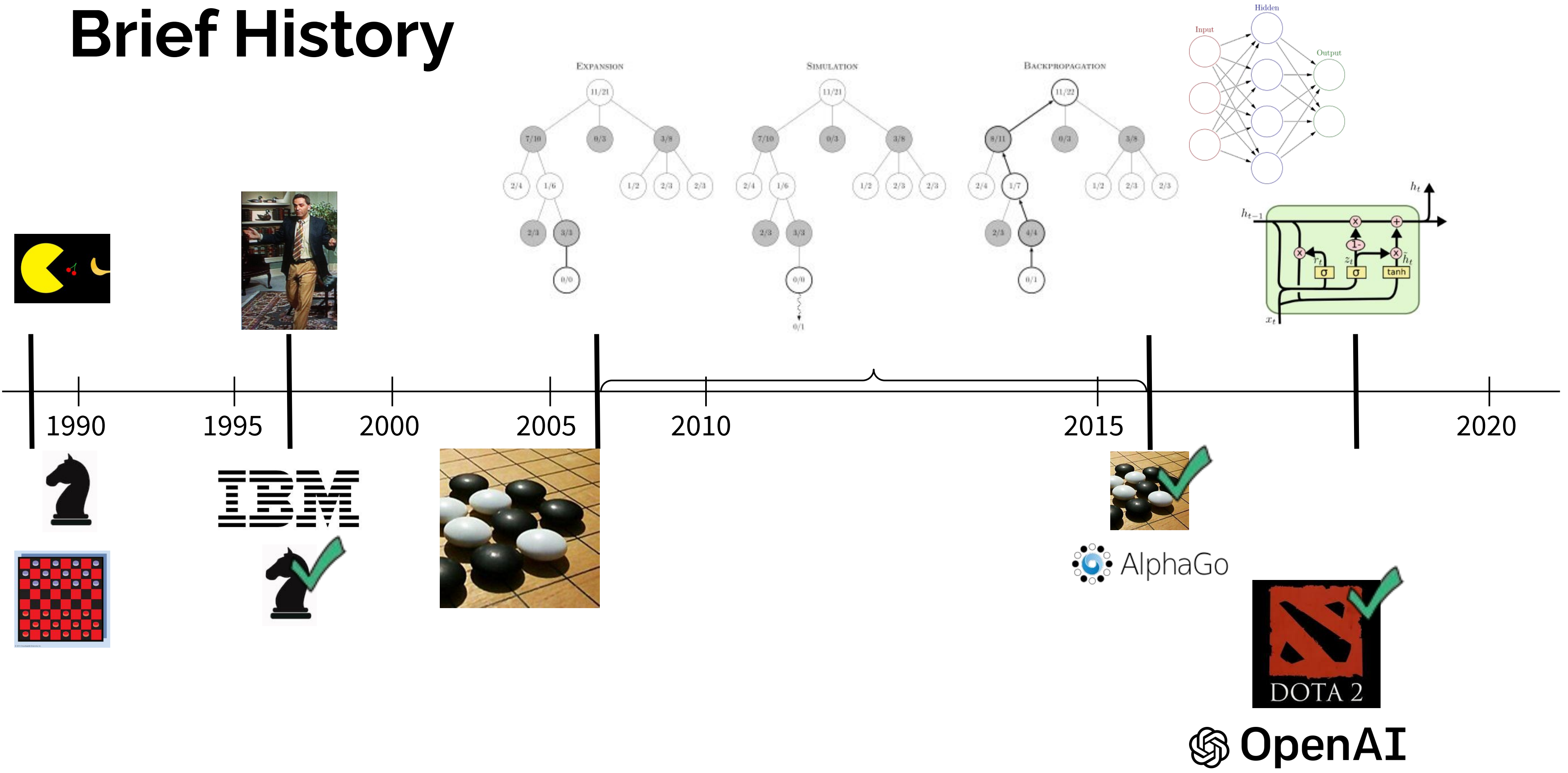
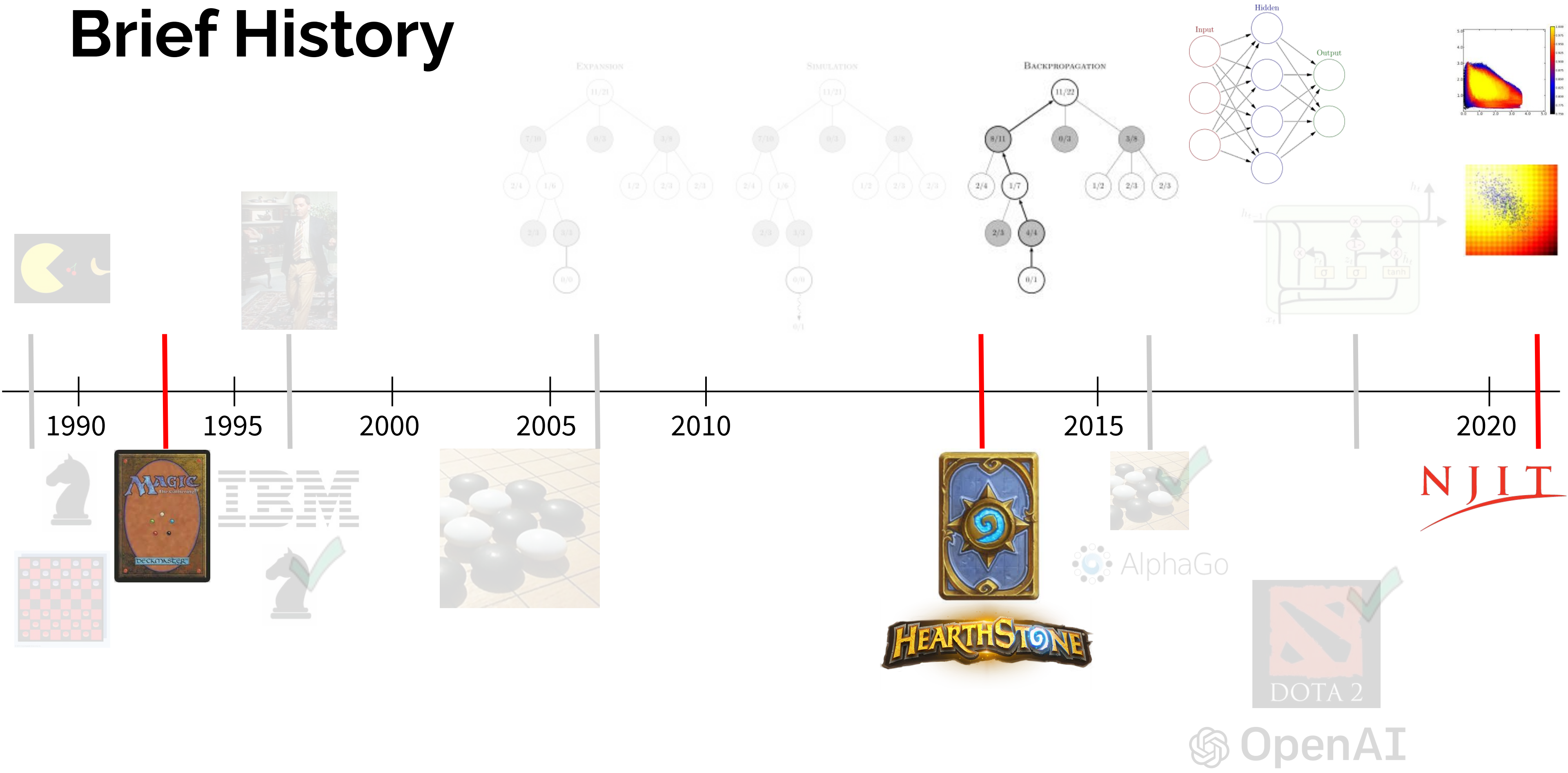
Analysis of Gameplay Strategies in Hearthstone: A Data Science Approach

Connor W. Watson | Dr. Amy K. Hoover | Dr. Usman W. Roshan | Dr. Senjuti Basu Roy

Brief History



Brief History



Why Games?

- Hard to “predict”:
  - Chess has 6 pieces, Hearthstone over 3000
  - Multiple strategies (5) + sub-strategies (100?)
  - Observability
  - Chess (deterministic) vs Hearthstone (stochastic) [1]
- Hearthstone has orders of magnitude larger solution space than Chess/Go
  - Game tree + level of uncertainty grow (MCTS) ✖
  - Brute force is challenging - shown for Backgammon, Checkers, Chess ✖
  - Need a better way to make decisions

Abstract

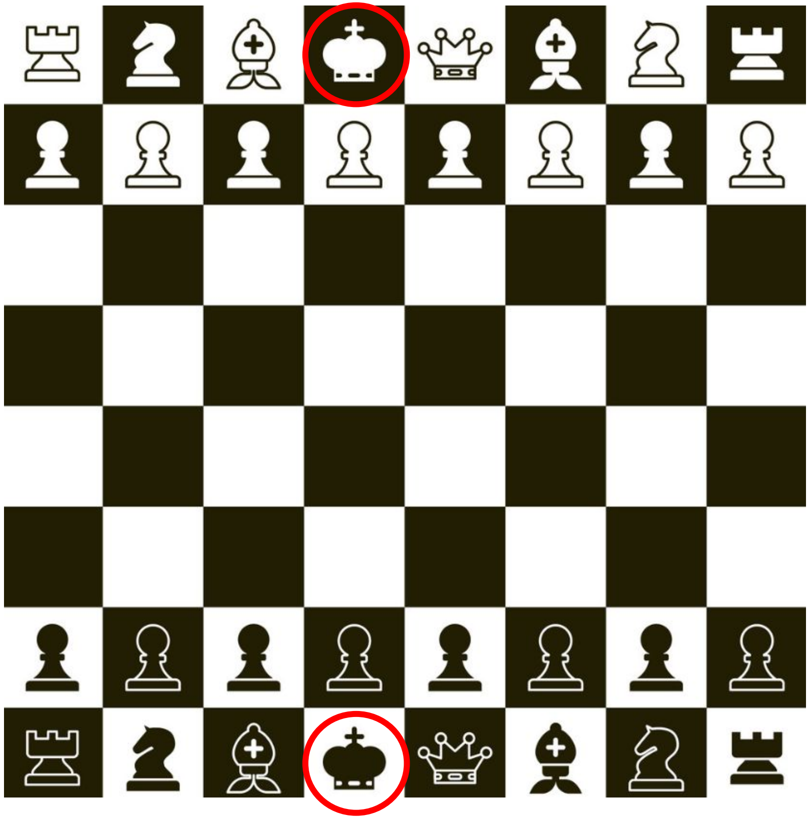
High Level Goals:

1. Compare/contrast & visualize gameplay strategies
2. Use new algorithm to make better players
   a. Can agents make better decisions
3. Predict gameplay strategies

What is ~~Hearthstone~~...Chess?

Remove the pieces…

Leave the King on board



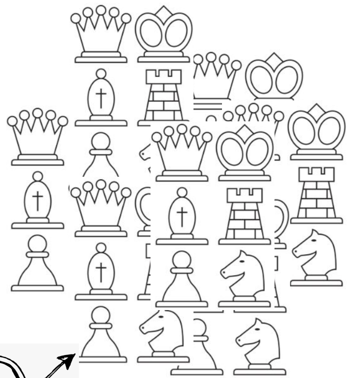
What is ~~Hearthstone~~...Chess?

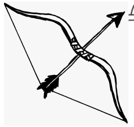
Pick 30

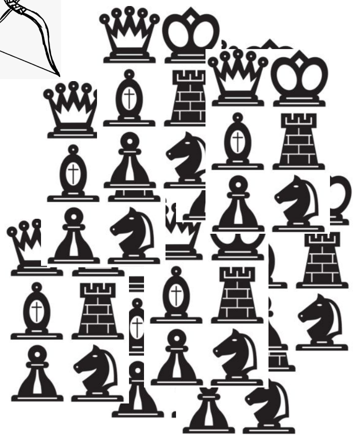
Bring those to games

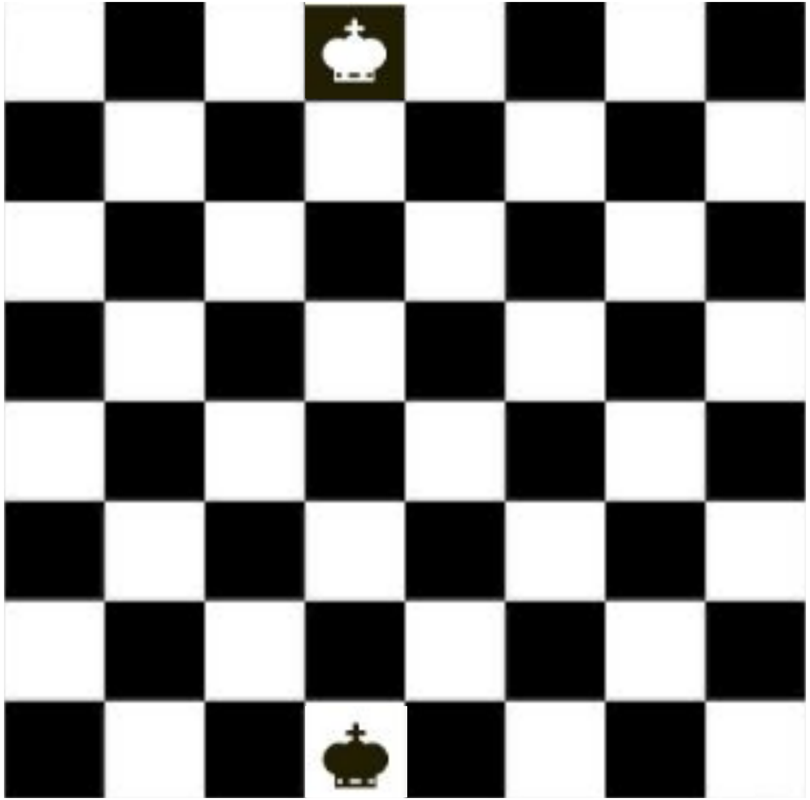
King has abilities

Where are the pieces...







