The Web Conference 2023

University of Science and Technology of China

MassNE: Exploring Higher-Order Interactions with Marginal Effect for Massive Battle Outcome Prediction

Yin Gu[1], Kai Zhang[1], Qi Liu[1,2,*], Xin Lin[1], Zhenya Huang[1], Enhong Chen[1]
[1]Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China
& State Key Laboratory of Cognitive Intelligence, Hefei, China,
{gy128,sa517494,linx}@mail.ustc.edu.cn,{qiliuql,huangzhy,cheneh}@ustc.edu.cn

Presented by : Yin Gu

Outline

- Background and Challenges
- Related Work
- Problem Definition
- MassNE Model
- Experiments
- Conclusion and Future Work

Background

➢ **Large-scale battle are ubiquitous in online strategy games**

- two teams fight each other on the battlefield
- different unit types of varying numbers are involved

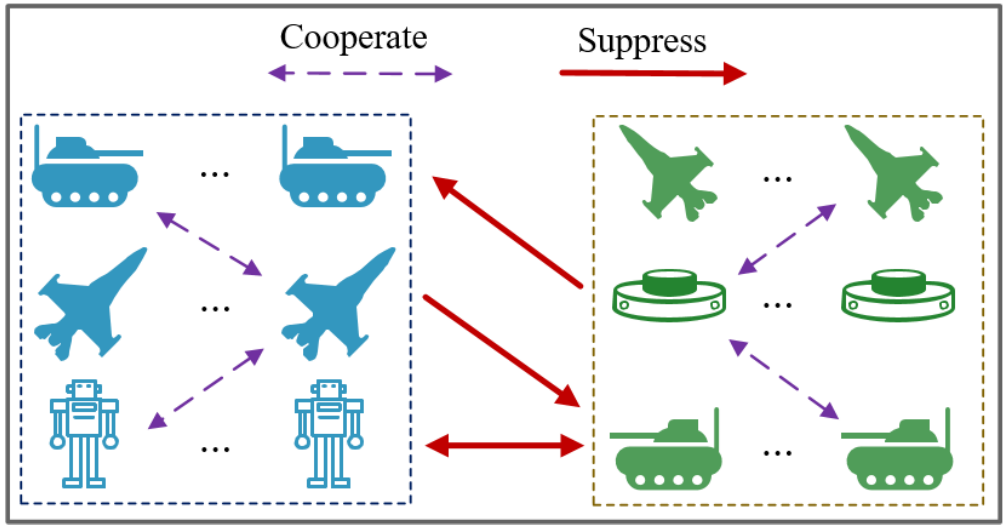
Starcraft II

EVE online

Mount & Blade II

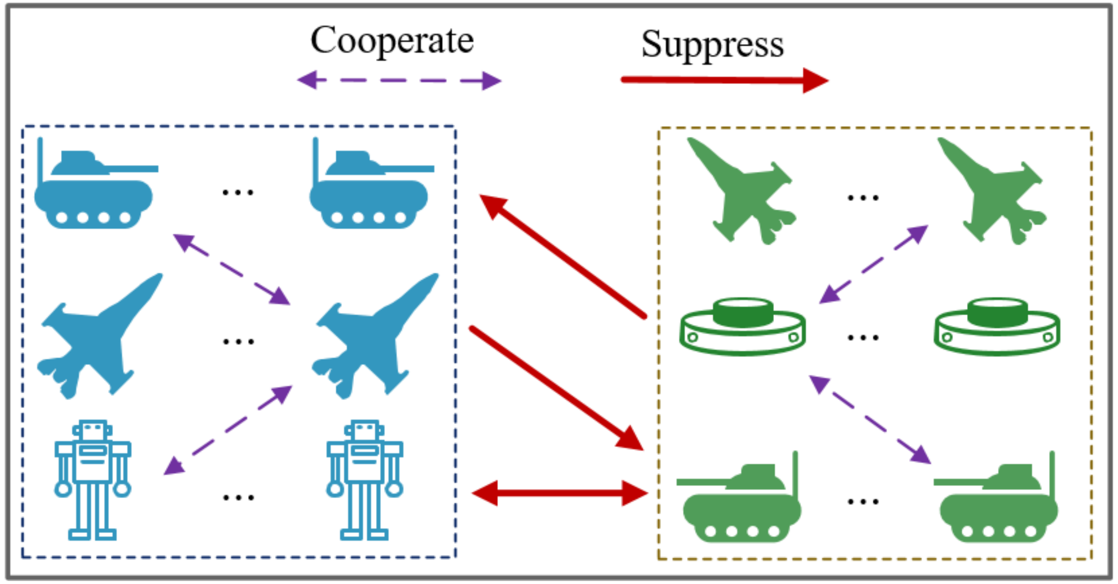
Background

- **Predicting the outcome of massive battle is a fundamental task**
  - Provide a valuable reference for optimizing team configuration before battle
  - For decision-making (e.g., attack or retreat) in battle
  - Game balance analysis



Challenges 1

- How to model the higher-order interactions between **massive** units？
  - ✓ Tanks can cover ground units from enemy fire (i.e., cooperate), but a few tanks can't cover hundreds of units.
  - ✓ One landmine can incapacitate a tank (i.e., suppress), but a few landmine cannot incapacitate an entire tank squad.



Challenges 2

