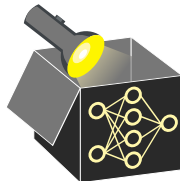
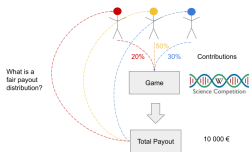


Interpretable Machine Learning



Shapley Shapley Values



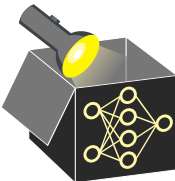
Learning goals

- 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 $S \subseteq P = \{1, \dots, p\}$ can form a coalition to cooperate in a game, each achieving a payout $v(S)$

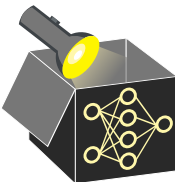


COOPERATIVE GAMES IN GAME THEORY

► SHAPLEY_1951

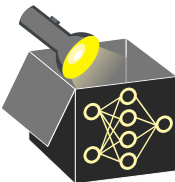
- **Game theory:** Studies strategic interactions among "players" (who act to maximize their utility), where outcomes depend on collective behavior
- **Cooperative games:** Any subset $S \subseteq P = \{1, \dots, p\}$ can form a coalition to cooperate in a game, each achieving a payout $v(S)$
- **Value function:** $v : 2^P \rightarrow \mathbb{R}$ 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

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



COOPERATIVE GAMES IN GAME THEORY

► SHAPLEY_1951



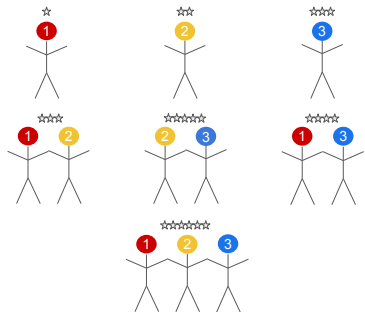
- **Game theory:** Studies strategic interactions among "players" (who act to maximize their utility), where outcomes depend on collective behavior
- **Cooperative games:** Any subset $S \subseteq P = \{1, \dots, p\}$ can form a coalition to cooperate in a game, each achieving a payout $v(S)$
- **Value function:** $v : 2^P \rightarrow \mathbb{R}$ 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

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

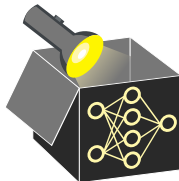
- **Challenge:** Players vary in their contrib. & how they influence each other
- **Goal:** Distribute $v(P)$ among players by considering player interactions
 \rightsquigarrow Assign each player $j \in P$ a fair share ϕ_j (**Shapley value**)

COOPERATIVE GAMES - NO INTERACTIONS

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

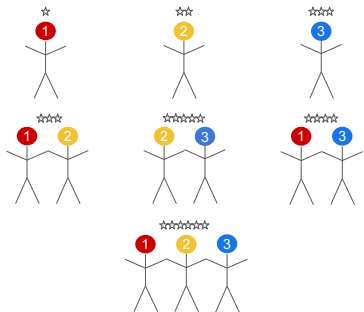


Player	Coalition S	$v(S \cup \{j\})$	$v(S)$	$\Delta(j, S)$
1	\emptyset	1000	0	1000
1	$\{2\}$	3000	2000	1000
1	$\{3\}$	4000	3000	1000
1	$\{2, 3\}$	6000	5000	1000
2	\emptyset	2000	0	2000
2	$\{1\}$	3000	1000	2000
2	$\{3\}$	5000	3000	2000
2	$\{1, 3\}$	6000	4000	2000
3	\emptyset	3000	0	3000
3	$\{1\}$	4000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