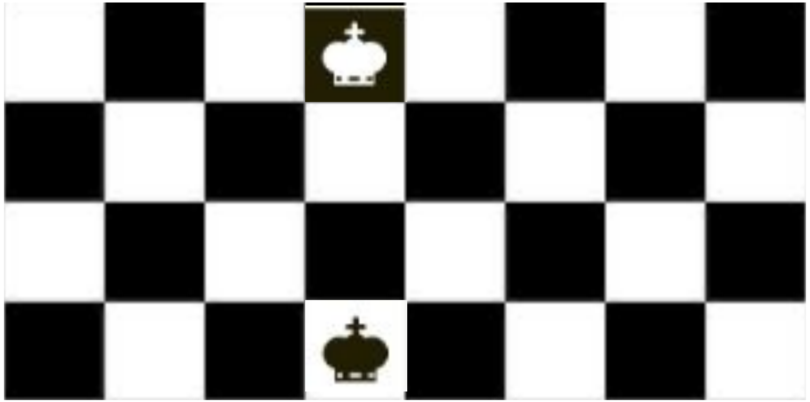


What is ~~Hearthstone~~...Chess?

Hide pieces

Every turn you put some

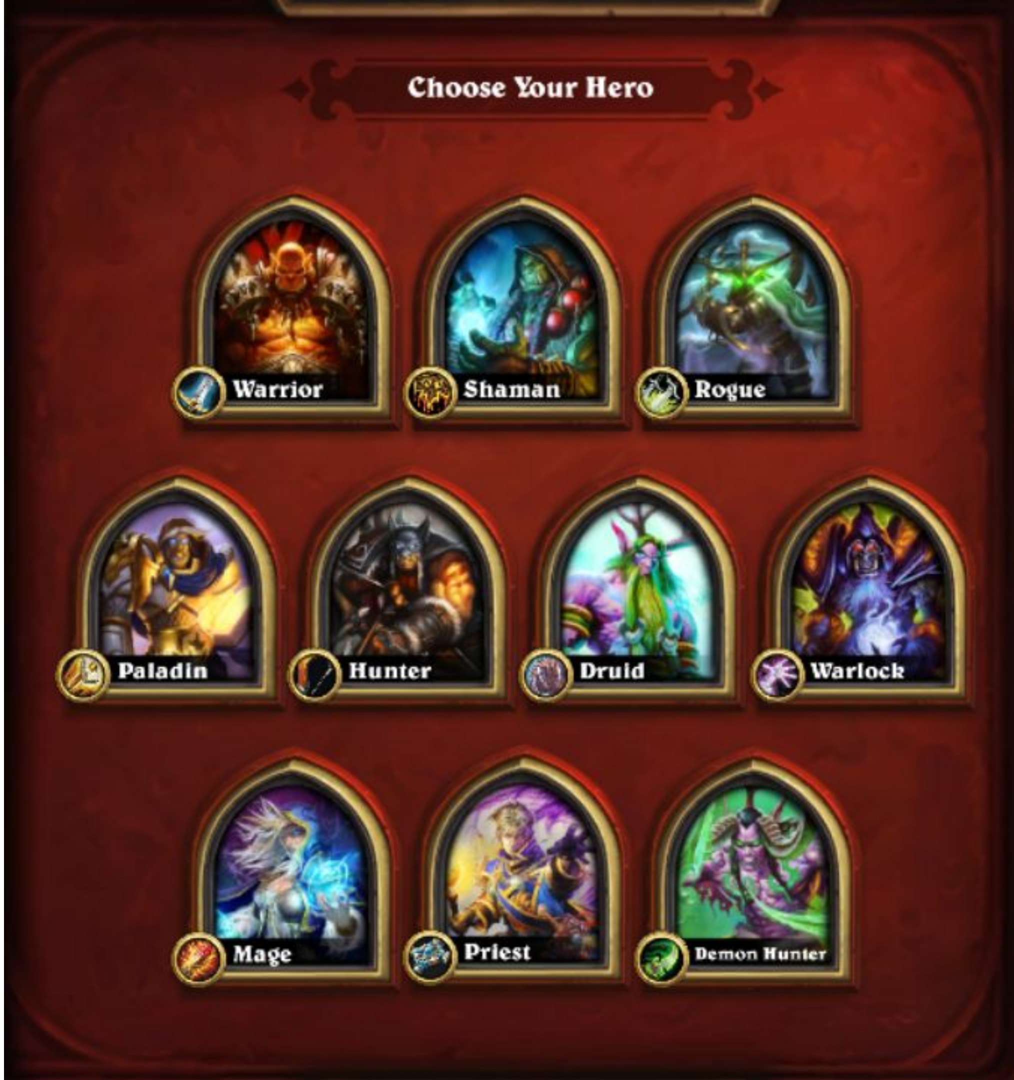
What's the goal?



Goal: hit king 30 times

Hero Classes

- 10 to choose from
- Each has specific cards
- “Basic” cards - shared with all



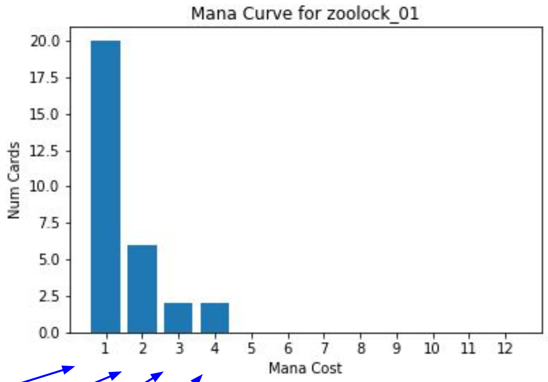
Types of Cards - Minions/Spells

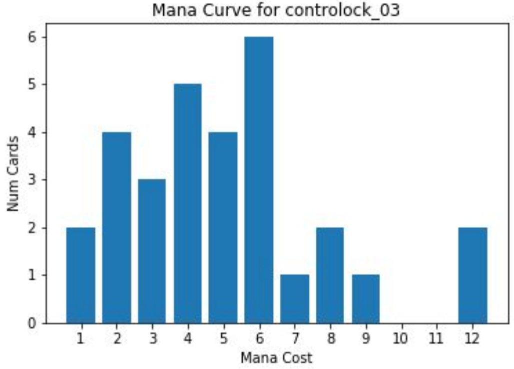
- Mana cost (top left)
- Name (center) /Effect (lower middle)
- Attack power (bottom left)
- Health points (bottom right)
- Played onto the board

- Mana cost (top left)
- Name (center) /Effect (lower middle)
- Played from hand (single use)
  - Not on the board

Deck Building Strategies

- Mana Curve - amount of cards in each bin of mana cost
  - Resource management tied to strategy
- Aggro deck (top)
  - Many low cost cards
- Control deck (bottom)
  - Scattered across all bins



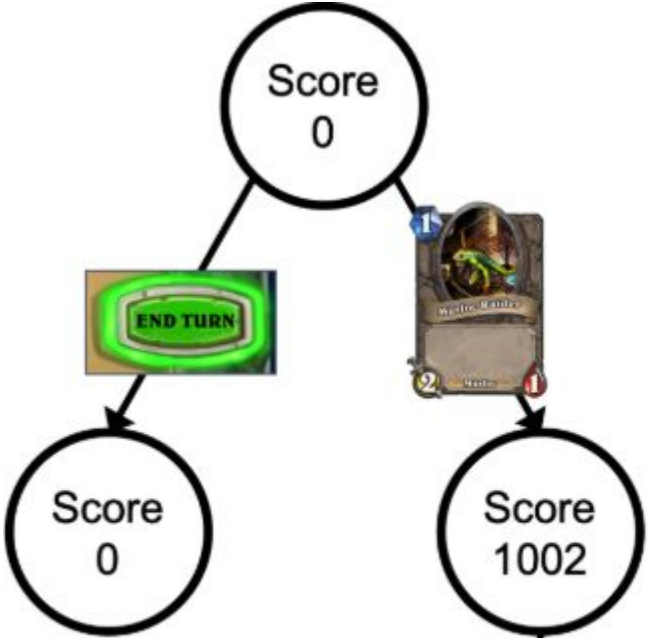


SabberStone: a HS Simulator

- Hearthstone is not open source
- SabberStone is a community repository
  - GitHub: https://github.com/HearthSim/SabberStone
- Comes packaged with AI for research / testing
  - Turn local game tree
  - Heuristic scoring functions

Heuristic Scoring Functions

- AggroScore
  - Rewards attacking opponent
  - End game faster
- ControlScore
  - Rewards board control
  - End game slower
- Can an algorithm evolve better scoring functions?



Heuristic Scoring Functions

```
//ControlScore
public override int Rate()
{
    if (OpHeroHp < 1)
        return int.MaxValue;
    if (HeroHp < 1)
        return int.MinValue;
    int result = 0;
    if (OpBoardZone.Count == 0 && BoardZone.Count > 0)
        result += 1000;

    result += (BoardZone.Count - OpBoardZone.Count) * 50;
    result += (MinionTotHealthTaunt - OpMinionTotHealthTaunt) * 25;
    result += (HeroHp - OpHeroHp) * 10;

    result += MinionTotAtk;
    return result;
}
```

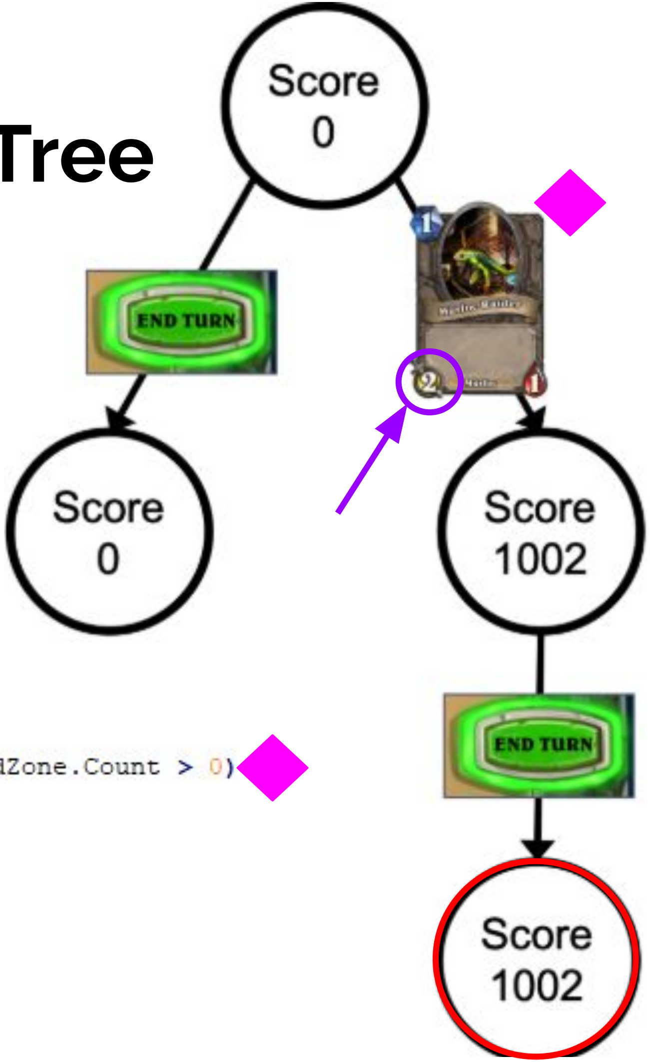
```
//AggroScore
public override int Rate()
{
    if (OpHeroHp < 1)
        return int.MaxValue;
    if (HeroHp < 1)
        return int.MinValue;
    int result = 0;
    if (OpBoardZone.Count == 0 && BoardZone.Count > 0)
        result += 1000;

    if (OpMinionTotHealthTaunt > 0)
        result += OpMinionTotHealthTaunt * -1000;
    result += (HeroHp - OpHeroHp) * 1000;

    result += MinionTotAtk;
    return result;
}
```

Game States

- Starting turn 1 (bottom)
- Both players start with 0
- Gain 1 mana at start of turn
- 1 mana to spend
    - 1 cost cards only