- **Marginal effect in massive battle:**
  - A team has no air force, and we constantly add fighters to it

First Fighter | Second Fighter | Third Fighter | The 100th Fighter

Marginal Utility: 100 | 95 | 92 | … | 10

- How to consider such diminishing marginal utility pattern?

Related Work

- **Methods that consider individual abilities**
  - TrueSkill, Generalized Bradly-Terry (BT)
  - Omit the interaction between individuals
- **Models that implicitly model interactions**
  - Optmatch, DraftRec, BattleNet
  - utilize neural networks to obtain team representations for prediction
- **Methods that explicitly models pairwise interactions between individuals**
  - NeuralAC, HOI
  - Pairwise interactions are unsuitable for massive unit
- **Hand-craft model:**
  - Life Time Damage based model, Lanchester’s Square Law based model
  - rely on additional unit feature, such as attack damage, hit point

Problem Definition

- **Preliminary:**

  - Each battle involves two teams T_A and T_B
  - Suppose there are N types of combat unit $\{1, 2, \ldots, N\}$
  - The unit type combinations of a team is a subset of $\{1, 2, \ldots, N\}$
  - q_i $(q_i \geq 1)$ denotes the quantity of unit type i in a given team.

- **Given:**

  - Two teams T_A and T_B

- **Goal:**
  - Predict the probability of T_A beats T_B (win or loss)

Note: *squad* refers to units of the same type in a team.
For instance, a tank squad of T_A represents all the tank units in the T_A.

Model Overview

- We assume each team has an underlying score S indicating team's capability.

