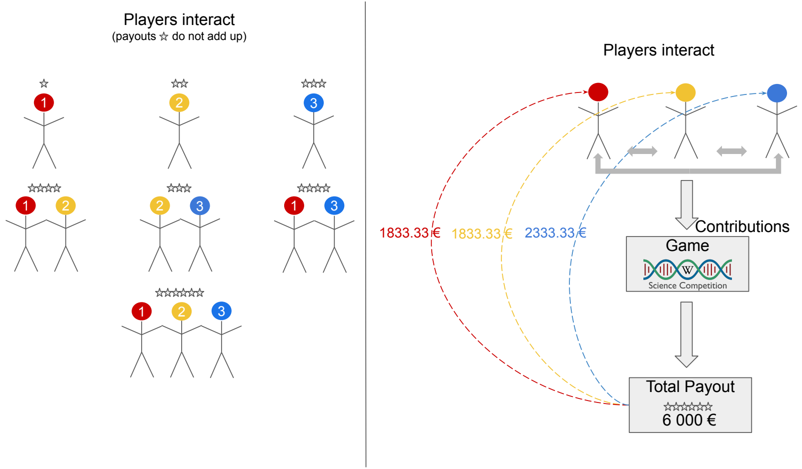
Shapley value of player 3: **+2333.33**

©

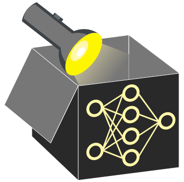
SHAPLEY VALUES - ILLUSTRATION



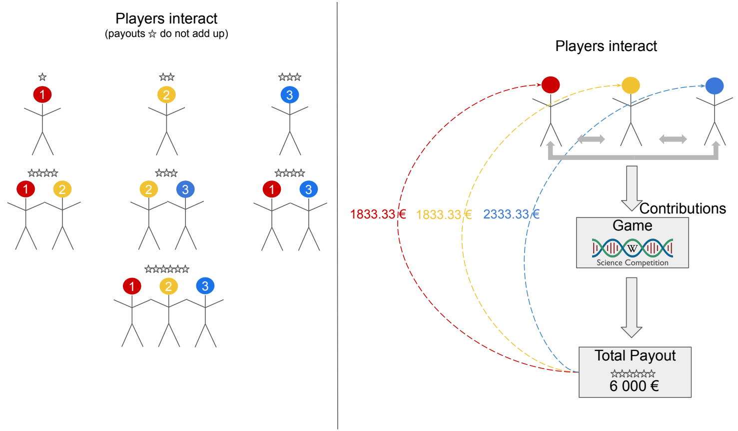
- Generate all possible joining orders of players (all permutations of full set P)
- For each order: track player j-th marginal contribution when j joins a coalition
- Shapley value of j: Average this marginal contribution over all joining orders



SHAPLEY VALUES - ILLUSTRATION



- Generate all possible joining orders (all permutations of full set P)
- For each order: track player j-th marginal contrib when j joins a coalition
- Shapley value of j: Average this marginal contrib over all joining orders



©

SHAPLEY VALUE - ORDER DEFINITION

The Shapley value order definition averages the marginal contribution of a player across all possible player orderings:

$$\phi_j = \frac{1}{|P|!} \sum_{\tau \in \Pi} (v(S_j^\tau \cup \{j\}) - v(S_j^\tau))$$

- Π: Set of all permutations (joining orders) of the players – there are $|P|!$ in total

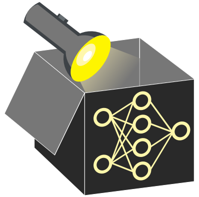
SHAPLEY VALUE - ORDER DEFINITION

The Shapley value order definition averages the marginal contribution of a player across all possible player orderings:

$$\phi_j = \frac{1}{|P|!} \sum_{\tau \in \Pi} (v(S_j^\tau \cup \{j\}) - v(S_j^\tau))$$

- Π: Set of all permutations (joining orders) of the players – $|P|!$ in total

SHAPLEY VALUE - ORDER DEFINITION

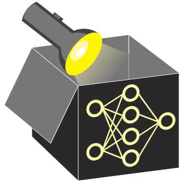


The Shapley value order definition averages the marginal contribution of a player across all possible player orderings:

$$\phi_j = \frac{1}{|P|!} \sum_{\tau \in \Pi} (v(S_j^\tau \cup \{j\}) - v(S_j^\tau))$$