Game States/Tree

- Most positive score
- Best decision



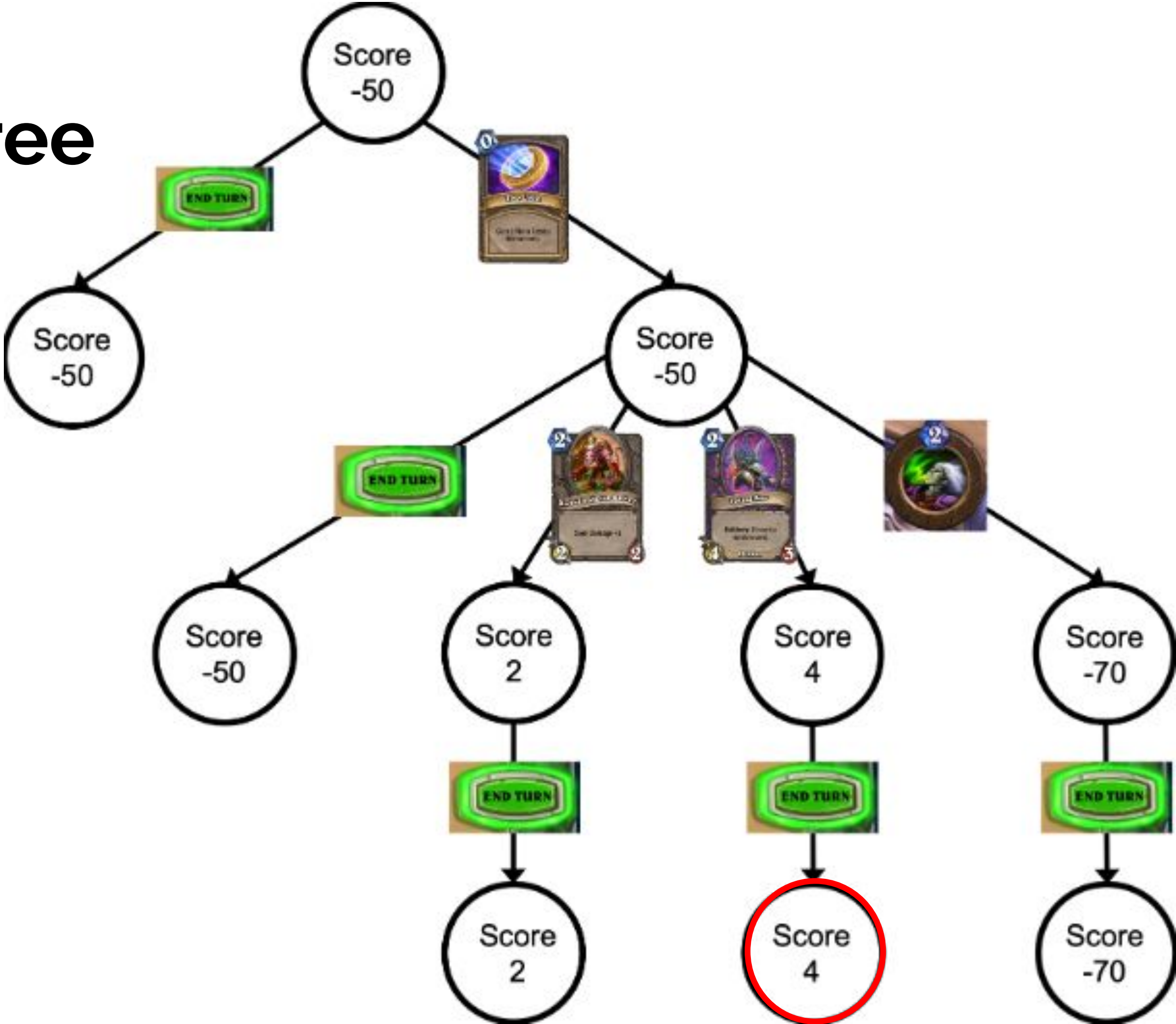

```
int result = 0;
if (OpBoardZone.Count == 0 && BoardZone.Count > 0)
    result += 1000;
•••
result += MinionTotAtk;
return result;
}
```

Game States

- Choice: play minion card
- Maintain board control

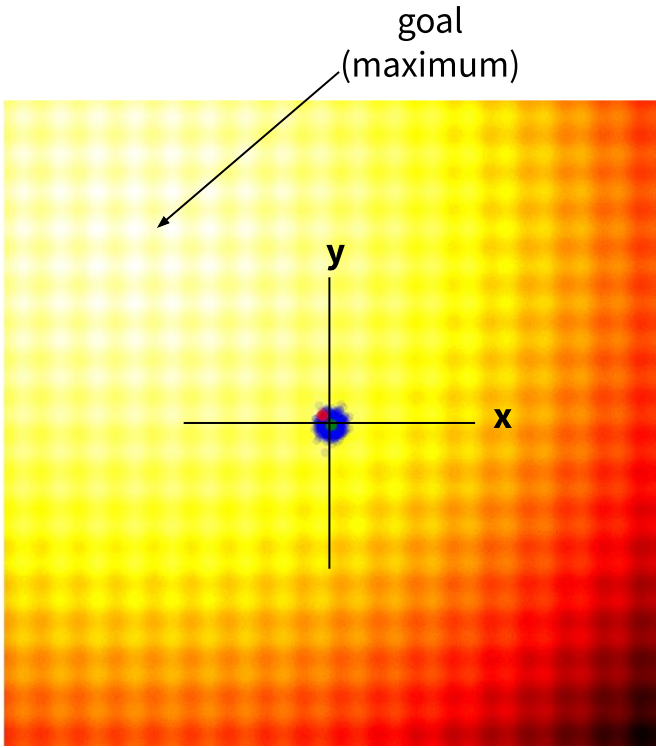
Game States/Tree

- Tree ends at current turn
- No look-ahead
- Start at disadvantage



CMA-ES

- Calculate fitness for generation (solutions)
- Isolate top N %
- Calculate covariance matrix of next generation
  - Uses covariance matrix only
- Sample new solutions from it

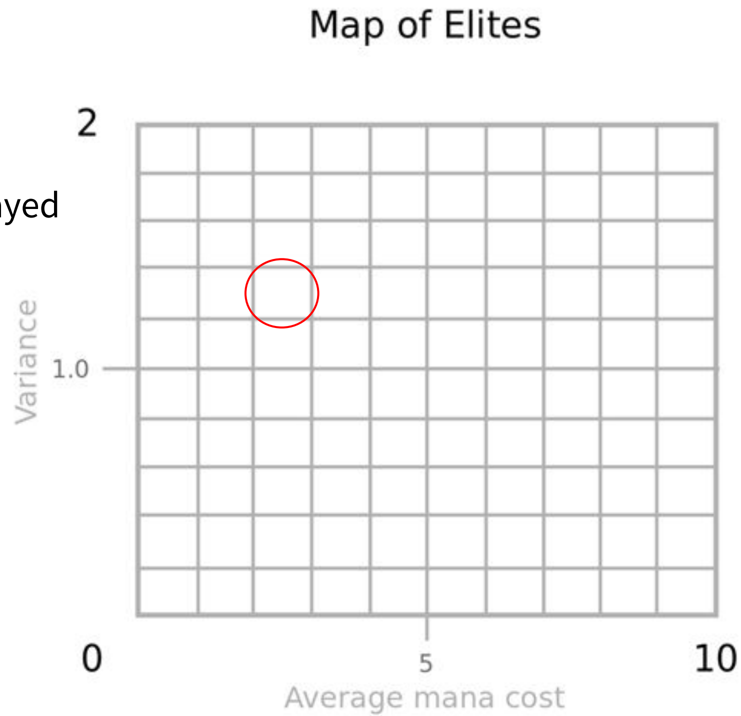


Source [7]

MAP-Elites

- Average mana cost = 2.5
- Variance = 1.25
- Behavior Vector: (2.5, 1.25)
- Fitness (win rate) = 100/200
  - NumGamesWon / NumGamesPlayed

D1: F=100



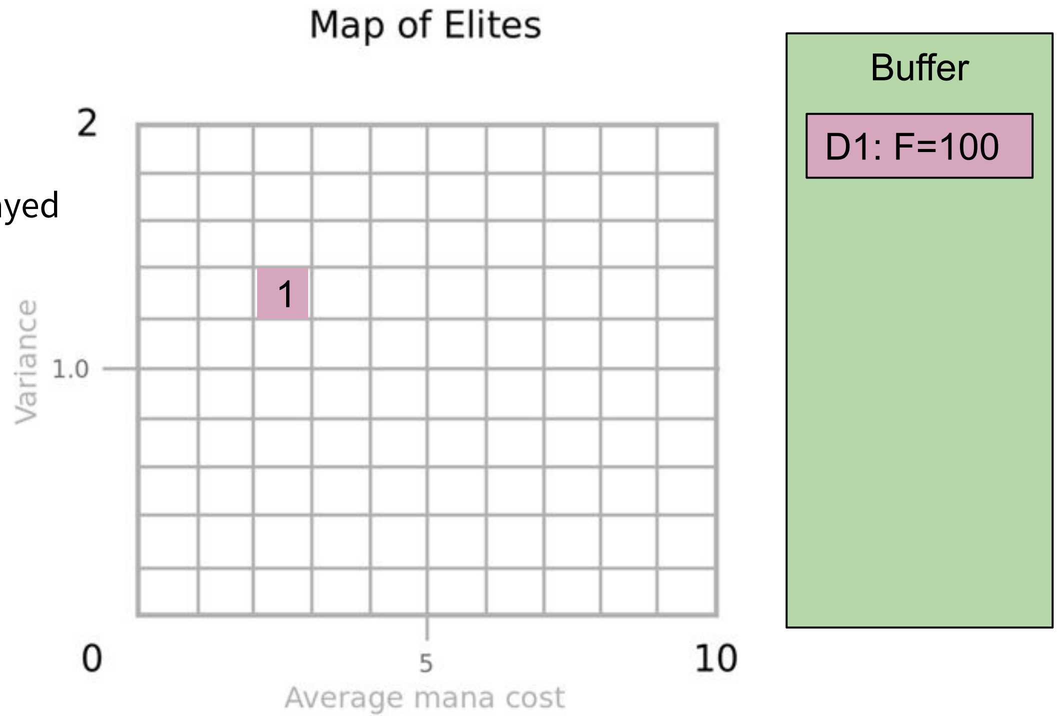
Buffer

D1: F=100

MAP-Elites

- Average mana cost = 2.5
- Variance = 1.25
- Behavior Vector: (2.5, 1.25)
- Fitness (win rate) = 100/200
    - NumGamesWon / NumGamesPlayed

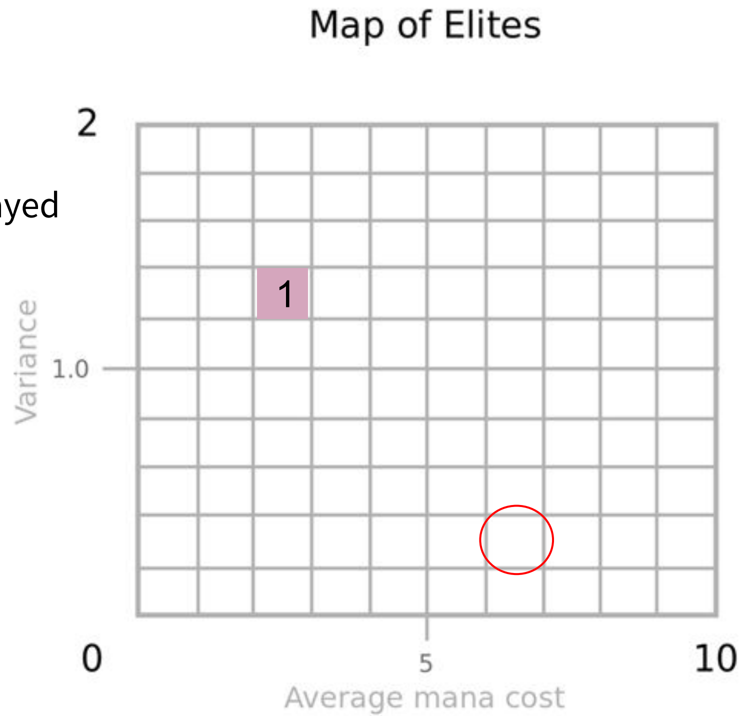
D1: F=100



MAP-Elites

- Average mana cost = 6.5
- Variance = 0.25
- Behavior Vector: (6.5, 0.25)
- Fitness (win rate) = 80/200
  - NumGamesWon / NumGamesPlayed

D2: F=80



Buffer

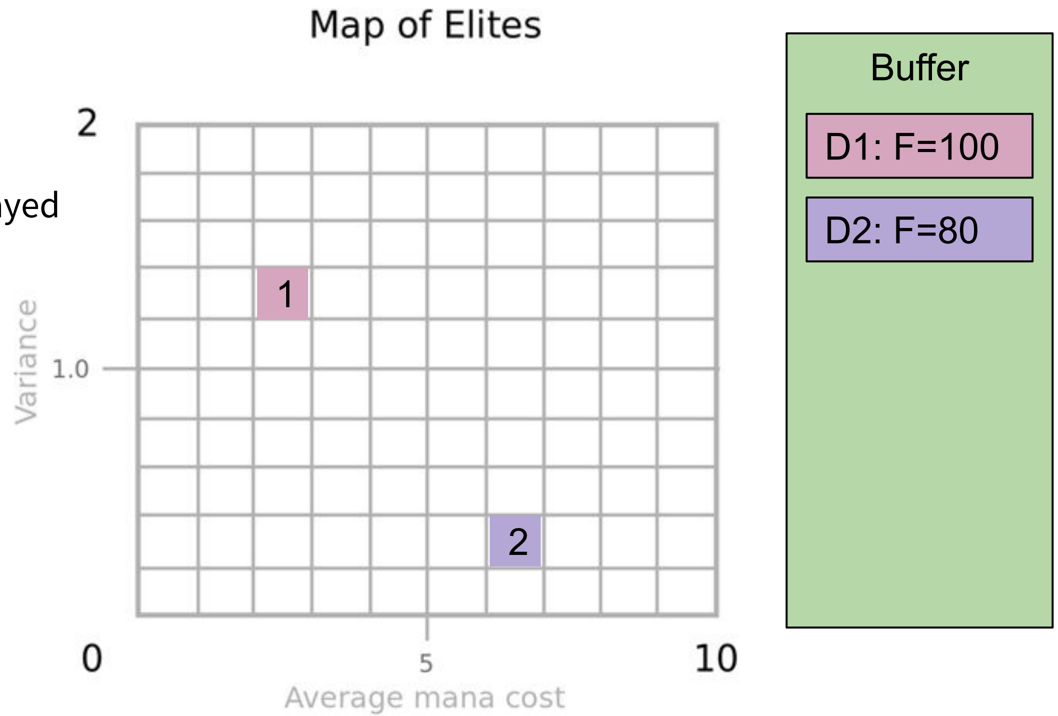
D1: F=100

D2: F=80

MAP-Elites

- Average mana cost = 6.5
- Variance = 0.25
- Behavior Vector: (6.5, 0.25)
- Fitness (win rate) = 80/200
    - NumGamesWon / NumGamesPlayed