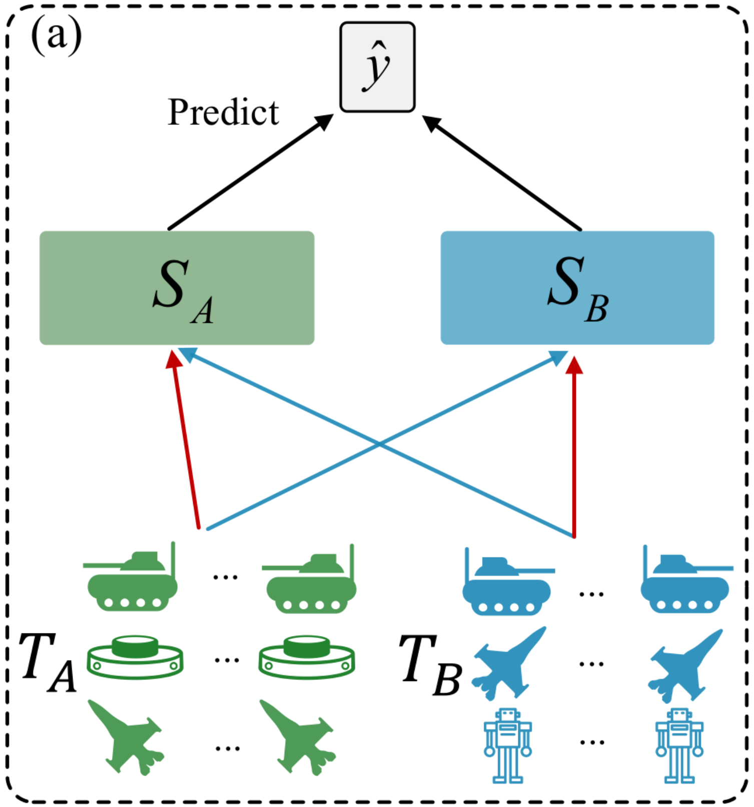
- The probability of team T_A defeating team T_B:

$$P(A \text{ defeats } B) = \frac{\exp(S_A)}{\exp(S_A) + \exp(S_B)}$$

- Overall ability of the team T_A:

$$S_A = F_{\text{indi}}(T_A) + F_{\text{coop}}(T_A) + F_{\text{supp}}(T_A, T_B)$$

  - Individual effects
  - Cooperation effect
  - Suppression effect



Basic MassNE

- The basic version of MassNE without marginal effect modules.
- We treat the units of the same type (i.e., squad) as a whole
  - preserving the characteristic of each unit
  - reducing interaction complexity.

- Individual Effects:
  - The ability of a individual squad that is independent of the other squad.
  - individual ability w_i, squad i's quantity q_i:

$$F_{\text{indi}}(T_A) = \sum_{i \in T_A} q_i \cdot w_i$$

Basic MassNE

- If two squads perform well when they team up together, the cooperation ability between them should be high.
- Cooperation Effects:
  - Cooperation vector $\mathbf{v}_i$ (represents i's cooperation characteristics)

$$F_{\text{coop}}(T_A) = \sum_{i \in T_A} \sum_{j \in T_A, i \neq j} q_i \cdot q_j \cdot f_1(\mathbf{v}_i \odot \mathbf{v}_j)$$

  - f_1 is an interaction function, which output the cooperation ability (a scalar).
  - An increase in the number of either squad will increase the cooperation score between them.

Basic MassNE

- If squad i can suppress squad j (e.g., landmines counter tanks), the corresponding suppression ability of i against j should be high.
- Suppression Effects:
  - Strength vector $\mathbf{p}_i$, weakness vector $\mathbf{c}_i$ (each unit type has its strengths and weaknesses)

$$F_{\text{supp}}(T_A, T_B) = \sum_{i \in T_A} \sum_{j \in T_B} q_i \cdot q_j \cdot f_2(\mathbf{p}_i \odot \mathbf{c}_j)$$

  - An increase in the number of attackers i increases the suppression score.
  - The increase in the number of defenders will also increase the suppression score.

MassNE with Marginal Effect Modules

- The marginal effect should not be ignored in teamwork, especially when the team size is large. For example:
  - when teammates cooperate, if two tanks can cover a teammate, then a third tank will not contribute much.
  - When attacking the enemy, if ten cannonballs can destroy a tank, then an 11-th cannonball is possibly wasted.
  - If one missile can destroy one battle robot, we only have five missiles while the enemy has ten battle robots, then the increase of enemy robots (i.e., defender) will not give us more advantage.

MassNE with Marginal Effect Modules

- E denotes a marginal effect module, whose input is the number of a squad and outputs the squad's utility (a scalar).

