GameNetViz

Jason Hartford
Neil Newman
CPSC547

What?

Behavioural Game Theory aims to predict the behaviour of **people** as they interact **strategically**

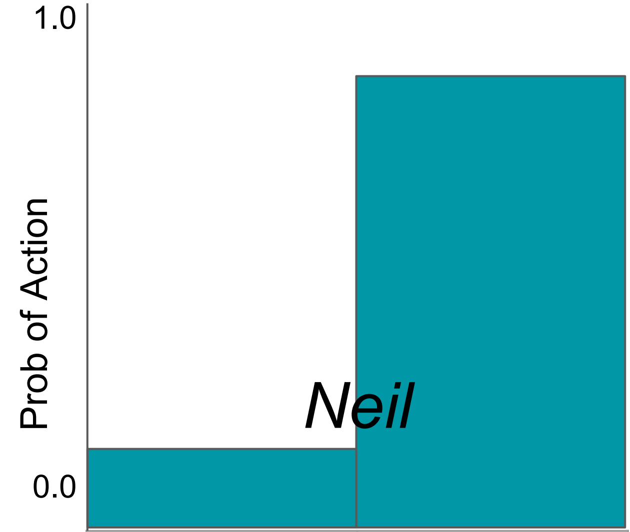
		Neil	
		Left	*Right*
Jason	Up	10, *15*	0, *19*
	Down	15, *5*	14, *3*

How might you reason about this game?

This is known as the “Max max payoff” feature

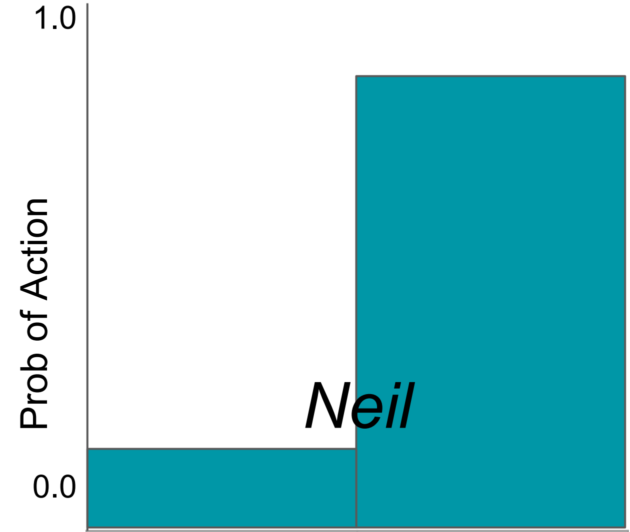
		Neil	
		Left	*Right*
Jason	Up	10, *15*	0, *19*
	Down	15, *5*	14, *3*