- Π: Set of all permutations (joining orders) of the players – there are $|P|!$ in total
- S_j^τ: Set of players before j joins, for each ordering $\tau = (\tau^{(1)}, \dots, \tau^{(p)})$
  E.g.: $\Pi = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$
  $\leadsto$ For joining order $\tau = (2, 1, 3)$ and player $j = 3 \Rightarrow S_j^\tau = \{2, 1\}$
  $\leadsto$ For joining order $\tau = (3, 1, 2)$ and player $j = 1 \Rightarrow S_j^\tau = \{3\}$

SHAPLEY VALUE - ORDER DEFINITION



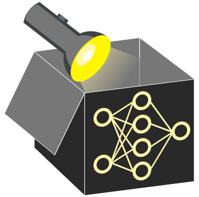
The Shapley value order definition averages the marginal contribution of a player across all possible player orderings:

$$\phi_j = \frac{1}{|P|!} \sum_{\tau \in \Pi} (v(S_j^\tau \cup \{j\}) - v(S_j^\tau))$$

- Π: Set of all permutations (joining orders) of the players – $|P|!$ in total
- S_j^τ: Set of players before j joins, for each ordering $\tau = (\tau^{(1)}, \dots, \tau^{(p)})$
  E.g.: $\Pi = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$
  $\leadsto$ For joining order $\tau = (2, 1, 3)$ and player $j = 3 \Rightarrow S_j^\tau = \{2, 1\}$
  $\leadsto$ For joining order $\tau = (3, 1, 2)$ and player $j = 1 \Rightarrow S_j^\tau = \{3\}$

©

SHAPLEY VALUE - ORDER DEFINITION



The Shapley value order definition averages the marginal contribution of a player across all possible player orderings:

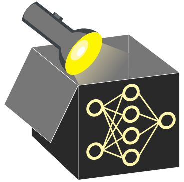
$$\phi_j = \frac{1}{|P|!} \sum_{\tau \in \Pi} (v(S_j^\tau \cup \{j\}) - v(S_j^\tau))$$

- Π: Set of all permutations (joining orders) of the players – there are $|P|!$ in total
- S_j^τ: Set of players before j joins, for each ordering $\tau = (\tau^{(1)}, \dots, \tau^{(p)})$
  E.g.: $\Pi = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$
  $\rightsquigarrow$ For joining order $\tau = (2, 1, 3)$ and player $j = 3 \Rightarrow S_j^\tau = \{2, 1\}$
  $\rightsquigarrow$ For joining order $\tau = (3, 1, 2)$ and player $j = 1 \Rightarrow S_j^\tau = \{3\}$
- Order definition allows to approximate Shapley values by sampling permutations
  $\rightsquigarrow$ Sample a fixed number $M \ll |P|!$ of random permutations and average:

$$\phi_j \approx \frac{1}{M} \sum_{\tau \in \Pi_M} (v(S_j^\tau \cup \{j\}) - v(S_j^\tau))$$

where $\Pi_M \subset \Pi$ is the random sample of M player orderings

SHAPLEY VALUE - ORDER DEFINITION



The Shapley value order definition averages the marginal contribution of a player across all possible player orderings:

$$\phi_j = \frac{1}{|P|!} \sum_{\tau \in \Pi} (v(S_j^\tau \cup \{j\}) - v(S_j^\tau))$$

- Π: Set of all permutations (joining orders) of the players – $|P|!$ in total
- S_j^τ: Set of players before j joins, for each ordering $\tau = (\tau^{(1)}, \dots, \tau^{(p)})$
  E.g.: $\Pi = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$
  $\rightsquigarrow$ For joining order $\tau = (2, 1, 3)$ and player $j = 3 \Rightarrow S_j^\tau = \{2, 1\}$
  $\rightsquigarrow$ For joining order $\tau = (3, 1, 2)$ and player $j = 1 \Rightarrow S_j^\tau = \{3\}$
- Order definition allows to approximate Shapley values by sampling permutations
  $\rightsquigarrow$ Sample a fixed $M \ll |P|!$ random permutations and average:

$$\phi_j \approx \frac{1}{M} \sum_{\tau \in \Pi_M} (v(S_j^\tau \cup \{j\}) - v(S_j^\tau))$$