- Basic MassNE:

$$S_A = \sum_{i \in T_A} q_i w_i + \sum_{i \in T_A} \sum_{j \in T_A, i \neq j} q_i \cdot q_j \cdot f_1(\mathbf{v}_i \odot \mathbf{v}_j) + \sum_{i \in T_A} \sum_{j \in T_B} q_i \cdot q_j \cdot f_2(\mathbf{p}_i \odot \mathbf{c}_j).$$

- MassNE:

$$S_A = \sum_{i \in T_A} E_i^{in}(q_i) w_i + \sum_{i \in T_A} \sum_{j \in T_A, i \neq j} E_i^{co}(q_i) \cdot E_j^{co}(q_j) \cdot f_1(\mathbf{v}_i \odot \mathbf{v}_j) + \sum_{i \in T_A} \sum_{j \in T_B} E_i^{ak}(q_i) \cdot E_j^{df}(q_j) \cdot f_2(\mathbf{p}_i \odot \mathbf{c}_j).$$

Monotonicity assumption

- *If one unit is added to a team, the overall ability of the team should not decrease.*
- This assumption can be mathematically formulated as:

$$E_i(q_i + 1) \geq E_i(q_i),$$
$$E_i(q_i) \geq 0.$$

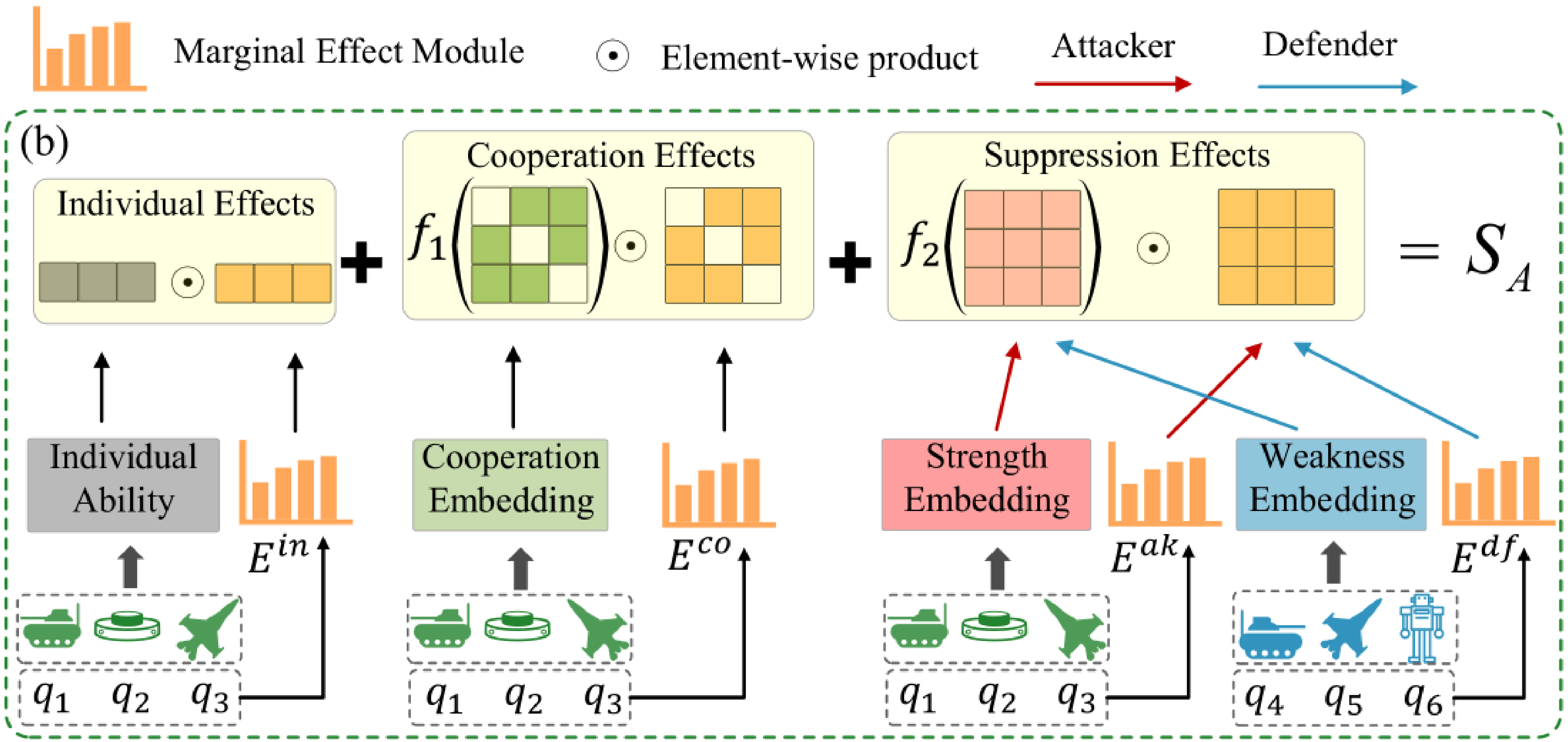
- Monotonicity constraint:

$$\mathcal{L}^m = \sum_{i=0}^{N} \sum_{k=0}^{MQ(i)-1} [E_i(k) - E_i(k+1)] \cdot I\,[E_i(k) > E_i(k+1)]$$

  - Which force $E_i(k+1)$ to be greater than $E_i(k)$

- We implement the E fuctions with a look-up table followed by the ReLU

MassNE framework



Experiments

➢ **Dataset Generation**

- We utilize a famous online strategy game **StarCraft II** to simulate massive battles.
- Two armies will be randomly generated on the battlefield to fight.
- The eliminated army is considered defeated.
- We use three types of terrain as the battlefield: **Plain**, **Corridor** and **Bush**

Table 1: Statistics of the datasets.

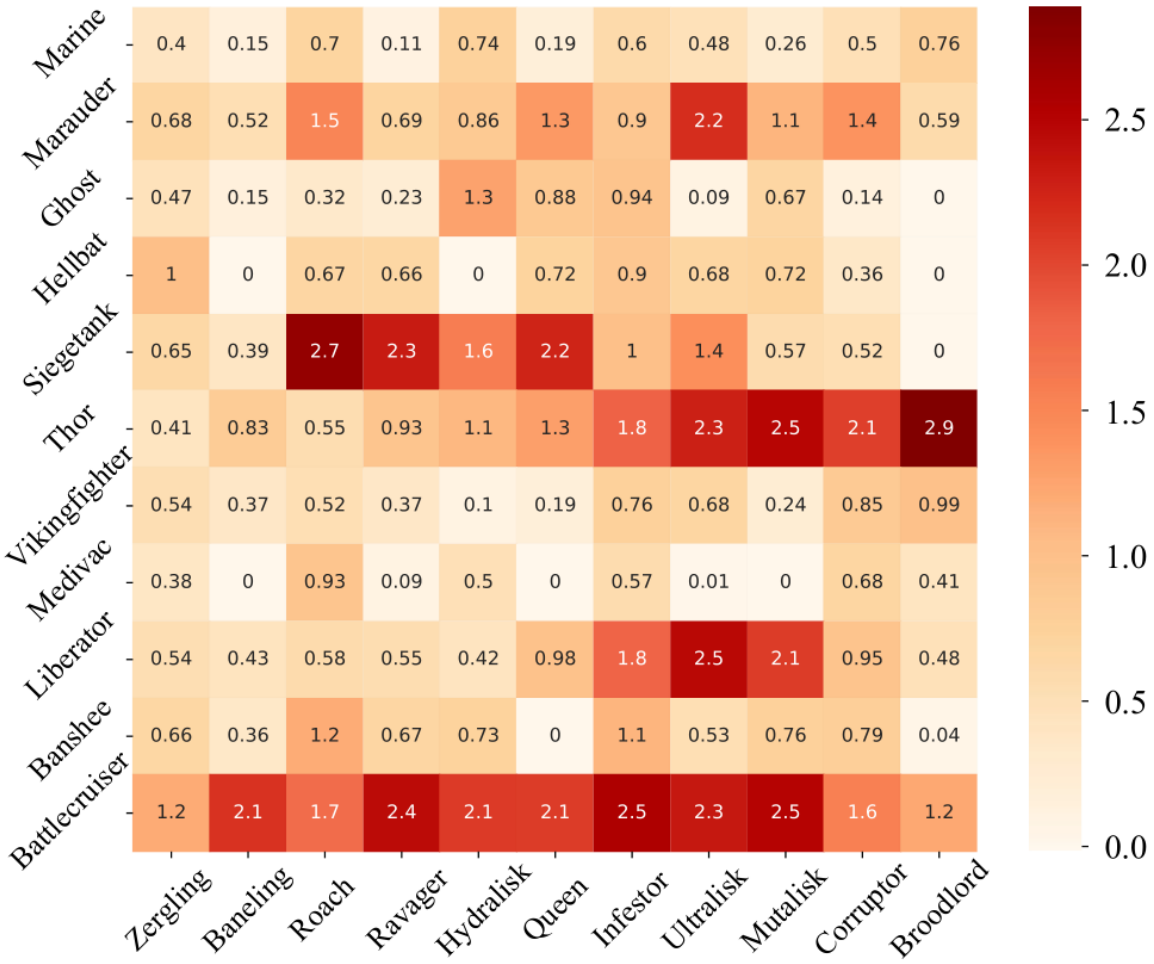
Dataset	**Plain**	**Corridor**	**Bush**
Samples	34,540	33,104	34,494
#Unit types	39	39	39
Avg. team size	64.6	42.7	49.0
Max team size	797	479	461