D2: F=80

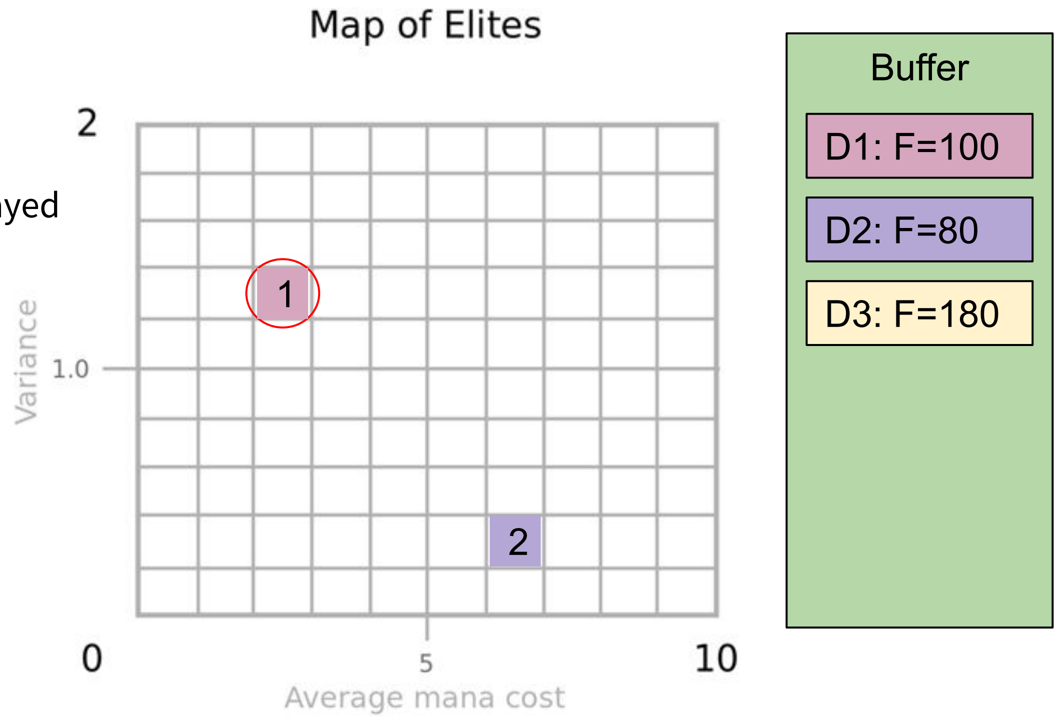


MAP-Elites

- Average mana cost = 2.5
- Variance = 1.25
- Behavior Vector: (2.5, 1.25)
- Fitness (win rate) = 180/200
  - NumGamesWon / NumGamesPlayed

What happens?

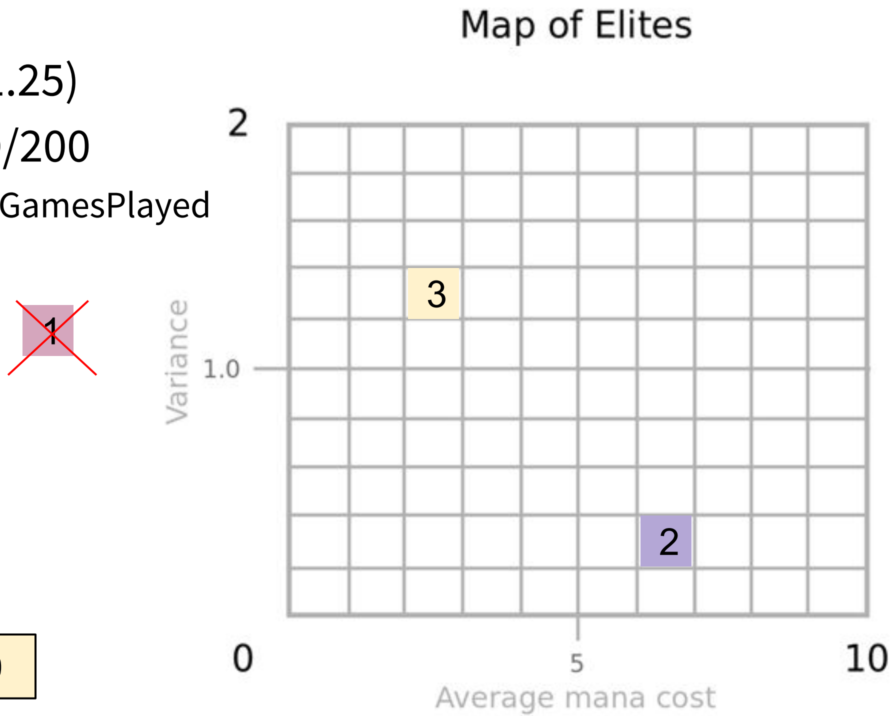
D3: F=180



MAP-Elites

- Average mana cost = 2.5
- Variance = 1.25
- Behavior Vector: (2.5, 1.25)
- Fitness (win rate) = 150/200
    - NumGamesWon / NumGamesPlayed

D3: F=180



Buffer

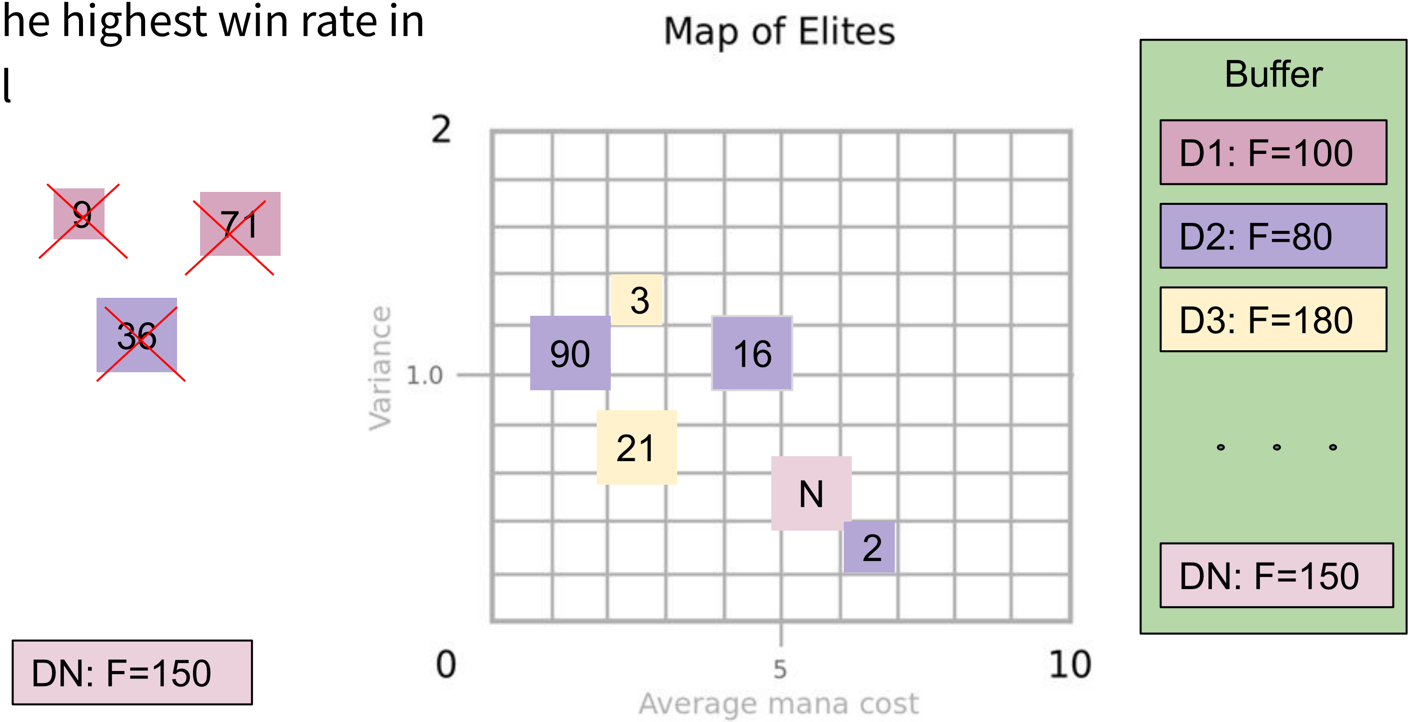
D1: F=100

D2: F=80

D3: F=180

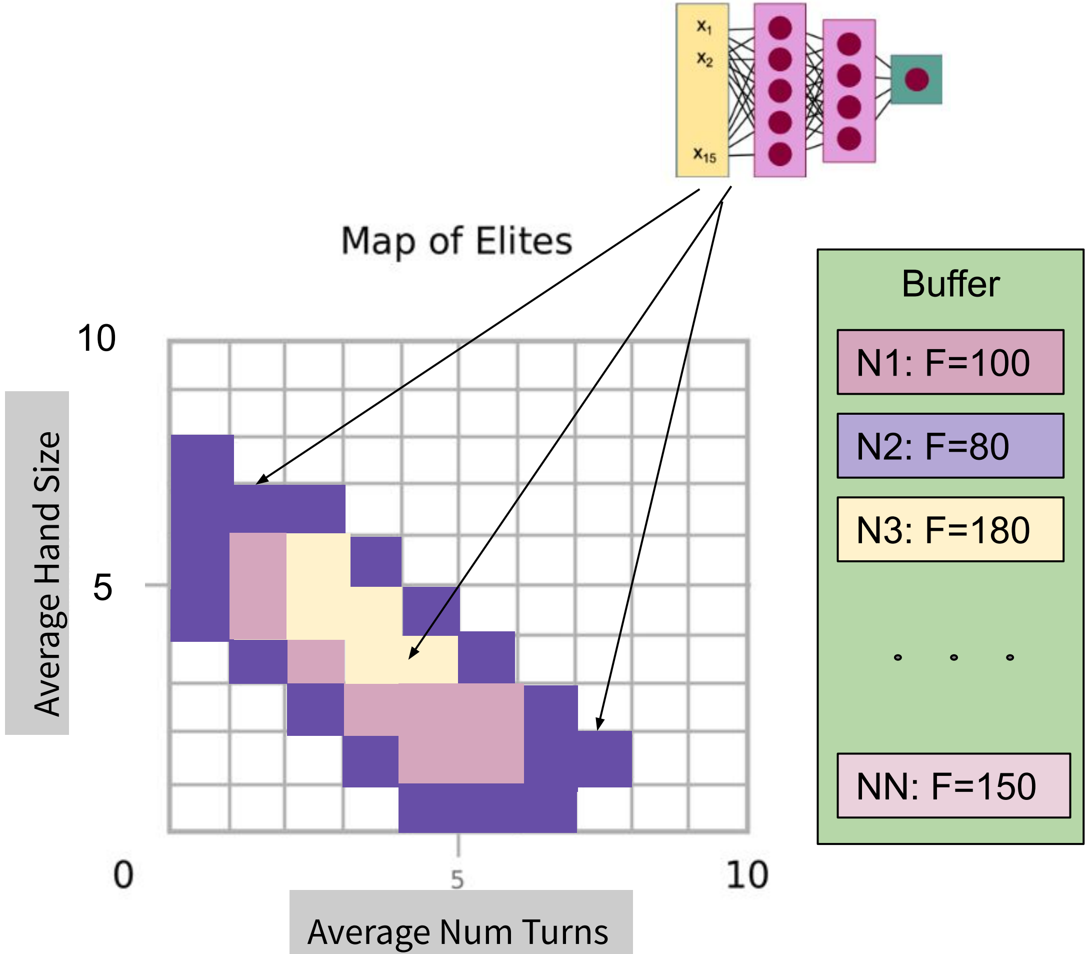
MAP-Elites

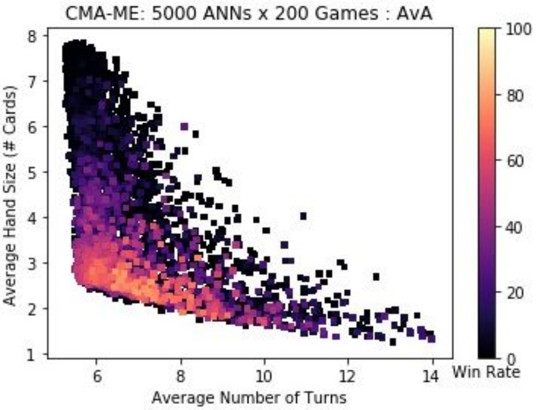
- Keep the “Elite” decks (solutions)
- These have the highest win rate in each grid cell