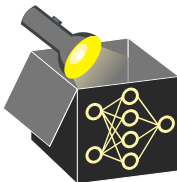


COOPERATIVE GAMES - NO INTERACTIONS

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



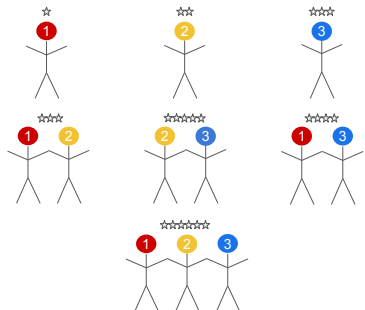
Player	Coalition S	$v(S \cup \{j\})$	$v(S)$	$\Delta(j, S)$
1	\emptyset	1000	0	1000
1	$\{2\}$	3000	2000	1000
1	$\{3\}$	4000	3000	1000
1	$\{2, 3\}$	6000	5000	1000
2	\emptyset	2000	0	2000
2	$\{1\}$	3000	1000	2000
2	$\{3\}$	5000	3000	2000
2	$\{1, 3\}$	6000	4000	2000
3	\emptyset	3000	0	3000
3	$\{1\}$	4000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



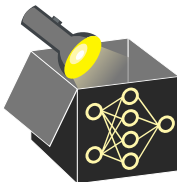
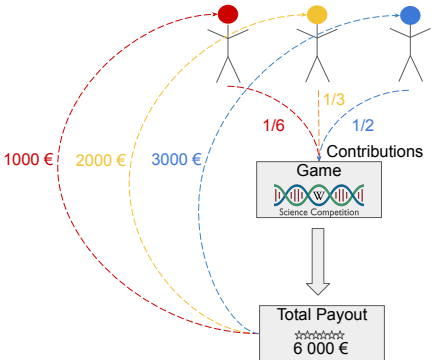
- **No interactions:** Each player contrib.s same fixed value to each coalition
 \rightsquigarrow Player 1 always adds 1000, 2 adds 2000, and 3 adds 3000
 \rightsquigarrow Marginal contributions are constant across all coalitions S
- **Conclusion:** Fair payout = average marginal contribution across all S
 \rightsquigarrow Total value $v(P) = 6000$ splits proportionally by individual contribs:
 $1 = \frac{1}{6}, \quad 2 = \frac{1}{3}, \quad 3 = \frac{1}{2}$

COOPERATIVE GAMES - NO INTERACTIONS

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

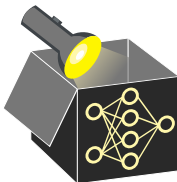
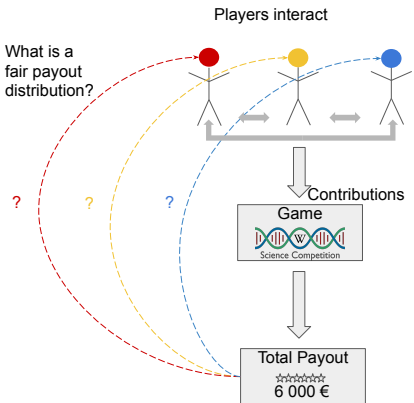
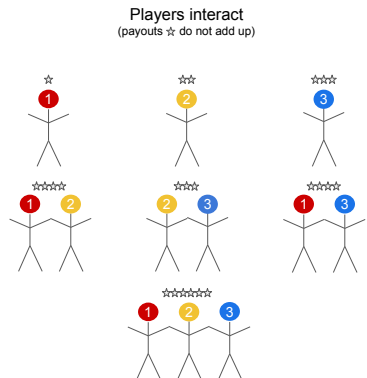


Players do not interact



⇒ Fair payouts are trivial without interactions

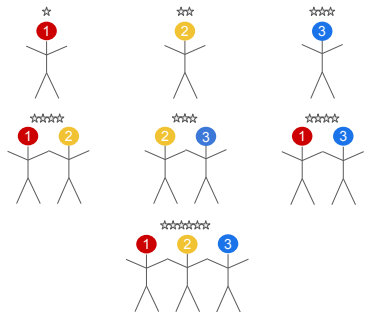
COOPERATIVE GAMES - INTERACTIONS



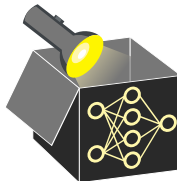
⇒ Unclear how to fairly distribute payouts when players interact