where $\Pi_M \subset \Pi$ is the random sample of M player orderings

©

FROM ORDER DEFINITION TO SET DEFINITION

- **Note:** The same subset S_j^τ can occur in multiple permutations (joining orders)
  $\rightsquigarrow$ Its marginal contribution is included multiple times in the sum in ϕ_j
- **Example (for set of players $P = \{❶, ❷, ❸\}$, player of interest $j = ❸$):**
  $\Pi = \{\underline{(❶, ❷, ❸)}, (❶, ❸, ❷), \underline{(❷, ❶, ❸)}, (❷, ❸, ❶), (❸, ❶, ❷), (❸, ❷, ❶)\}$
  $\rightsquigarrow$ In both $\underline{(❶, ❷, ❸)}$ and $\underline{(❷, ❶, ❸)}$, player ❸ joins after coalition $S_j^\tau = \{❶, ❷\}$
  $\Rightarrow$ Marginal contribution $v(\{❶, ❷, ❸\}) - v(\{❶, ❷\})$ occurs twice in ϕ_j
- **Reason:** Each subset S appears in $|S|!(|P| - |S| - 1)!$ orderings before j joins
  $\Rightarrow$ There are $|S|!$ possible orders of players within coalition S
  $\Rightarrow$ There are $(|P| - |S| - 1)!$ possible orders of players without S and j

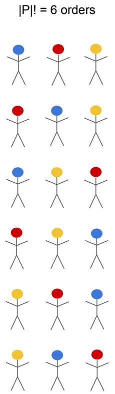
$$\underbrace{\overbrace{\tau^{(1)} \mid \ldots \mid \tau^{(|S|)}}^{|S|!\text{ permutations}}}_{\text{Players before player } j} \mid \underbrace{\tau^{(|S|+1)}}_{\text{player } j} \mid \underbrace{\overbrace{\tau^{(|S|+2)} \mid \ldots \mid \tau^{(p)}}^{(|P|-|S|-1)!\text{ permutations}}}_{\text{Players after player } j}$$

FROM ORDER DEFINITION TO SET DEFINITION

- **Note:** The same subset S_j^τ can occur in multiple permutations
  $\rightsquigarrow$ Its marginal contribution is included multiple times in the sum in ϕ_j
- **Example Π (for players $P = \{❶, ❷, ❸\}$, player of interest $j = ❸$):**
  $\{\underline{(❶, ❷, ❸)}, (❶, ❸, ❷), \underline{(❷, ❶, ❸)}, (❷, ❸, ❶), (❸, ❶, ❷), (❸, ❷, ❶)\}$
  $\rightsquigarrow$ In $\underline{(❶, ❷, ❸)}$ and $\underline{(❷, ❶, ❸)}$, player ❸ joins after coal. $S_j^\tau = \{❶, ❷\}$
  $\Rightarrow$ Marginal contribution $v(\{❶, ❷, ❸\}) - v(\{❶, ❷\})$ occurs twice in ϕ_j
- **Reason:** Each subset S appears in $|S|!(|P| - |S| - 1)!$ orderings before j joins
  $\Rightarrow$ There are $|S|!$ possible orders of players within coalition S
  $\Rightarrow$ There are $(|P| - |S| - 1)!$ possible orders of players without S and j

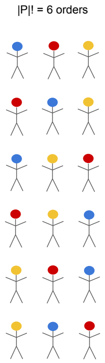
$$\underbrace{\overbrace{\tau^{(1)} \mid \ldots \mid \tau^{(|S|)}}^{|S|!\text{ permutations}}}_{\text{Players before player } j} \mid \underbrace{\tau^{(|S|+1)}}_{\text{player } j} \mid \underbrace{\overbrace{\tau^{(|S|+2)} \mid \ldots \mid \tau^{(p)}}^{(|P|-|S|-1)!\text{ permutations}}}_{\text{Players after player } j}$$

FROM ORDER DEFINITION TO SET DEFINITION



- **Order view:** Each of the $|P|!$ permutations contributes one term with weight $\frac{1}{|P|!}$
- Same subset $S \subseteq P \setminus \{j\}$ can appear before j in multiple orders
  $\rightsquigarrow$ e.g., S = {🔵, 🔴} = {🔴, 🔵}
- **Set view:** Group by unique subsets S, not permutations
- Each S occurs in $|S|!(|P| - |S| - 1)!$ orderings $\rightsquigarrow$ Weight: $\frac{|S|!(|P|-|S|-1)!}{|P|!}$

