float(len(game.get\_legal\_moves(player)))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: ID\_Improved

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: ID\_Improved vs Random Result: 18 to 2

Match 2: ID\_Improved vs MM\_Null Result: 16 to 4

tournament.py:100: UserWarning: One or more agents lost a match this round due to timeout. The get\_move() function must return before time\_left() reaches 0 ms. You will need to leave some time for the function to return, and may need to increase this margin to avoid timeouts during tournament play.

warnings.warn(TIMEOUT\_WARNING)

Match 3: ID\_Improved vs MM\_Open Result: 11 to 9

Match 4: ID\_Improved vs MM\_Improved Result: 6 to 14

Match 5: ID\_Improved vs AB\_Null Result: 13 to 7

Match 6: ID\_Improved vs AB\_Open Result: 12 to 8

Match 7: ID\_Improved vs AB\_Improved Result: 14 to 6

Results:

----------

ID\_Improved 64.29%

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: Student

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: Student vs Random Result: 16 to 4

Match 2: Student vs MM\_Null Result: 16 to 4

Match 3: Student vs MM\_Open Result: 12 to 8

Match 4: Student vs MM\_Improved Result: 9 to 11

Match 5: Student vs AB\_Null Result: 12 to 8

Match 6: Student vs AB\_Open Result: 9 to 11

Match 7: Student vs AB\_Improved Result: 10 to 10

Results:

----------

Student 60.00%

float(len(game.get\_legal\_moves(player))) - len(game.get\_legal\_moves(game.get\_opponent(player)))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: ID\_Improved

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: ID\_Improved vs Random Result: 19 to 1

Match 2: ID\_Improved vs MM\_Null Result: 15 to 5

Match 3: ID\_Improved vs MM\_Open Result: 12 to 8

Match 4: ID\_Improved vs MM\_Improved Result: 8 to 12

Match 5: ID\_Improved vs AB\_Null Result: 9 to 11

Match 6: ID\_Improved vs AB\_Open Result: 8 to 12

Match 7: ID\_Improved vs AB\_Improved Result: 10 to 10

Results:

----------

ID\_Improved 57.86%

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: Student

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: Student vs Random Result: 19 to 1

Match 2: Student vs MM\_Null Result: 12 to 8

Match 3: Student vs MM\_Open Result: 13 to 7

Match 4: Student vs MM\_Improved Result: 11 to 9

Match 5: Student vs AB\_Null Result: 12 to 8

Match 6: Student vs AB\_Open Result: 11 to 9

Match 7: Student vs AB\_Improved Result: 13 to 7

Results:

----------

Student 65.00%

score = 0

division = 0

for move in game.get\_legal\_moves(player):

game\_copy = game.forecast\_move(move)

score += float(len(game\_copy.get\_legal\_moves(player))) - len(game\_copy.get\_legal\_moves(game.get\_opponent(player)))

division += 1

division = (division if division != 0 else 0.001)

return score / division

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: ID\_Improved

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: ID\_Improved vs Random Result: 20 to 0

Match 2: ID\_Improved vs MM\_Null Result: 18 to 2

Match 3: ID\_Improved vs MM\_Open Result: 11 to 9

Match 4: ID\_Improved vs MM\_Improved Result: 6 to 14

Match 5: ID\_Improved vs AB\_Null Result: 12 to 8

Match 6: ID\_Improved vs AB\_Open Result: 13 to 7

Match 7: ID\_Improved vs AB\_Improved Result: 10 to 10

Results:

----------

ID\_Improved 64.29%

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: Student

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: Student vs Random Result: 19 to 1

Match 2: Student vs MM\_Null Result: 18 to 2

Match 3: Student vs MM\_Open Result: 12 to 8

Match 4: Student vs MM\_Improved Result: 9 to 11

Match 5: Student vs AB\_Null Result: 15 to 5

Match 6: Student vs AB\_Open Result: 12 to 8

Match 7: Student vs AB\_Improved Result: 12 to 8

Results:

----------

Student 69.29%

score = 0

division = 0

for move in game.get\_legal\_moves(player):

game\_copy = game.forecast\_move(move)

for second\_move in game\_copy.get\_legal\_moves(game.get\_opponent(player)):

game\_second\_copy = game\_copy.forecast\_move(second\_move)

score += (

float(len(game\_second\_copy.get\_legal\_moves(player))) -

len(game\_second\_copy.get\_legal\_moves(game.get\_opponent(player))))

division += 1

division = (division if division != 0 else 0.001)

return score / division

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: ID\_Improved

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: ID\_Improved vs Random Result: 16 to 4

Match 2: ID\_Improved vs MM\_Null Result: 14 to 6

Match 3: ID\_Improved vs MM\_Open Result: 17 to 3

Match 4: ID\_Improved vs MM\_Improved Result: 11 to 9

Match 5: ID\_Improved vs AB\_Null Result: 12 to 8

Match 6: ID\_Improved vs AB\_Open Result: 11 to 9

tournament.py:100: UserWarning: One or more agents lost a match this round due to timeout. The get\_move() function must return before time\_left() reaches 0 ms. You will need to leave some time for the function to return, and may need to increase this margin to avoid timeouts during tournament play.

warnings.warn(TIMEOUT\_WARNING)

Match 7: ID\_Improved vs AB\_Improved Result: 11 to 9

Results:

----------

ID\_Improved 65.71%

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: Student

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: Student vs Random Result: 16 to 4

Match 2: Student vs MM\_Null Result: 14 to 6

Match 3: Student vs MM\_Open Result: 9 to 11

Match 4: Student vs MM\_Improved Result: 6 to 14

Match 5: Student vs AB\_Null Result: 6 to 14

Match 6: Student vs AB\_Open Result: 9 to 11

Match 7: Student vs AB\_Improved Result: 8 to 12

Results:

----------

Student 48.57%

score = 0

for move in game.get\_legal\_moves(player):

game\_copy = game.forecast\_move(move)

score += float(len(game\_copy.get\_legal\_moves(player))) - len(game\_copy.get\_legal\_moves(game.get\_opponent(player)))

division = (len(game.get\_legal\_moves(player)) if len(game.get\_legal\_moves(player)) != 0 else 0.001)

return score / division

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: ID\_Improved

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: ID\_Improved vs Random Result: 17 to 3

Match 2: ID\_Improved vs MM\_Null Result: 18 to 2

Match 3: ID\_Improved vs MM\_Open Result: 12 to 8

Match 4: ID\_Improved vs MM\_Improved Result: 9 to 11

Match 5: ID\_Improved vs AB\_Null Result: 13 to 7

Match 6: ID\_Improved vs AB\_Open Result: 13 to 7

Match 7: ID\_Improved vs AB\_Improved Result: 12 to 8

Results:

----------

ID\_Improved 67.14%

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: Student

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: Student vs Random Result: 19 to 1

Match 2: Student vs MM\_Null Result: 19 to 1

Match 3: Student vs MM\_Open Result: 13 to 7

Match 4: Student vs MM\_Improved Result: 10 to 10

Match 5: Student vs AB\_Null Result: 15 to 5

Match 6: Student vs AB\_Open Result: 16 to 4

Match 7: Student vs AB\_Improved Result: 11 to 9

Results:

----------

Student 73.57%