CMA-ME

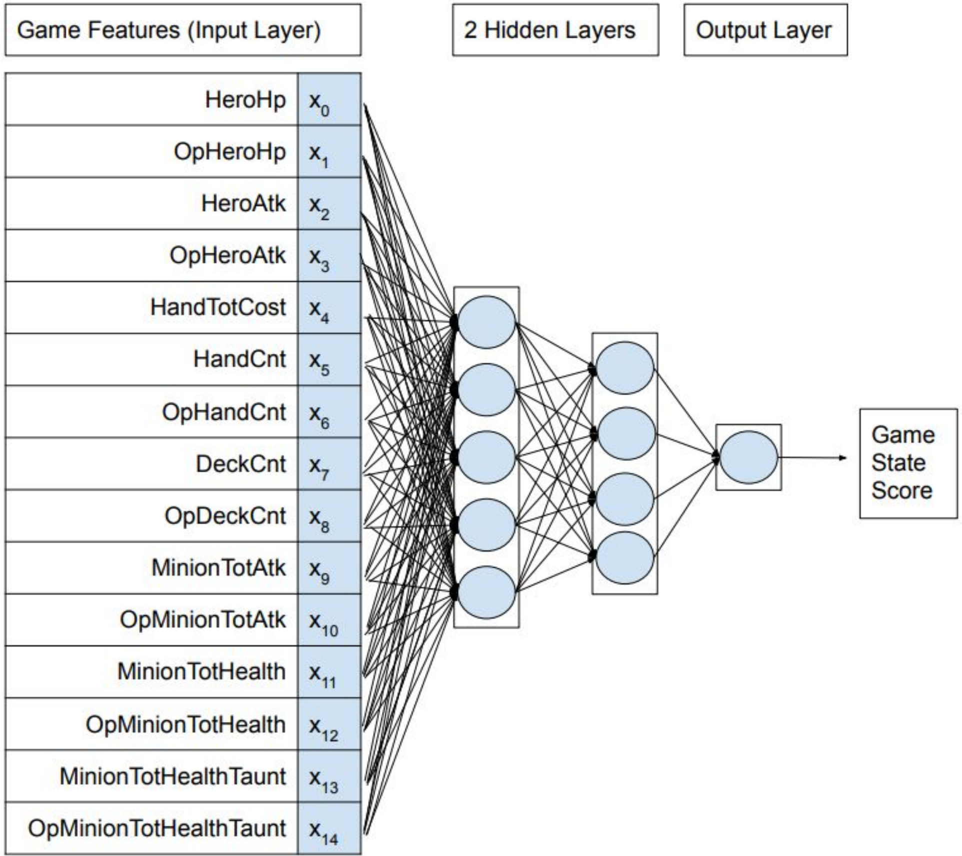
- Use ANNs instead of decks
- Evolve with CMA-ES
- MAP of Elite ANNs
- Play 200 games with 5000 ANNs





Artificial Neural Network

- Fully connected
  - All nodes connect from one layer to next
- Feed forward
  - No backward connections
- Inspired by [8, 9]
- Input:
  - Visible game pieces
- Output:
  - Game state score



Research Questions

1. Is control really better than aggro?
    a. Or is ControlScore a better heuristic
2. Can ANNs perform better?
3. Can gameplay strategies be predicted?
4. Can gameplay strategies be visualized?

The Data

25 total columns...

- Collection of turns (all games)
- Num turns per game, which player won, etc..
- How to reduce this?

TURN_NO	P1_HEALTH	P2_HEALTH	CURRENT_PLAYER	AMOUNTHEALEDTHISTURN
1	30	30	P1 FitzVonGerald	0
2	30	30	P2 RehHausZuckFuchs	0
3	30	29	P1 FitzVonGerald	0
4	27	29	P2 RehHausZuckFuchs	0
5	27	24	P1 FitzVonGerald	0
6	21	24	P2 RehHausZuckFuchs	0
7	21	16	P1 FitzVonGerald	0
8	8	16	P2 RehHausZuckFuchs	0
1	30	30	P1 FitzVonGerald	0

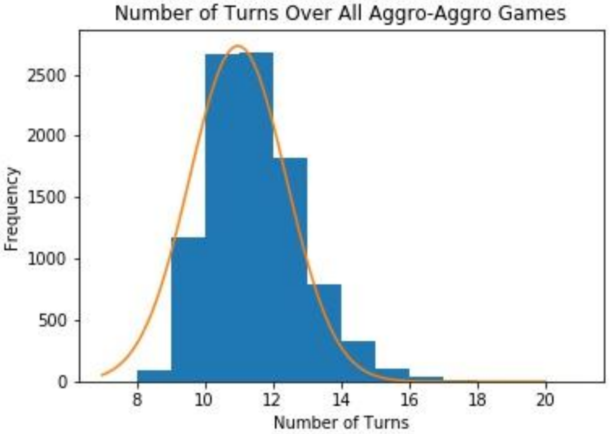
The Data

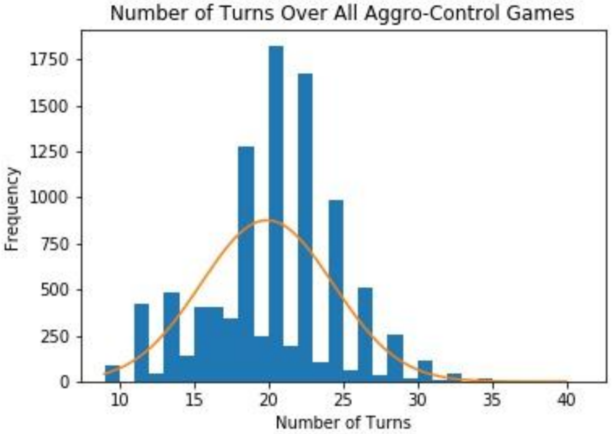
13 total columns...

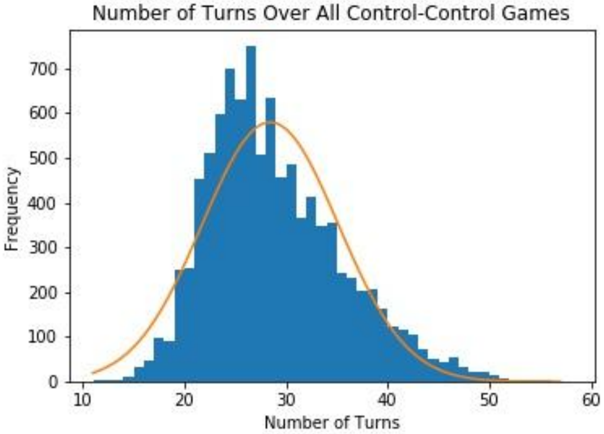
- Lots of feature engineering
- Player statistics per turn/game

PlayerStrategy	AvgHealedPerTurn	AvgHeroAttacksPerTurn	AvgCardsDrawnPerTurn	AvgCardsPlayedPerTurn
0.0	1.17	0.0	1.50	1.50
0.0	0.00	0.0	1.14	0.86
0.0	0.00	0.0	1.00	1.17
0.0	0.00	0.0	1.50	1.00
0.0	0.00	0.0	1.00	1.40
1.0	0.83	0.0	1.17	1.00
1.0	0.36	0.0	1.91	1.36
1.0	0.33	0.0	1.17	1.00
1.0	0.36	0.0	2.09	1.18
1.0	0.14	0.0	1.21	1.21

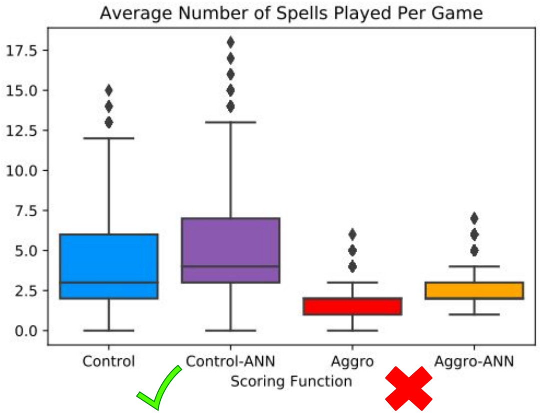
The Data


Number of Turns Over All Aggro-Aggro Games
Frequency
Number of Turns
0
500
1000
1500
2000
2500
8
10
12
14
16
18
20


Number of Turns Over All Aggro-Control Games
Frequency
Number of Turns
0
250
500
750
1000
1250
1500
1750
10
15
20
25
30
35
40

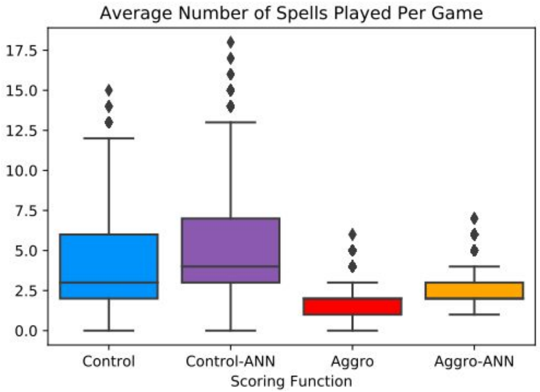

Number of Turns Over All Control-Control Games
Frequency
Number of Turns
0
100
200
300
400
500
600
700
10
20
30
40
50
60

The Data

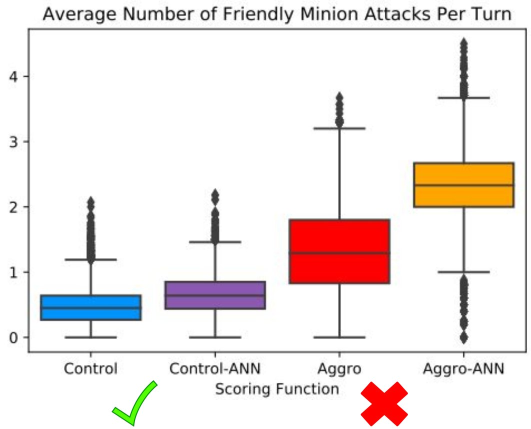

Average Number of Spells Played Per Game
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Control
Control-ANN
Aggro
Aggro-ANN
Scoring Function


1
The Soularium
Draw 3 cards.
At the end of your turn, discard them.

The Data



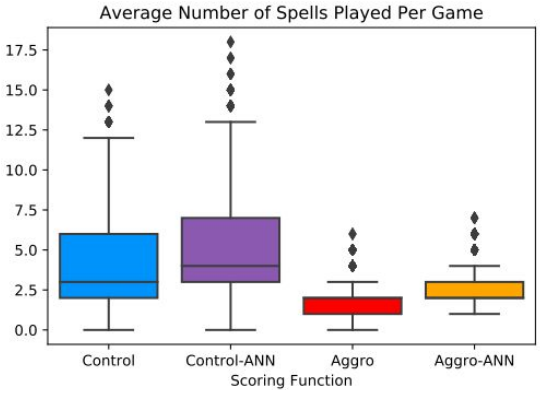
Average Number of Spells Played Per Game
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Control
Control-ANN
Aggro
Aggro-ANN
Scoring Function

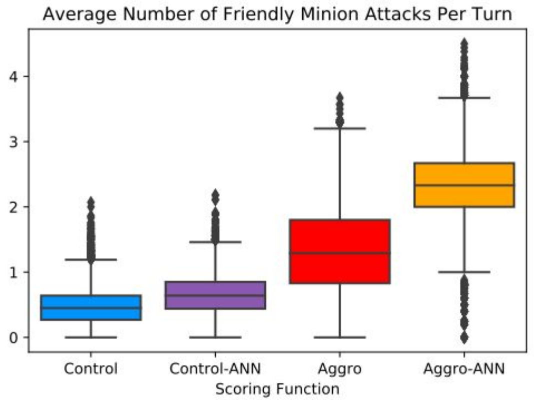


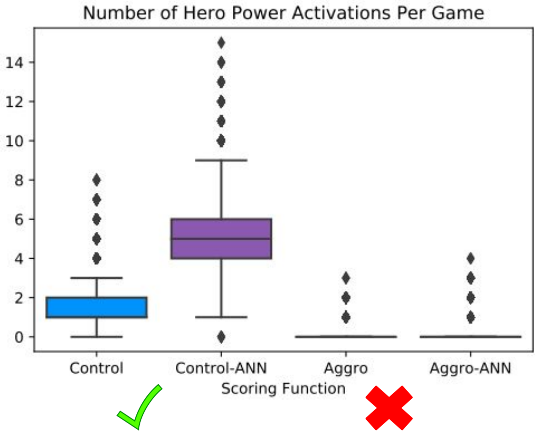
Average Number of Friendly Minion Attacks Per Turn
4
3
2
1
0
Control
Control-ANN
Aggro
Aggro-ANN
Scoring Function

1
Flame Imp
Battlecry: Deal 3 damage to your hero.
3
Demon
2
1
Flame Imp
Battlecry: Deal 3 damage to your hero.
3
Demon
2
1
Imp
1
Demon
1
1
Imp
1
Demon
1

The Data


Average Number of Spells Played Per Game
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Control
Control-ANN
Aggro
Aggro-ANN
Scoring Function


Average Number of Friendly Minion Attacks Per Turn
4
3
2
1
0
Control
Control-ANN
Aggro
Aggro-ANN
Scoring Function


Number of Hero Power Activations Per Game
14
12
10
8
6
4
2
0
Control
Control-ANN
Aggro
Aggro-ANN
Scoring Function