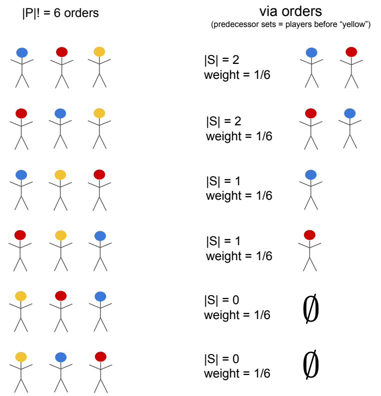
FROM ORDER DEFINITION TO SET DEFINITION

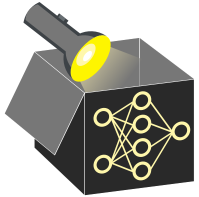


- **Order view:** Each of the $|P|!$ permuts contributes 1 term with weight $\frac{1}{|P|!}$
- Same subset $S \subseteq P \setminus \{j\}$ can appear before j in multiple orders
  $\rightsquigarrow$ e.g., S = {🔵, 🔴} = {🔴, 🔵}
- **Set view:** Group by unique subsets S, not permutations
- Each S occurs in $|S|!(|P| - |S| - 1)!$ orderings $\rightsquigarrow$ Weight: $\frac{|S|!(|P|-|S|-1)!}{|P|!}$

©

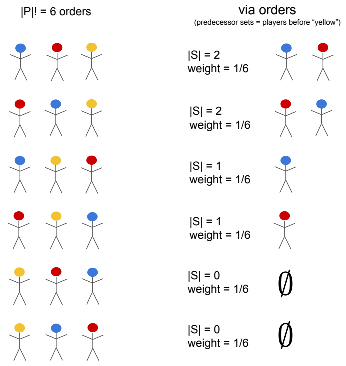
FROM ORDER DEFINITION TO SET DEFINITION

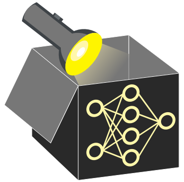




- **Order view:** Each of the $|P|!$ permutations contributes one term with weight $\frac{1}{|P|!}$
- Same subset $S \subseteq P \setminus \{j\}$ can appear before j in multiple orders
  $\rightsquigarrow$ e.g., S = {●, ●} = {●, ●}
- **Set view:** Group by unique subsets S, not permutations
- Each S occurs in $|S|!(|P| - |S| - 1)!$ orderings $\rightsquigarrow$ Weight: $\frac{|S|!(|P|-|S|-1)!}{|P|!}$

FROM ORDER DEFINITION TO SET DEFINITION

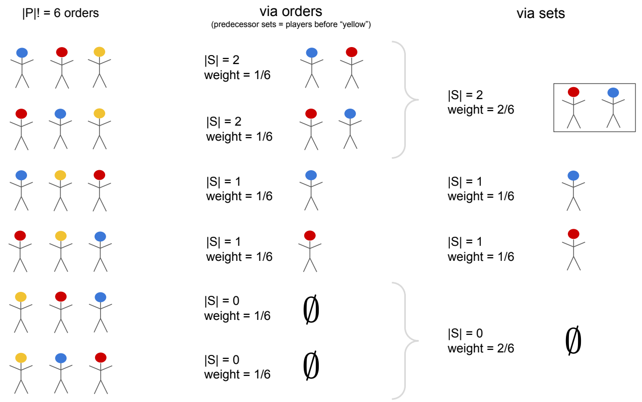


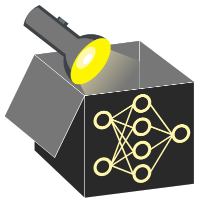


- **Order view:** Each of the $|P|!$ permuts contributes 1 term with weight $\frac{1}{|P|!}$
- Same subset $S \subseteq P \setminus \{j\}$ can appear before j in multiple orders
  $\rightsquigarrow$ e.g., S = {●, ●} = {●, ●}
- **Set view:** Group by unique subsets S, not permutations
- Each S occurs in $|S|!(|P| - |S| - 1)!$ orderings $\rightsquigarrow$ Weight: $\frac{|S|!(|P|-|S|-1)!}{|P|!}$