COOPERATIVE GAMES - INTERACTIONS

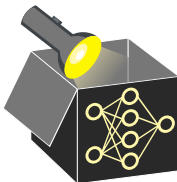
Players interact
(payouts ☆ do not add up)



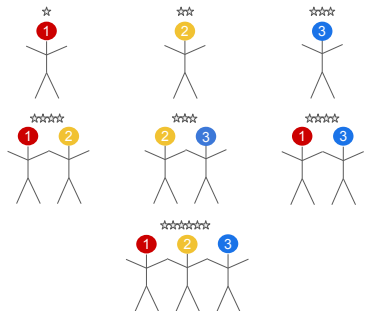
Player	Coalition S	$v(S \cup \{j\})$	$v(S)$	$\Delta(j, S)$
1	\emptyset	1000	0	1000
1	$\{2\}$	4000	2000	2000
1	$\{3\}$	4000	3000	1000
1	$\{2, 3\}$	6000	3000	3000
2	\emptyset	2000	0	2000
2	$\{1\}$	4000	1000	3000
2	$\{3\}$	3000	3000	0
2	$\{1, 3\}$	6000	4000	2000
3	\emptyset	3000	0	3000
3	$\{1\}$	4000	1000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



COOPERATIVE GAMES - INTERACTIONS



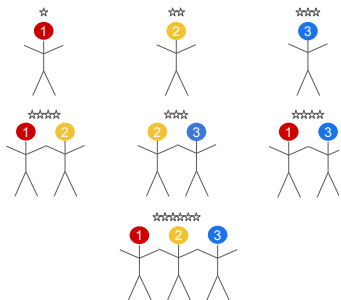
Players interact
(payouts ☆ do not add up)



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

- **With interactions:** Players contribute differently depending on coalition
 ~> Marginal contribs vary across coalitions S (e.g. overlap, synergy)
- Averaging over subsets does not recover total payout $v(P)$
 ~> unfair payout distribution
 ~> avg. contrib. 1 = 1750 2 = 1750 3 = 2250 don't sum to $v(P) = 6000$
- Value a player adds depends on joining order, not just who's in coalition
 ~> Shapley values fairly average over all possible joining orders

COOPERATIVE GAMES - INTERACTIONS



Ordering 1: ③ → ② → ①

③ joins alone: 3 ☆

② joins: total = 3 ☆, marginal = 0

① joins: total = 6 ☆, marginal = +3

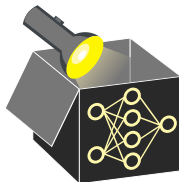
But what if ① joins before ②?

Ordering 2: ③ → ① → ②

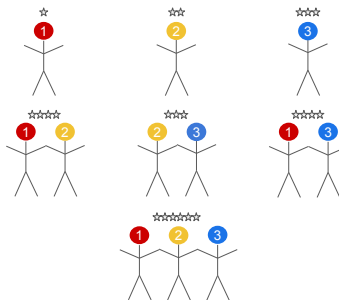
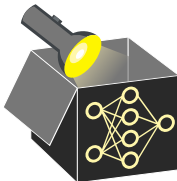
③ joins alone: 3 ☆

① joins: total = 4 ☆, marginal = +1

② joins: total = 6 ☆, marginal = +2



COOPERATIVE GAMES - INTERACTIONS



Ordering 1: ③ → ② → ①

③ joins alone: 3 ☆

② joins: total = 3 ☆, marginal = 0

① joins: total = 6 ☆, marginal = +3

But what if ① joins before ②?