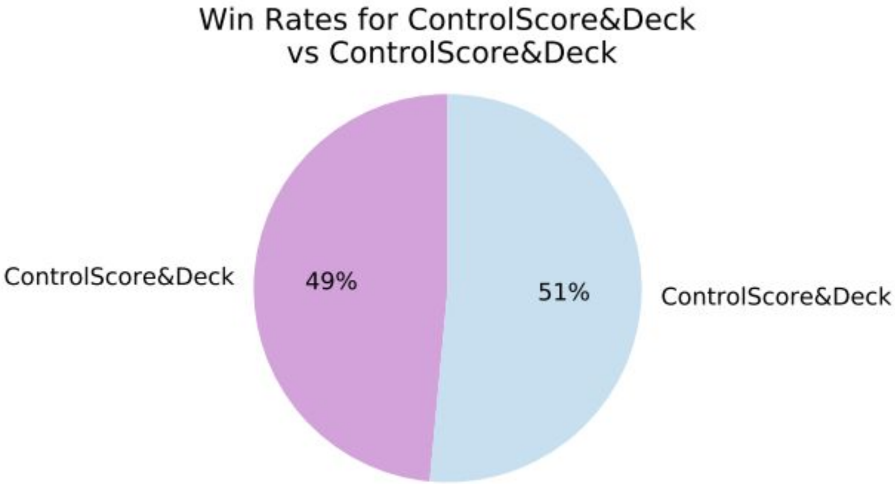

2
Life Tap
Hero Power
Draw a card and take 2 damage.

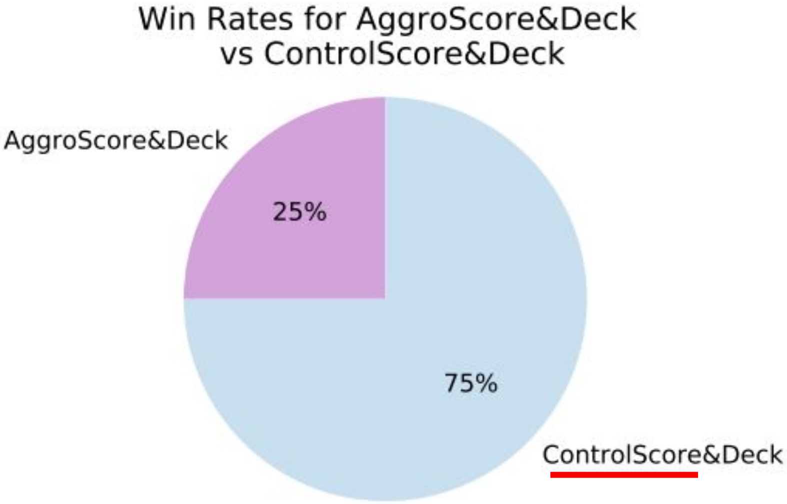
Experiment 1

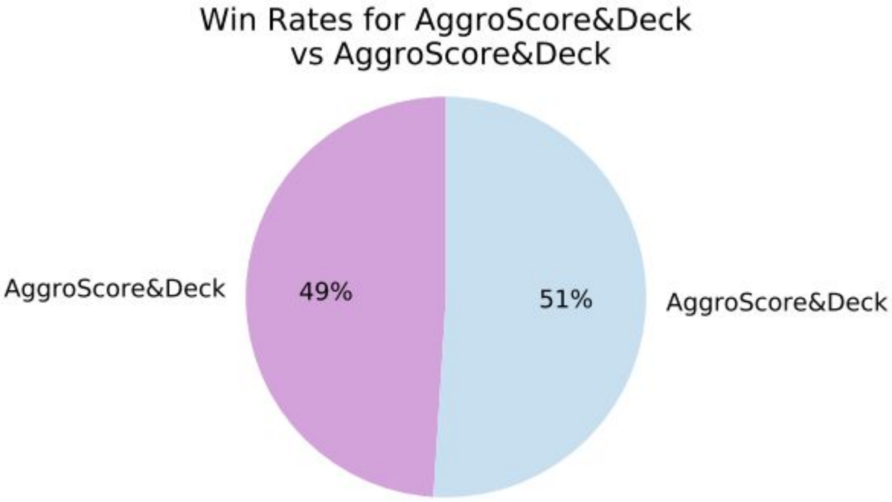
- Test aggro players vs control players
- Test decks using opposite scoring function
- Which performs better?
  - Is control > aggro generally?
  - Or is ControlScore > AggroScore?
- Hypotheses:
  - $\bar{x}_{control} > \bar{x}_{aggro}$
  - $\bar{x}_{ControlScore} > \bar{x}_{AggroScore}$

Results

- ControlScore > AggroScore
  - bottom
- 50/50 in mirror matches
  - (right)

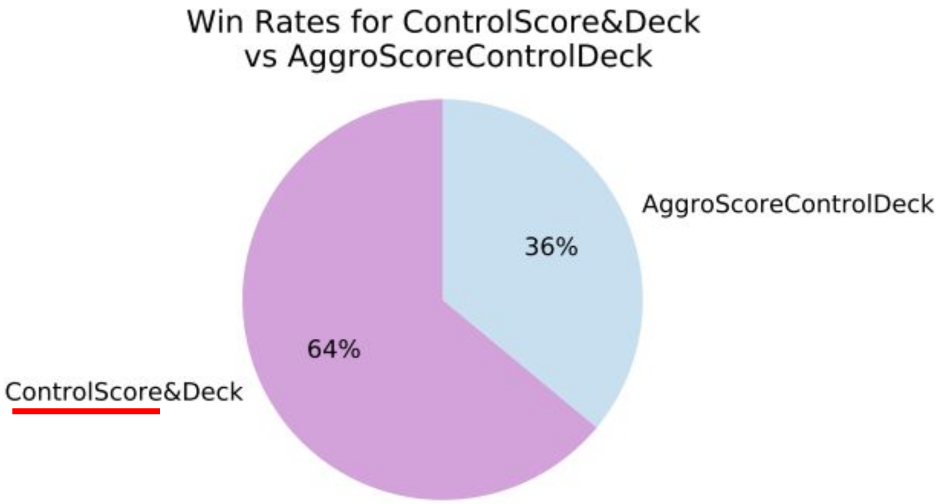


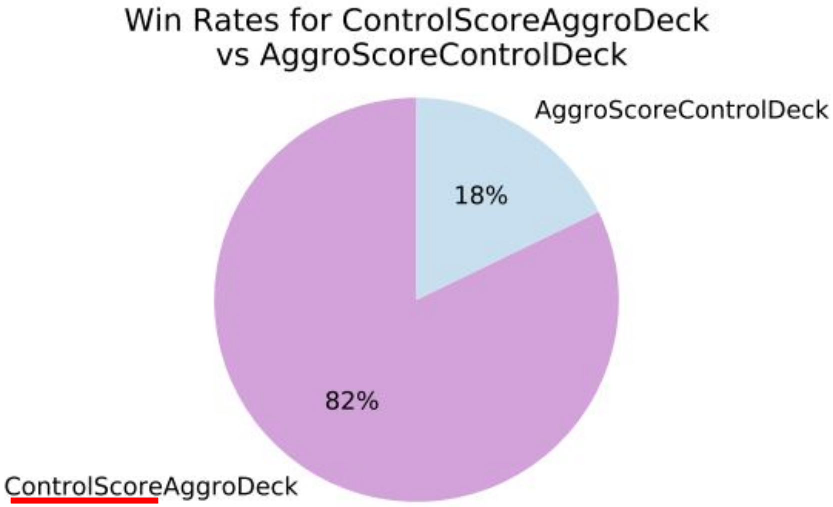


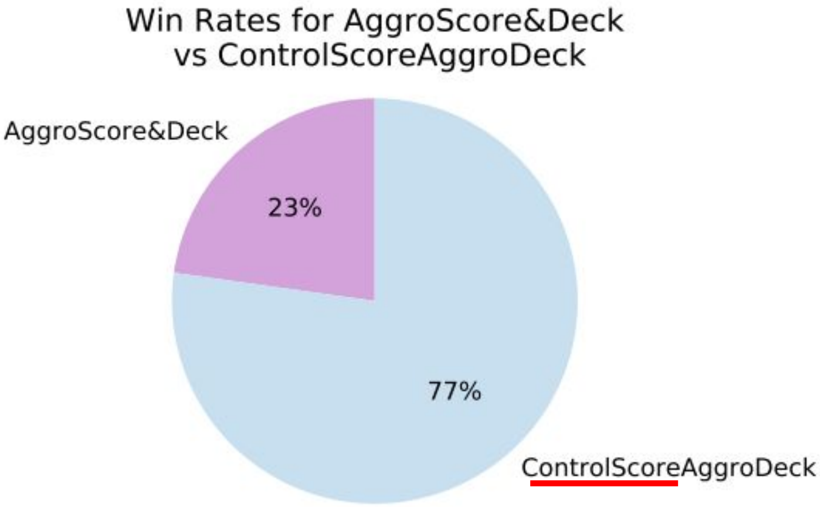


Results

- ControlScore > AggroScore
  - ControlScore won in all three charts







Experiment 2

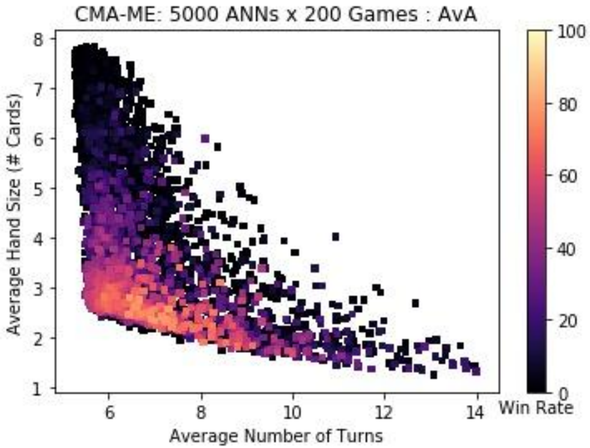
- Evolve networks with different behavior features
  - Number of ANNs to evolve
  - Change num turns / hand size
- Hypothesis:
  - $\bar{x}_{ANN} > \bar{x}_{Sabber}$

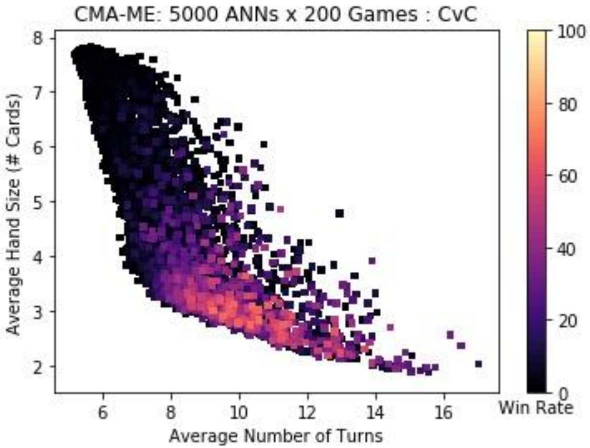
Table 4.3 CMA-ME Behavior Configurations[a]

Network Name	*Num Games per ANN*	*Num ANNs To Evaluate*	*HandSize [Min,Max]*	*NumTurns [Min,Max]*	*PlayerScore, OpponentScore*
WarlockNet_CC_sm	100	5000	[1,7]	[5,15]	Control,Control
CvsNNC_2.0	100	5000	[1,7]	[25,35]	Control,Control
CvsNNC_Large	200	50000	[1,9]	[5,45]	Control,Control
WarlockNet_AA_sm	100	5000	[1,7]	[5,15]	Aggro,Aggro
WarlockNet_AA_lg	200	50000	[1,9]	[5,45]	Aggro,Aggro

Results

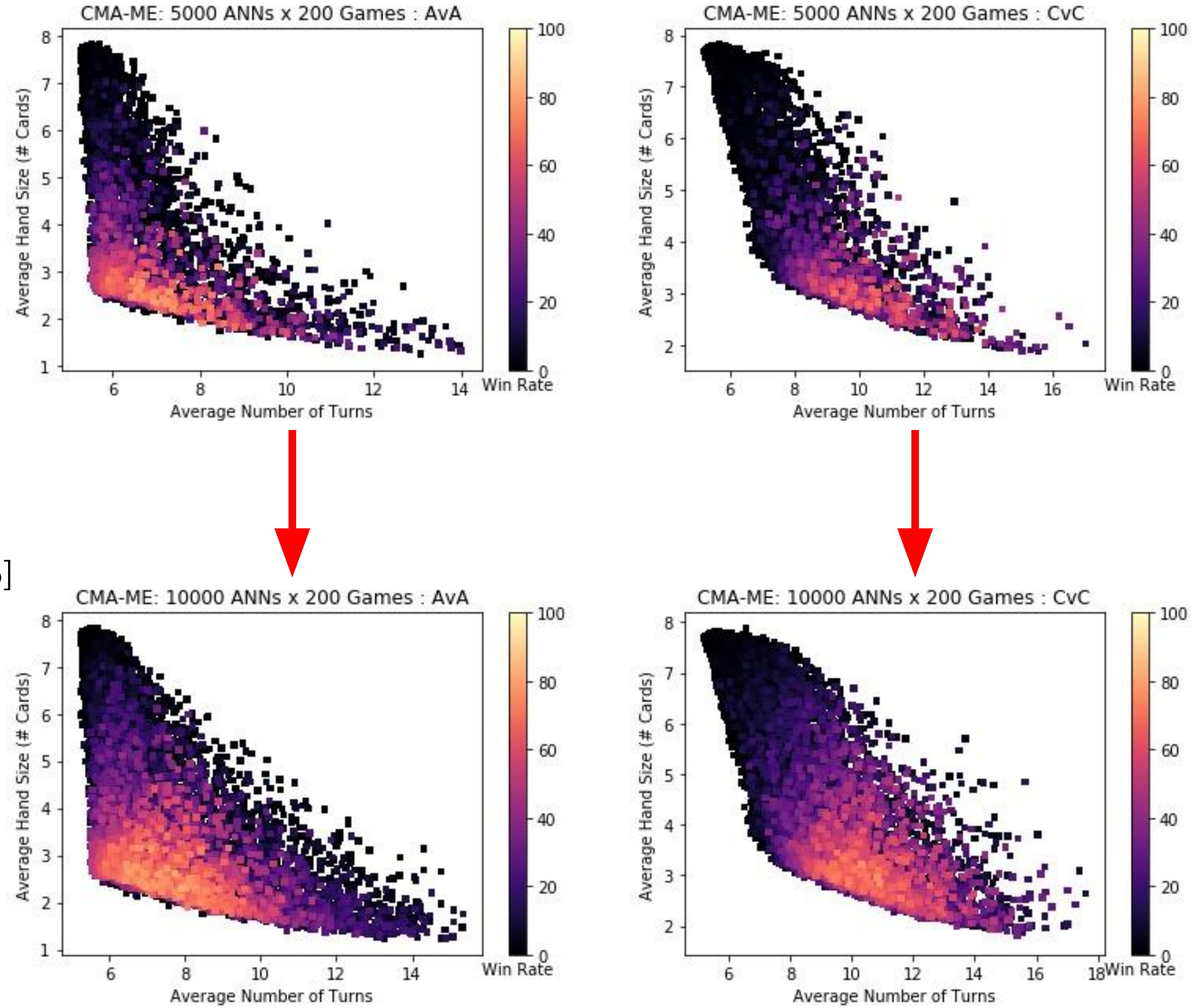
- Aggro networks (left) look much different than control networks (right)
- Main difference:
  - num turns (x axis)
- Where are the ANNs focused?