Experiments

Model	Plain		Corridor		Bush	
	AUC	Acc	AUC	Acc	AUC	Acc
LTD2	0.6868 (0.0055)	0.6792 (0.0063)	0.6567 (0.0061)	0.6476 (0.0046)	0.6687 (0.0047)	0.6589 (0.0044)
TS_Lanchester[2]	0.7551 (0.0074)	0.7276 (0.0061)	0.6683 (0.0042)	0.6567 (0.0052)	0.7436 (0.0072)	0.7159 (0.0066)
BT	0.8931 (0.0036)	0.8159 (0.0014)	0.8646 (0.0055)	0.7843 (0.0062)	0.8887 (0.0061)	0.8033 (0.0073)
LR	0.8999 (0.0028)	0.8207 (0.0043)	0.8653 (0.0072)	0.7881 (0.0069)	0.8891 (0.0071)	0.8021 (0.0034)
TrueSkill	0.8954 (0.0028)	0.8113 (0.0034)	0.8276 (0.0129)	0.7458 (0.0089)	0.8699 (0.0073)	0.7869 (0.0049)
HOI	0.8943 (0.0053)	0.8129 (0.0063)	0.8616 (0.0051)	0.7818 (0.0071)	0.8894 (0.0072)	0.8044 (0.0075)
NeuralAC	0.8976 (0.0037)	0.8194 (0.0074)	0.8632 (0.0049)	0.7852 (0.0052)	0.8905 (0.006)	0.8085 (0.0065)
BattleNet	0.9234 (0.0027)	0.8245 (0.0039)	0.9133 (0.0021)	0.8137 (0.0049)	0.9074 (0.0046)	0.8117 (0.0074)
MassNE	**0.9489** (0.0024)	**0.8802** (0.0038)	**0.9460** (0.0035)	**0.8758** (0.0043)	**0.9412** (0.0021)	**0.8662** (0.0046)

- ML-based methods (BT, LR) are better than hand-craft model
- HOI and NeuralAC which model pairwise interactions between massive units, their performance is close to BT.
- MassNE outperforms the other baselines.