©

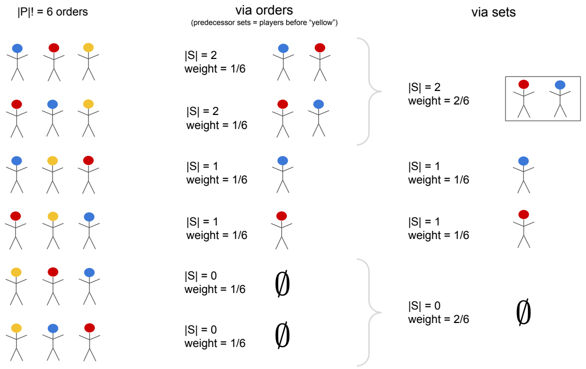
FROM ORDER DEFINITION TO SET DEFINITION

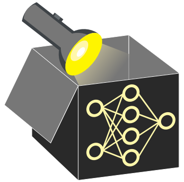




- **Order view:** Each of the $|P|!$ permutations contributes one term with weight $\frac{1}{|P|!}$
- Same subset $S \subseteq P \setminus \{j\}$ can appear before j in multiple orders
  $\rightsquigarrow$ e.g., S = {🔵, 🔴} = {🔴, 🔵}
- **Set view:** Group by unique subsets S, not permutations
- Each S occurs in $|S|!(|P|-|S|-1)!$ orderings $\rightsquigarrow$ Weight: $\frac{|S|!(|P|-|S|-1)!}{|P|!}$

FROM ORDER DEFINITION TO SET DEFINITION





- **Order view:** Each of the $|P|!$ permuts contributes 1 term with weight $\frac{1}{|P|!}$
- Same subset $S \subseteq P \setminus \{j\}$ can appear before j in multiple orders
  $\rightsquigarrow$ e.g., S = {🔵, 🔴} = {🔴, 🔵}
- **Set view:** Group by unique subsets S, not permutations
- Each S occurs in $|S|!(|P|-|S|-1)!$ orderings $\rightsquigarrow$ Weight: $\frac{|S|!(|P|-|S|-1)!}{|P|!}$

SHAPLEY VALUE - SET DEFINITION

Shapley value via **set definition** (weighting via multinomial coefficient):

$$\phi_j = \sum_{S \subseteq P \setminus \{j\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} (v(S \cup \{j\}) - v(S))$$

The coefficient gives the probability that, when randomly arranging all $|P|$ players, the exact set S appears before player j, and the remaining players appear afterward.

$$\underbrace{\boxed{\tau^{(1)}}\boxed{\ldots}\boxed{\tau^{(|S|)}}}_{|S|!\text{ permutations}}\underbrace{\boxed{\tau^{(|S|+1)}}}_{\text{player } j}\underbrace{\boxed{\tau^{(|S|+2)}}\boxed{\ldots}\boxed{\tau^{(|P|)}}}_{(|P|-|S|-1)!\text{ permutations}}$$

SHAPLEY VALUE - SET DEFINITION

Shapley value via **set definition** (weighting via multinomial coefficient):

$$\phi_j = \sum \frac{|S|!(|P| - |S| - 1)!}{|P|!} (v() - v(S))$$

The coefficient gives the probability that, when randomly arranging all $|P|$ players, the exact set S appears before player j, and the remaining players appear afterward.

$$\underbrace{\boxed{\tau^{(1)}}\boxed{\ldots}\boxed{\tau^{(|S|)}}}_{|S|!\text{ permutations}}\underbrace{\boxed{\tau^{(|S|+1)}}}_{\text{player } j}\underbrace{\boxed{\tau^{(|S|+2)}}\boxed{\ldots}\boxed{\tau^{(|P|)}}}_{(|P|-|S|-1)!\text{ permutations}}$$

©

SHAPLEY VALUE - SET DEFINITION

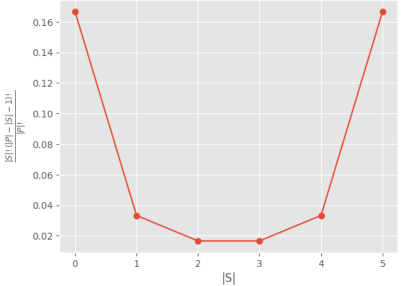
Shapley value via **set definition** (weighting via multinomial coefficient):

$$\phi_j = \sum_{S \subseteq P \setminus \{j\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} (v(S \cup \{j\}) - v(S))$$

The coefficient gives the probability that, when randomly arranging all $|P|$ players, the exact set S appears before player j, and the remaining players appear afterward.