Ordering 2: ③ → ① → ②

③ joins alone: 3 ☆

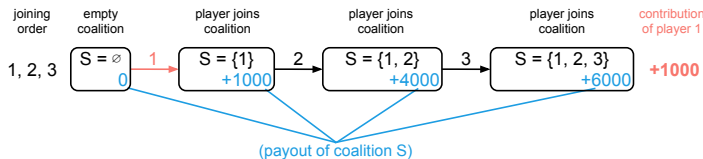
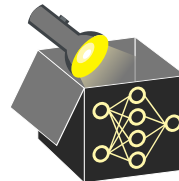
① joins: total = 4 ☆, marginal = +1

② 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., ③'s contribution/skill overlaps with ②'s)
 - ↪ Accounts for order sensitivity (e.g., ① brings more value if added last)
 - ↪ Ensures fairness (order of joining gives no advantage/disadvantage)

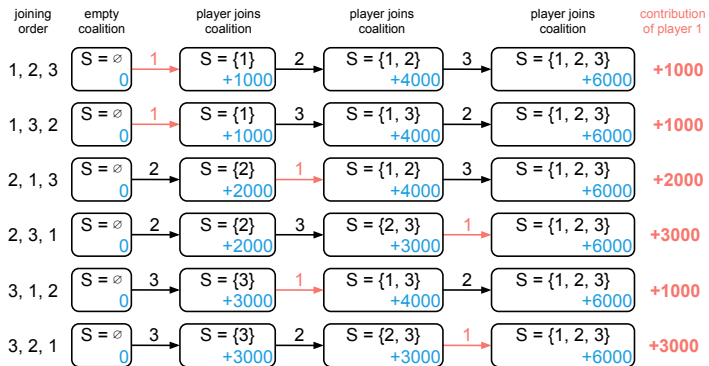
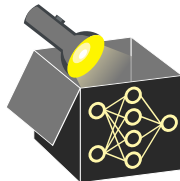
SHAPLEY VALUES - ILLUSTRATION

- 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

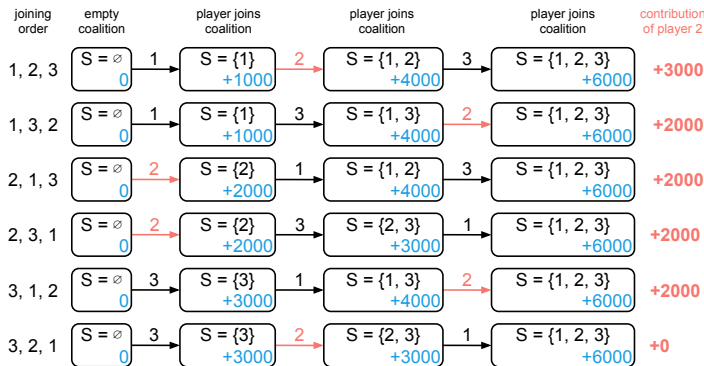
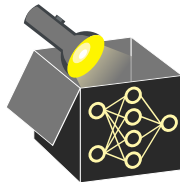
- Generate all possible joining orders (all permutations of full set P)
- For each order: track player j -th marginal contrib when j joins a coalition
- Shapley value of j : Average this marginal contrib over all joining orders
- **Example:** Compute payout diff. after player 1 enters coalition \rightsquigarrow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



Shapley value of player 1 : +1833.33

SHAPLEY VALUES - ILLUSTRATION

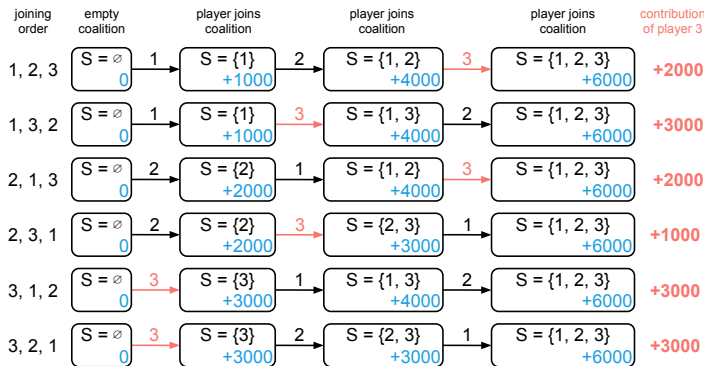
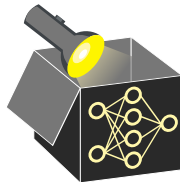
- Generate all possible joining orders (all permutations of full set P)
- For each order: track player j -th marginal contrib when j joins a coalition
- Shapley value of j : Average this marginal contrib over all joining orders
- **Example:** Compute payout diff. after player 2 enters coalition \rightsquigarrow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