Suppression scores of Terran against Zerg squads

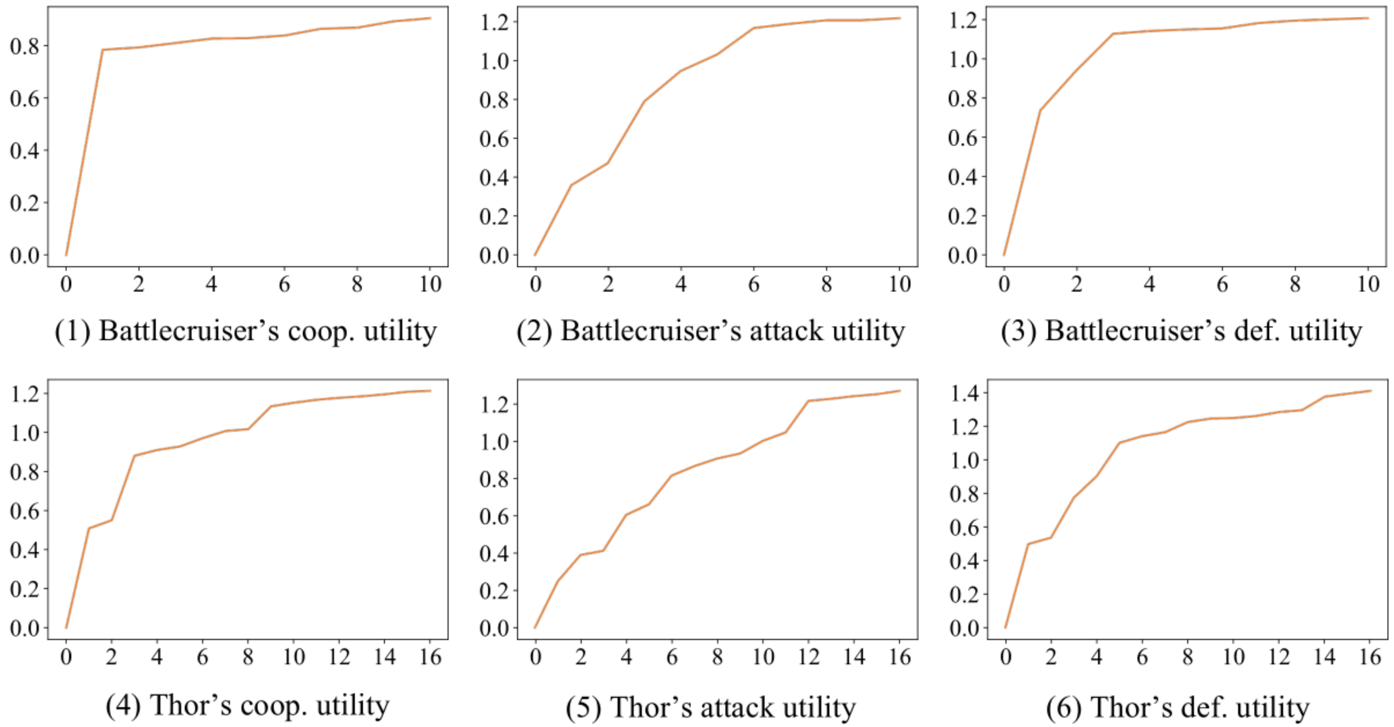


1. We noticed that the Siegetank counters the Zerg ground units (i.e., Roach and Ravager)

2. Tanks cannot attack air units, so tanks get a low score against the enemy air force (e.g., Corruptor and Broodlord)

We set the number of two squads to 10 to obtain these scores.

Diminishing Marginal Utility



The marginal utility will continue to decrease as the number increases.
Recall that marginal utility is $E(k+1) - E(k)$.

Conclusion

- We propose a new model, MassNE, which can effectively model the cooperation effect and restraint effect between arms, while considering the diminishing marginal utility pattern of massive units.

- **Future Work**
  - Consider more complex scenarios in other domains
  - Improve marginal effects modules
  - Team optimization to increase winning rate

Thank you for listening!

Data and codes: https://github.com/firepd/MassNE