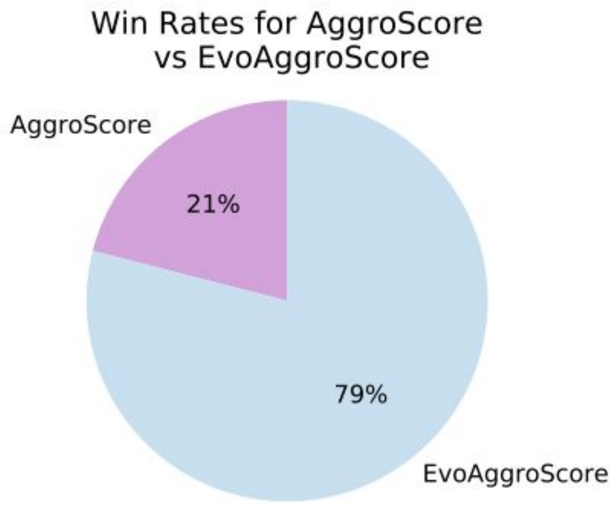
Results

- Aggro networks (left) look much different than control networks (right)
- Changed behaviors:
  - NumTurns [5,15] -> [5,45]
  - NumGames 100->200
- How do these change ANNs?



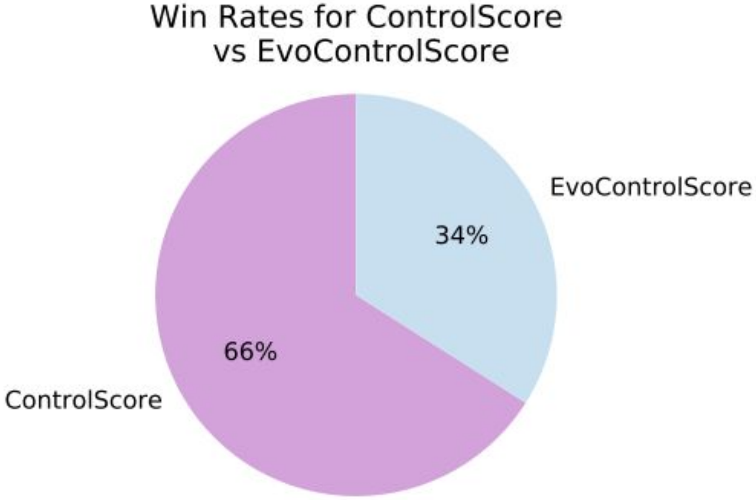
Results

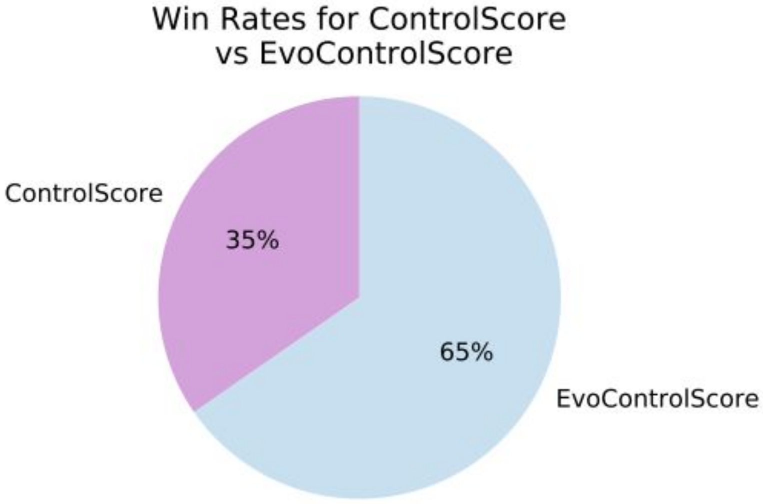
- Recall AggroScore v AggroScore had roughly 50% win rate
- ANN aggro players performed better than AggroScore



Results

- Top: same configs as aggro evolution
- Bottom: expanded num turns / hand size
- Change behaviors ->
  control ANNs perform better
  - Particularly for hand size / num turns
  - Evolves a stronger control network





Experiment 3

- Can a model predict / **generalize** gameplay strategy?
- Compare across **five** models [6]
- Train using aggro and control players
  - AggroScore/ControlScore
- Test using players with ANN heuristics

Results

- High accuracy on train/validation data
- Lower accuracy on test data
- Precision > Recall

Validation Data - Aggro/ControlScores

Model	*Accuracy*	*Precision*	*Recall*
Logistic Regression	0.9986	0.9981	0.9992
Random Forest	0.9990	0.9988	0.9992
SVM	0.9987	0.9983	0.9992
Decision Tree	0.9981	0.9978	0.9983
SGD Classifier	0.9988	0.9985	0.9992

Test Data - ANN Heuristics

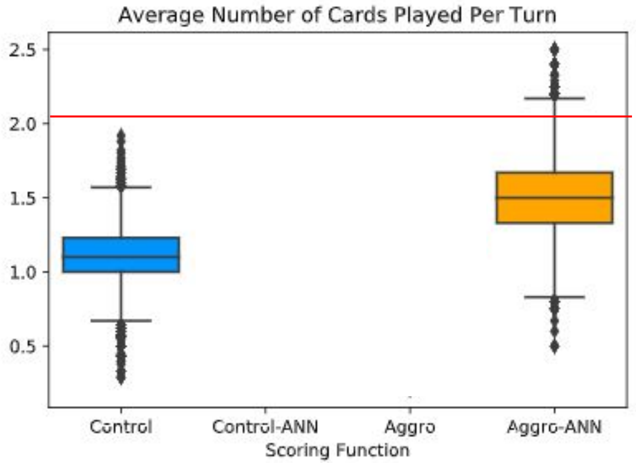
Model	*Accuracy*	*Precision*	*Recall*
Logistic Regression	0.6106	0.7390	0.3418
Random Forest	0.6623	0.8201	0.4157
SVM	0.6120	0.7408	0.3446
Decision Tree	0.6817	0.7854	0.4999
SGD Classifier	0.6245	0.7708	0.3543

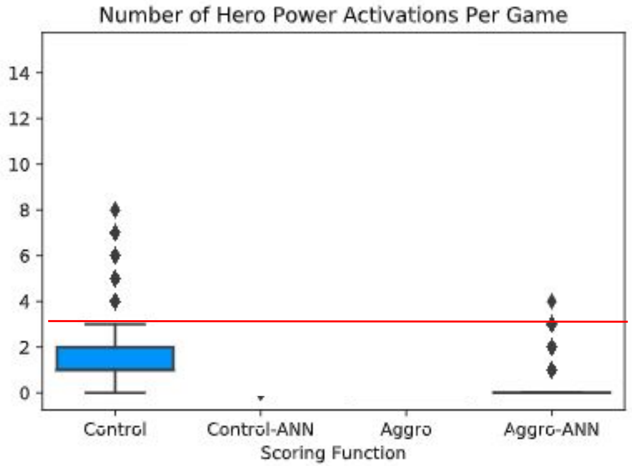
Results

- Control class mostly correct
- Aggro class mostly incorrect
  - Strategies tend to “bleed”
- Aggro-ANN features may fall into the ControlScore distribution

SVM Classifier Confusion Matrix

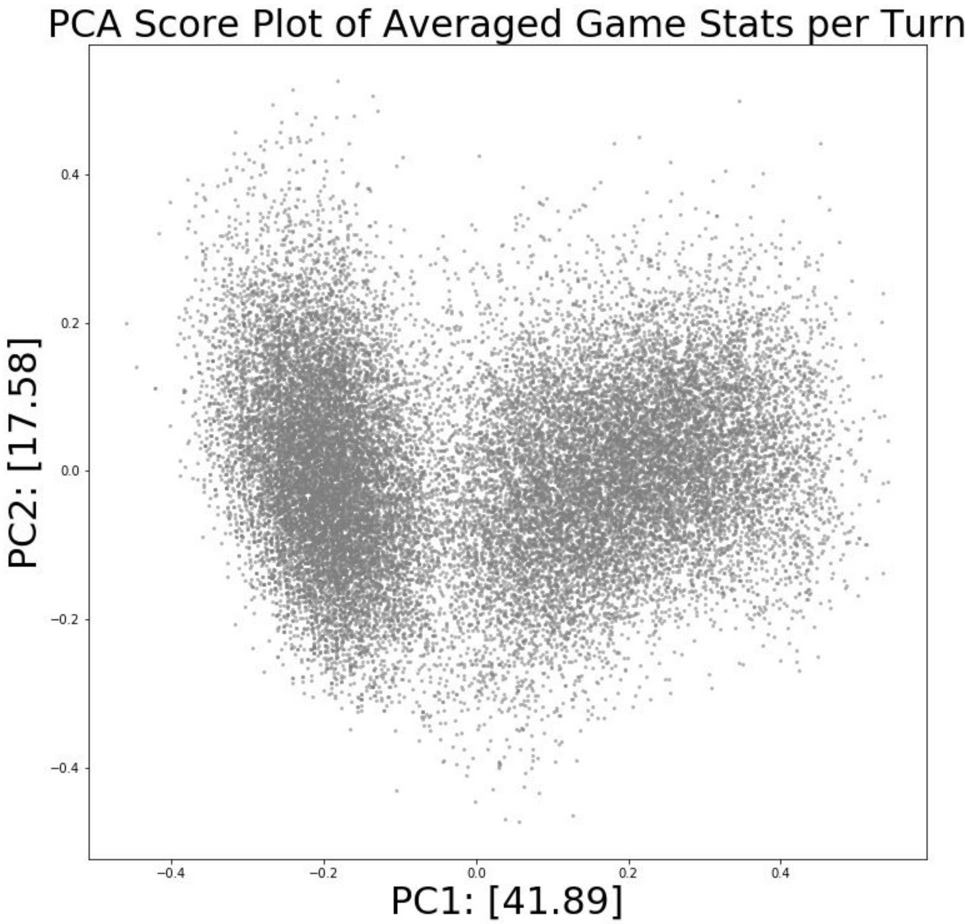
	Actual			
Predicted		Aggro	Control	
	Aggro	3532	1201	4733
	Control	6526	8757	15283
		9958	9958	19916



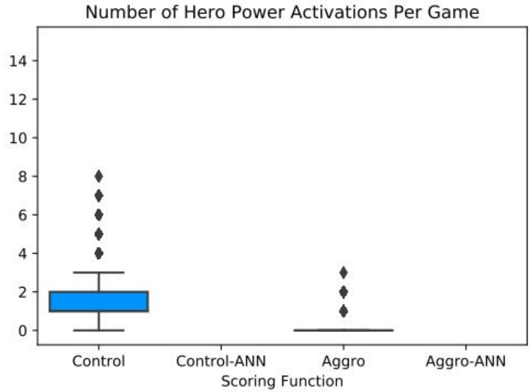


Experiment 4

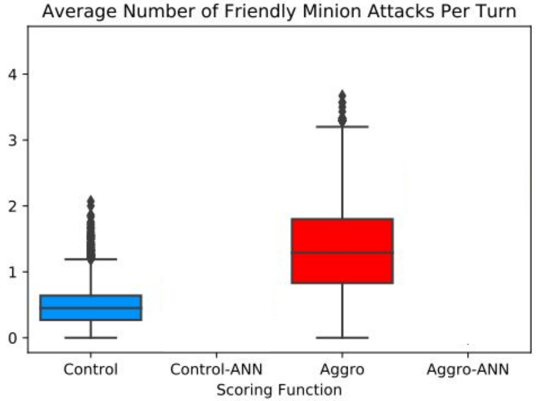
- How to visualize players in Cartesian space?
- PCA for space reduction
  - Projected on 2 axes
  - Aggro/Control Score only