Shapley value of player 2 : +1833.33

SHAPLEY VALUES - ILLUSTRATION

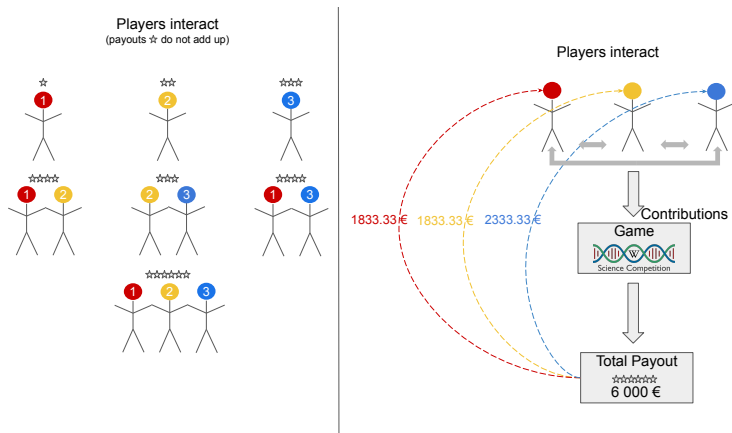
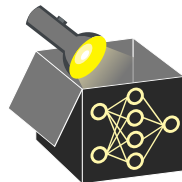
- Generate all possible joining orders (all permutations of full set P)
- For each order: track player j -th marginal contrib when j joins a coalition
- Shapley value of j : Average this marginal contrib over all joining orders
- **Example:** Compute payout diff. after player 3 enters coalition \rightsquigarrow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



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

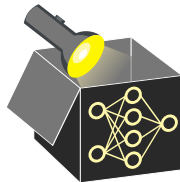


SHAPLEY VALUE - ORDER DEFINITION

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

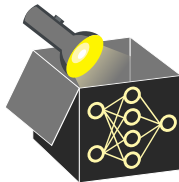


SHAPLEY VALUE - ORDER DEFINITION

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



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.: $\Pi = \{(\textcolor{red}{1}, \textcolor{yellow}{2}, \textcolor{blue}{3}), (\textcolor{red}{1}, \textcolor{blue}{3}, \textcolor{yellow}{2}), (\textcolor{yellow}{2}, \textcolor{red}{1}, \textcolor{blue}{3}), (\textcolor{yellow}{2}, \textcolor{blue}{3}, \textcolor{red}{1}), (\textcolor{blue}{3}, \textcolor{red}{1}, \textcolor{yellow}{2}), (\textcolor{blue}{3}, \textcolor{yellow}{2}, \textcolor{red}{1})\}$

\rightsquigarrow For joining order $\tau = (\textcolor{yellow}{2}, \textcolor{red}{1}, \textcolor{blue}{3})$ and player $j = \textcolor{blue}{3} \Rightarrow S_j^\tau = \{\textcolor{yellow}{2}, \textcolor{red}{1}\}$

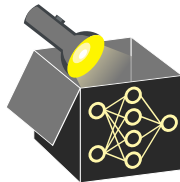
\rightsquigarrow For joining order $\tau = (\textcolor{blue}{3}, \textcolor{red}{1}, \textcolor{yellow}{2})$ and player $j = \textcolor{red}{1} \Rightarrow S_j^\tau = \{\textcolor{blue}{3}\}$