$$\overbrace{\tau^{(1)} \mid \ldots \mid \tau^{(|S|)}}^{|S|!\text{ permutations}} \mid \overbrace{\tau^{(|S|+1)}}^{\text{player } j} \mid \overbrace{\tau^{(|S|+2)} \mid \ldots \mid \tau^{(|P|)}}^{(|P|-|S|-1)!\text{ permutations}}$$

Different Weights Depending on the Size of S



- $|S| = 0$: player j joins first
  $\Rightarrow$ many permutations $\Rightarrow$ high weight
- $|S| = |P| - 1$: player j joins last
  $\Rightarrow$ many permutations $\Rightarrow$ high weight
- Middle-sized $|S|$: fewer exact matches
  $\Rightarrow$ lower weight
- Result: U-shaped weight distribution

SHAPLEY VALUE - SET DEFINITION

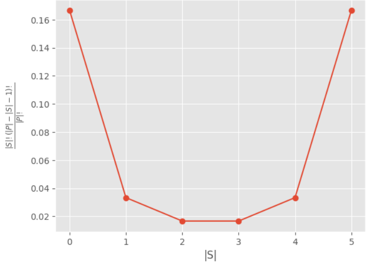
Shapley value via **set definition** (weighting via multinomial coefficient):

$$\phi_j = \sum \frac{|S|!(|P| - |S| - 1)!}{|P|!} (v() - v(S))$$

The coefficient gives the probability that, when randomly arranging all $|P|$ players, the exact set S appears before player j, and the remaining players appear afterward.

$$\overbrace{\tau^{(1)} \mid \ldots \mid \tau^{(|S|)}}^{|S|!\text{ permutations}} \mid \overbrace{\tau^{(|S|+1)}}^{\text{player } j} \mid \overbrace{\tau^{(|S|+2)} \mid \ldots \mid \tau^{(|P|)}}^{(|P|-|S|-1)!\text{ permutations}}$$

Different Weights Depending on the Size of S



- $|S| = 0$: player j joins first
  $\Rightarrow$ many permutations $\Rightarrow$ high weight
- $|S| = |P| - 1$: player j joins last
  $\Rightarrow$ many permutations $\Rightarrow$ high weight
- Middle-sized $|S|$: fewer exact matches
  $\Rightarrow$ lower weight
- Result: U-shaped weight distribution

©

AXIOMS OF FAIR PAYOUTS

What makes a payout fair? The Shapley value provides a fair payout ϕ_j for each player $j \in P$ and uniquely satisfies the following axioms for any value function v:

- **Efficiency**: Total payout $v(P)$ is fully allocated to players:

$$\sum_{j \in P} \phi_j = v(P)$$

AXIOMS OF FAIR PAYOUTS

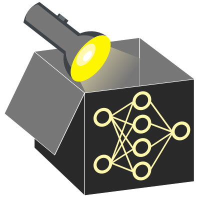
What makes a payout fair? The Shapley value provides a fair payout ϕ_j for each player $j \in P$ and uniquely satisfies the following axioms for any value function v:

- **Efficiency**: Total payout $v(P)$ is fully allocated to players:

$$\sum_{j \in P} \phi_j = v(P)$$

©

AXIOMS OF FAIR PAYOUTS



What makes a payout fair? The Shapley value provides a fair payout ϕ_j for each player $j \in P$ and uniquely satisfies the following axioms for any value function v:

- **Efficiency**: Total payout $v(P)$ is fully allocated to players:

$$\sum_{j \in P} \phi_j = v(P)$$

- **Symmetry**: Indistinguishable players $j, k \in P$ receive equal shares:

$$\text{If } v(S \cup \{j\}) = v(S \cup \{k\}) \text{ for all } S \subseteq P \setminus \{j, k\}, \text{ then } \phi_j = \phi_k$$

AXIOMS OF FAIR PAYOUTS

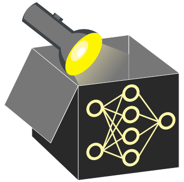
What makes a payout fair? The Shapley value provides a fair payout ϕ_j for each player $j \in P$ and uniquely satisfies the following axioms for any value function v:

- **Efficiency**: Total payout $v(P)$ is fully allocated to players:

$$\sum_{j \in P} \phi_j = v(P)$$

- **Symmetry**: Indistinguishable players $j, k \in P$ receive equal shares:

$$\text{If } v(S \cup \{j\}) = v(S \cup \{k\}) \text{ for all } S \subseteq P \setminus \{j, k\}, \text{ then } \phi_j = \phi_k$$