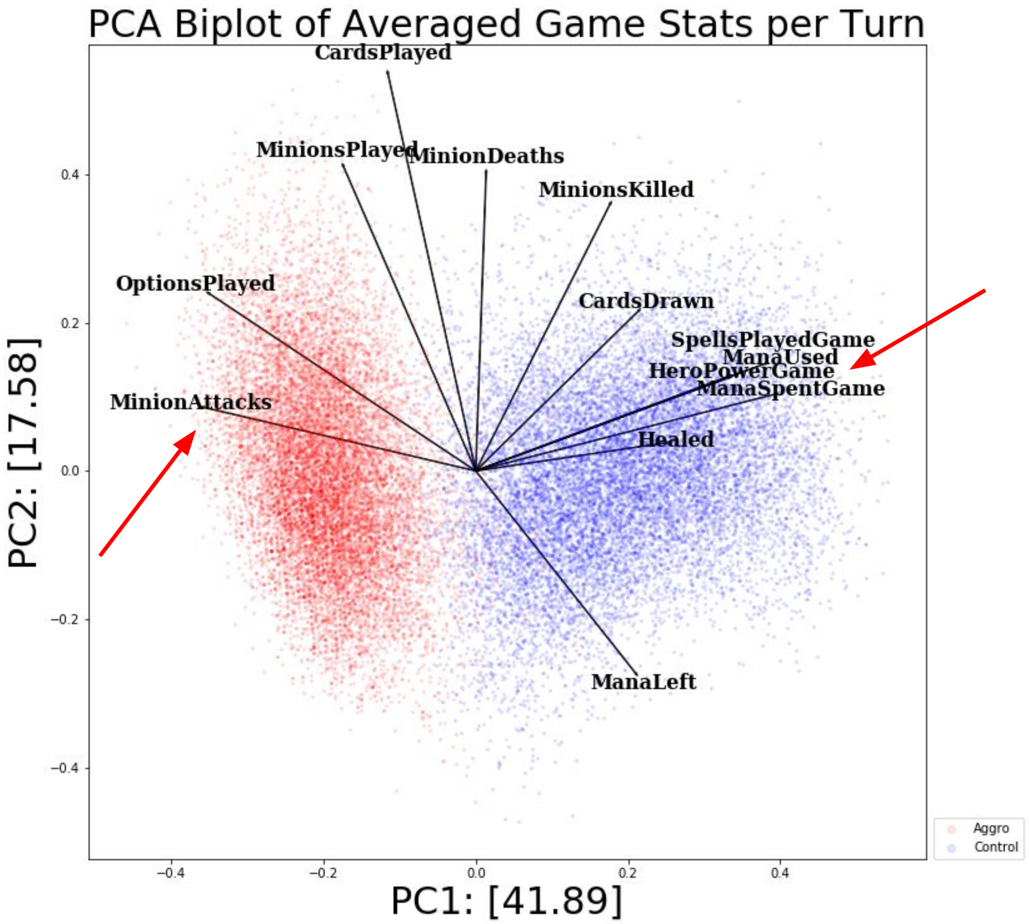
Results



Number of Hero Power Activations Per Game
14
12
10
8
6
4
2
0
Control
Control-ANN
Aggro
Aggro-ANN
Scoring Function



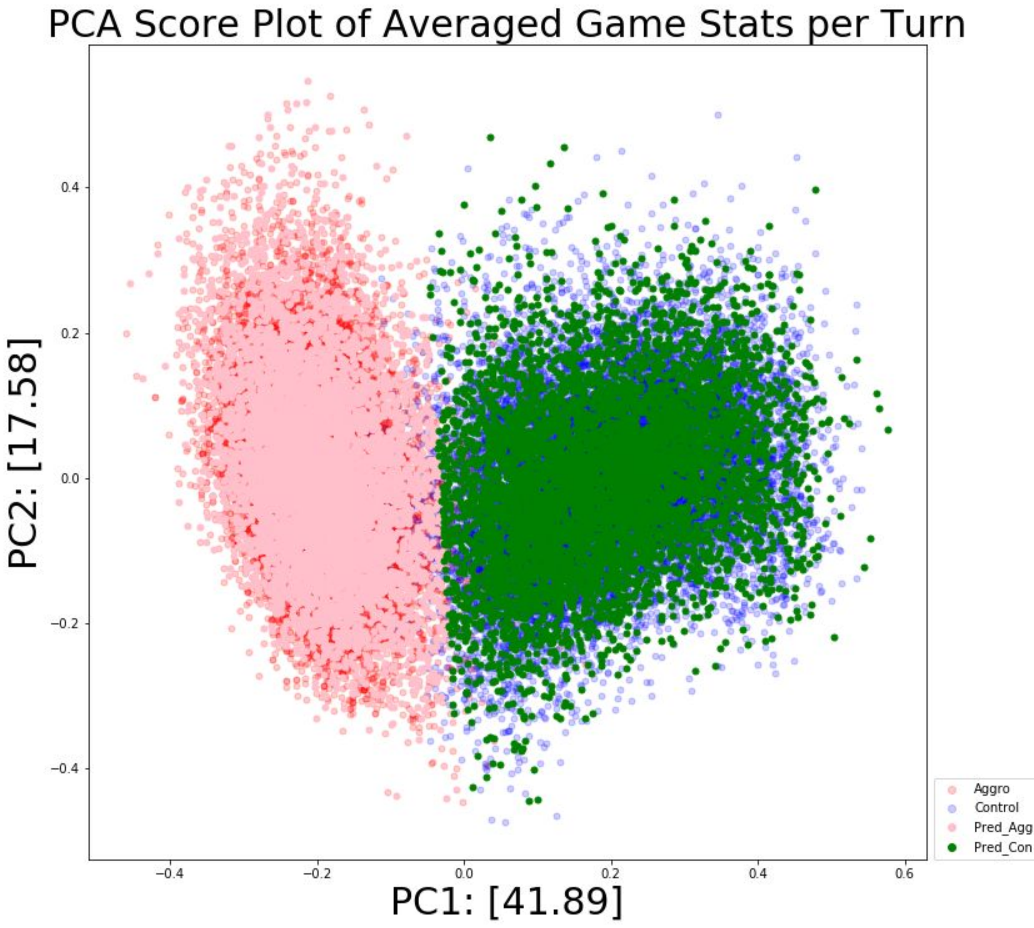
Average Number of Friendly Minion Attacks Per Turn
4
3
2
1
0
Control
Control-ANN
Aggro
Aggro-ANN
Scoring Function



PCA Biplot of Averaged Game Stats per Turn
CardsPlayed
MinionsPlayed
MinionDeaths
MinionsKilled
OptionsPlayed
CardsDrawn
SpellsPlayedGame
ManaUsed
HeroPowerGame
ManaSpentGame
MinionAttacks
Healed
ManaLeft
PC2: [17.58]
PC1: [41.89]
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4
Aggro
Control

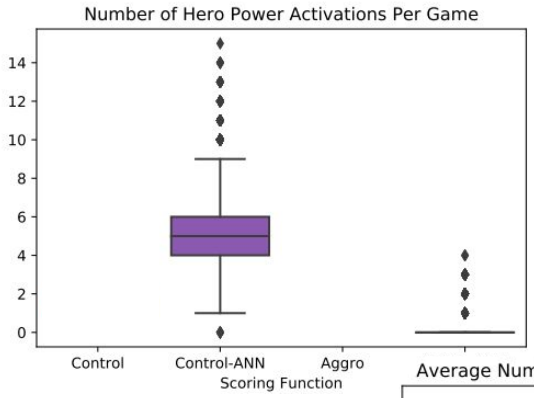
Results

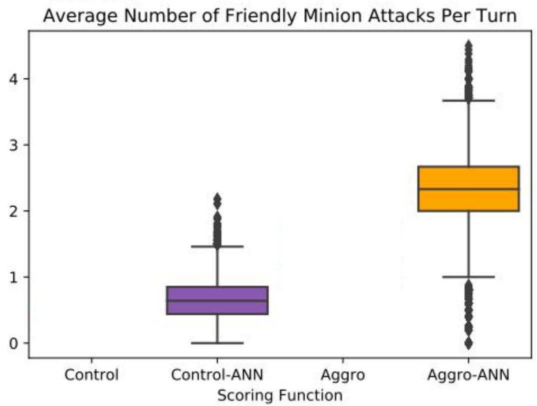
- Support Vector Classifier
- 98% accurate on validation data

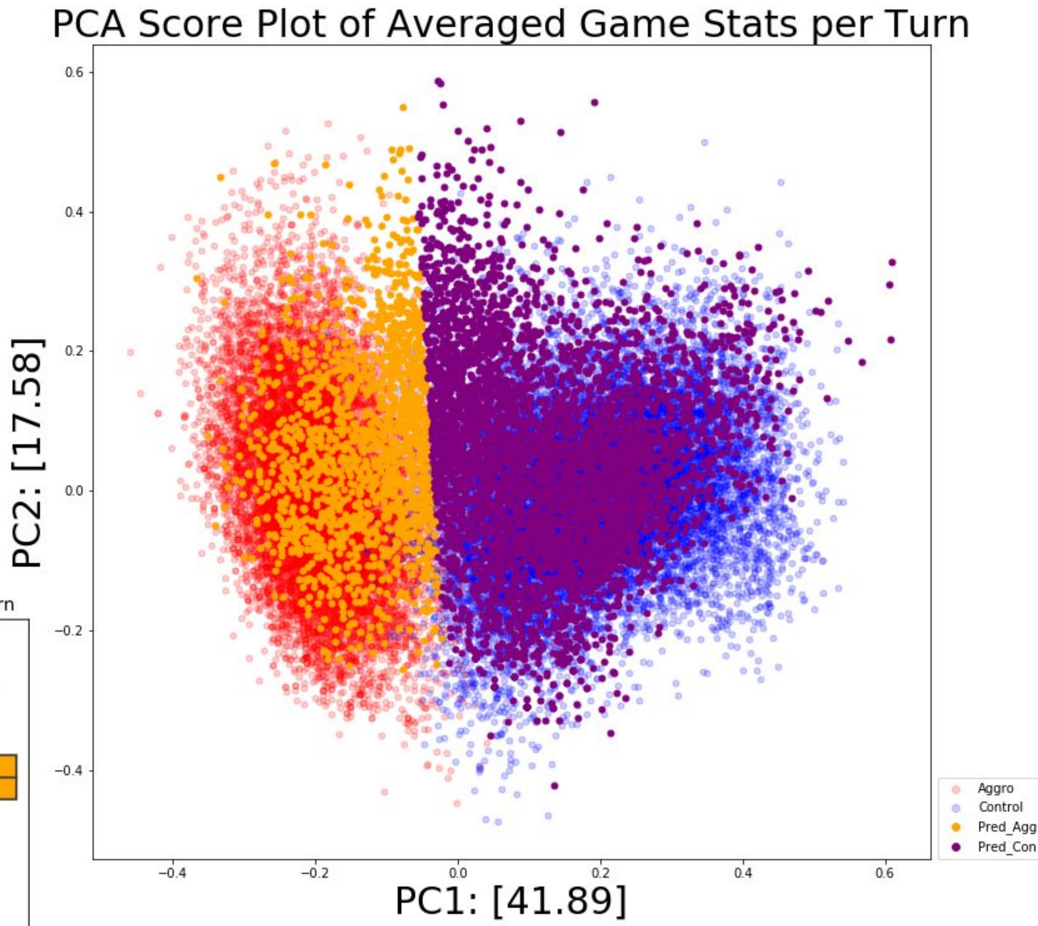


Results

- 75% accurate on test data (ANNs) vs 61% before







Conclusions

- What **is** validated
  - ControlScore better than AggroScore
  - CMA-ME produces better heuristics
  - Other applications?
- What **needs to be** improved
  - Supervised learning models
- Future work
  - More heroes / decks
  - Human players
  - Turn-by-turn prediction

Thank You

- Dr. Hoover
- Dr. Roshan
- Dr. Basu Roy
- The Public

Questions

Bibliography

[1] Stiegler, Andreas, et al. "Symbolic reasoning for hearthstone." IEEE Transactions on Games 10.2 (2017): 113-127.
[2] M. Campbell, A. J. Hoane Jr, and F.-h. Hsu, “Deep blue,” Artificial intelligence, vol. 134, no. 1-2, pp. 57–83, 2002.
[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot et al., “Mastering the game of go with deep neural networks and tree search,” nature, vol. 529, no. 7587, p. 484, 2016.
[4] M. C. Fontaine, J. Togelius, S. Nikolaidis, and A. K. Hoover, “Covariance matrix adaptation for the rapid illumination of behavior space,” arXiv preprint arXiv:1912.02400, 2019.
[5] A. Bhatt, S. Lee, F. de Mesentier Silva, C. W. Watson, J. Togelius, and A. K. Hoover, “Exploring the hearthstone deck space,” in Proceedings of the 13th International Conference on the Foundations of Digital Games. ACM, 2018, p. 18.
[6] Fernández-Delgado, Manuel, et al. "Do we need hundreds of classifiers to solve real world classification problems?." The journal of machine learning research 15.1 (2014): 3133-3181.
[7] D. Ha, “A visual guide to evolution strategies,”blog.otoro.net, 2017. [Online].Available: https://blog.otoro.net/2017/10/29/visual-evolution-strategies/
[8] Cuccu, Giuseppe, Julian Togelius, and Philippe Cudré-Mauroux. "Playing atari with six neurons." Proceedings of the 18th international conference on autonomous agents and multiagent systems. International Foundation for Autonomous Agents and Multiagent Systems, 2019.
[9] Bellemare, Marc G., et al. "The arcade learning environment: An evaluation platform for general agents." Journal of Artificial Intelligence Research 47 (2013): 253-279.

Appendix (Unused)

Appendix should contain supporting charts / slides with more details. This section was unused. Instead, some draft slides were pushed to the back.

Hyperparameter Search

- SVM / LogReg -
    - l1/l2 regularization,
    - C (0.1, 1, 10, 100) - regularization parameter
- RF number of estimators (50, 100, 200)
- DT
    - (gini, entropy)
    - splits (best, random)
    - max depth (4-10)
- SGD loss (hinge, log, perceptron, modified huber),
    - l1/l2 regularization
    - learning rate (1, 0.1, 0.01, 0.001)

Hyperparameter Search Resilts

The results of the hyperparameter search are below:

- Logistic Regression: l2 regularizer, C coefficient of 10
- Random Forest: 100 estimators
- Support Vector Classifier: l2 regularizer, C coefficient of 1
- Decision Tree Classifier: best splitters, max depth of 10, entropy criterion
- SGD CLassifier: modified huber loss, l2 regularizer, alpha 0.001