- Order definition allows to approximate Shapley values by sampling permutations

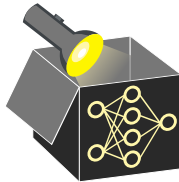
\rightsquigarrow Sample a fixed $M \ll |P|!$ random permutations and average:

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

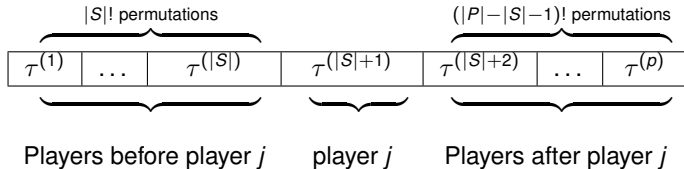
where $\Pi_M \subset \Pi$ is the random sample of M player orderings



FROM ORDER DEFINITION TO SET DEFINITION

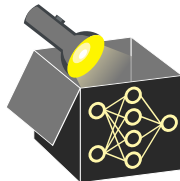
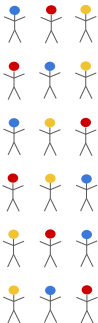


- **Note:** The same subset S_j^T can occur in multiple permutations
 \rightsquigarrow Its marginal contribution is included multiple times in the sum in ϕ_j
- **Example Π (for players $P = \{\textcolor{red}{1}, \textcolor{blue}{2}, \textcolor{blue}{3}\}$, player of interest $j = \textcolor{blue}{3}$):**
 $\{(\textcolor{red}{1}, \textcolor{blue}{2}, \textcolor{blue}{3}), (\textcolor{red}{1}, \textcolor{blue}{3}, \textcolor{blue}{2}), (\textcolor{blue}{2}, \textcolor{red}{1}, \textcolor{blue}{3}), (\textcolor{blue}{2}, \textcolor{blue}{3}, \textcolor{red}{1}), (\textcolor{blue}{3}, \textcolor{red}{1}, \textcolor{blue}{2}), (\textcolor{blue}{3}, \textcolor{blue}{2}, \textcolor{red}{1})\}$
 \rightsquigarrow In $(\textcolor{red}{1}, \textcolor{blue}{2}, \textcolor{blue}{3})$ and $(\textcolor{blue}{2}, \textcolor{red}{1}, \textcolor{blue}{3})$, player $\textcolor{blue}{3}$ joins after coal. $S_j^T = \{\textcolor{red}{1}, \textcolor{blue}{2}\}$
 \Rightarrow Marginal contribution $v(\{\textcolor{red}{1}, \textcolor{blue}{2}, \textcolor{blue}{3}\}) - v(\{\textcolor{red}{1}, \textcolor{blue}{2}\})$ 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



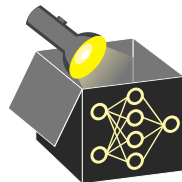
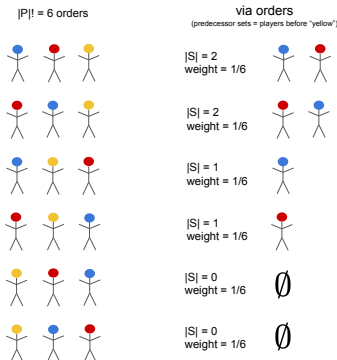
FROM ORDER DEFINITION TO SET DEFINITION