©

AXIOMS OF FAIR PAYOUTS

What makes a payout fair? The Shapley value provides a fair payout ϕ_j for each player $j \in P$ and uniquely satisfies the following axioms for any value function v:

- **Efficiency**: Total payout $v(P)$ is fully allocated to players:

$$\sum_{j \in P} \phi_j = v(P)$$

- **Symmetry**: Indistinguishable players $j, k \in P$ receive equal shares:

$$\text{If } v(S \cup \{j\}) = v(S \cup \{k\}) \text{ for all } S \subseteq P \setminus \{j, k\}, \text{ then } \phi_j = \phi_k$$

- **Null Player (Dummy)**: Players who contribute nothing receive nothing:

$$\text{If } v(S \cup \{j\}) = v(S) \text{ for all } S \subseteq P \setminus \{j\}, \text{ then } \phi_j = 0$$

AXIOMS OF FAIR PAYOUTS

What makes a payout fair? The Shapley value provides a fair payout ϕ_j for each player $j \in P$ and uniquely satisfies the following axioms for any value function v:

- **Efficiency**: Total payout $v(P)$ is fully allocated to players:

$$\sum_{j \in P} \phi_j = v(P)$$

- **Symmetry**: Indistinguishable players $j, k \in P$ receive equal shares:

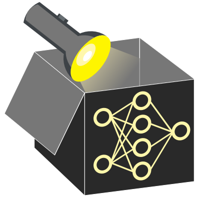
$$\text{If } v(S \cup \{j\}) = v(S \cup \{k\}) \text{ for all } S \subseteq P \setminus \{j, k\}, \text{ then } \phi_j = \phi_k$$

- **Null Player (Dummy)**: Players who contribute nothing receive nothing:

$$\text{If } v(S \cup \{j\}) = v(S) \text{ for all } S \subseteq P \setminus \{j\}, \text{ then } \phi_j = 0$$

©

AXIOMS OF FAIR PAYOUTS



What makes a payout fair? The Shapley value provides a fair payout ϕ_j for each player $j \in P$ and uniquely satisfies the following axioms for any value function v:

- **Efficiency**: Total payout $v(P)$ is fully allocated to players:

$$\sum_{j \in P} \phi_j = v(P)$$

- **Symmetry**: Indistinguishable players $j, k \in P$ receive equal shares:

$$\text{If } v(S \cup \{j\}) = v(S \cup \{k\}) \text{ for all } S \subseteq P \setminus \{j,k\}, \text{ then } \phi_j = \phi_k$$

- **Null Player (Dummy)**: Players who contribute nothing receive nothing:

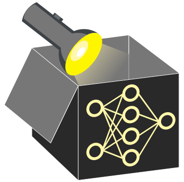
$$\text{If } v(S \cup \{j\}) = v(S) \text{ for all } S \subseteq P \setminus \{j\}, \text{ then } \phi_j = 0$$

- **Additivity**: For two separate games with value functions v_1, v_2, define a combined game with $v(S) = v_1(S) + v_2(S)$ for all $S \subseteq P$. Then:

$$\phi_{j,v_1+v_2} = \phi_{j,v_1} + \phi_{j,v_2}$$

$\rightsquigarrow$ Payout of combined game = payout of the two separate games

AXIOMS OF FAIR PAYOUTS



What makes a payout fair? The Shapley value provides a fair payout ϕ_j for each player $j \in P$ and uniquely satisfies the following axioms for any value function v:

- **Efficiency**: Total payout $v(P)$ is fully allocated to players:

$$\sum_{j \in P} \phi_j = v(P)$$

- **Symmetry**: Indistinguishable players $j, k \in P$ receive equal shares:

$$\text{If } v(S \cup \{j\}) = v(S \cup \{k\}) \text{ for all } S \subseteq P \setminus \{j,k\}, \text{ then } \phi_j = \phi_k$$

- **Null Player (Dummy)**: Players who contribute nothing receive nothing:

$$\text{If } v(S \cup \{j\}) = v(S) \text{ for all } S \subseteq P \setminus \{j\}, \text{ then } \phi_j = 0$$

- **Additivity**: For two separate games with value functions v_1, v_2, define a combined game with $v(S) = v_1(S) + v_2(S)$ for all $S \subseteq P$. Then:

$$\phi_{j,v_1+v_2} = \phi_{j,v_1} + \phi_{j,v_2}$$

$\rightsquigarrow$ Payout of combined game = payout of the two separate games

©