$|P|! = 6$ orders



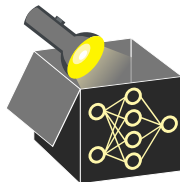
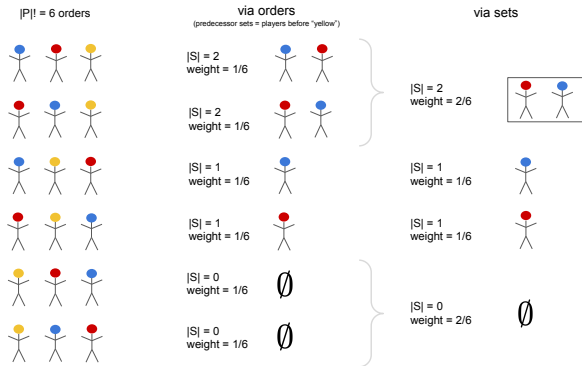
- **Order view:** Each of the $|P|!$ permuts contributes 1 term with weight $\frac{1}{|P|!}$
- Same subset $S \subseteq P \setminus \{j\}$ can appear before j in multiple orders
 \rightsquigarrow e.g., $S = \{\text{blue}, \text{red}\} = \{\text{red}, \text{blue}\}$
- **Set view:** Group by unique subsets 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 $|P|!$ perms 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., $S = \{\text{blue}, \text{red}\} = \{\text{red}, \text{blue}\}$
- **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



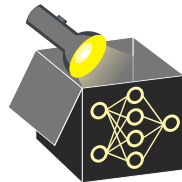
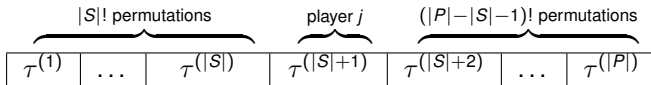
- **Order view:** Each of the $|P|!$ perms contributes 1 term with weight $\frac{1}{|P|!}$
- Same subset $S \subseteq P \setminus \{j\}$ can appear before j in multiple orders
 \rightsquigarrow e.g., $S = \{\text{Blue}, \text{Red}\} = \{\text{Red}, \text{Blue}\}$
- **Set view:** Group by unique subsets 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):

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

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.

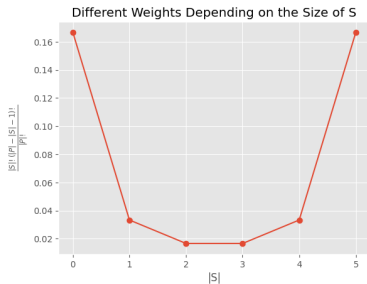
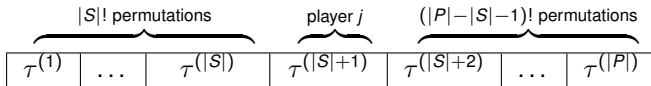


SHAPLEY VALUE - SET DEFINITION

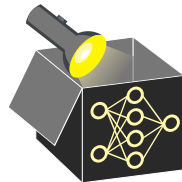
Shapley value via **set definition** (weighting via multinomial coefficient):

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

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.



- $|S| = 0$: 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

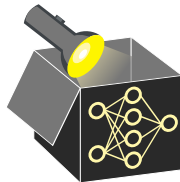


AXIOMS OF FAIR PAYOUTS

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

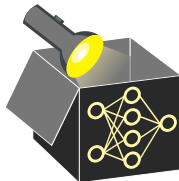
What makes a payout fair? The Shapley value provides a fair payout ϕ_j for each player $j \in P$ and uniquely satisfies the following axioms for any value function v :

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

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

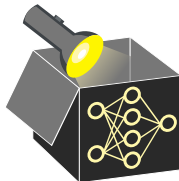
- **Symmetry:** Indistinguishable players $j, k \in P$ receive equal shares:

If $v(S \cup \{j\}) = v(S \cup \{k\})$ for all $S \subseteq P \setminus \{j, k\}$, 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:

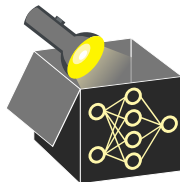
If $v(S \cup \{j\}) = v(S \cup \{k\})$ for all $S \subseteq P \setminus \{j, k\}$, then $\phi_j = \phi_k$

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

If $v(S \cup \{j\}) = v(S)$ for all $S \subseteq P \setminus \{j\}$, 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:

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

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

If $v(S \cup \{j\}) = v(S)$ for all $S \subseteq P \setminus \{j\}$, 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:

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

↪ Payout of combined game = payout of the two separate games