Assume Neil plays right or left with some probability

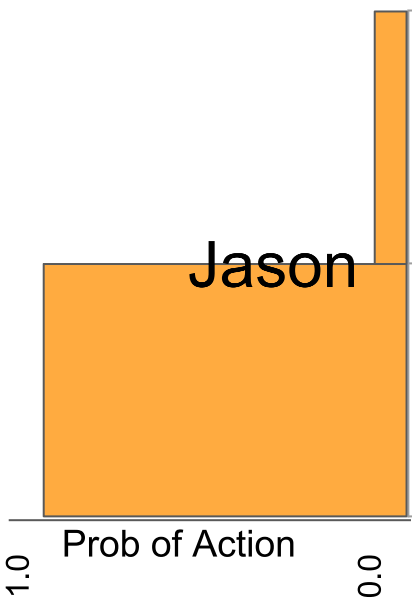


		Left	*Right*
Jason	Up	14, *15*	0, *19*
	Down	3, *5*	10, *3*

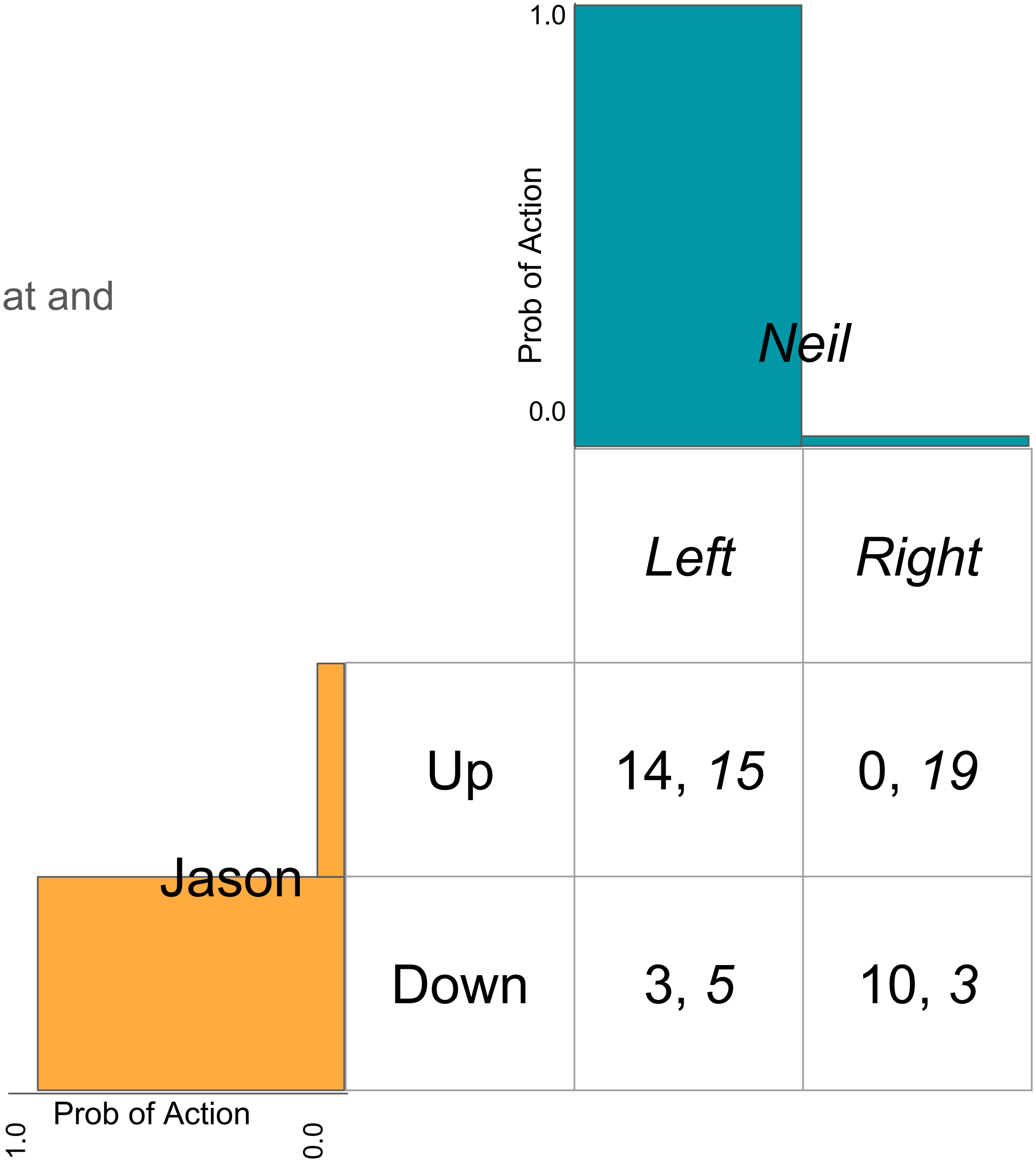
Respond by choosing the action that make you best off, given your assumption.



	Left	*Right*
Up	14, *15*	0, *19*
Down	3, *5*	10, *3*



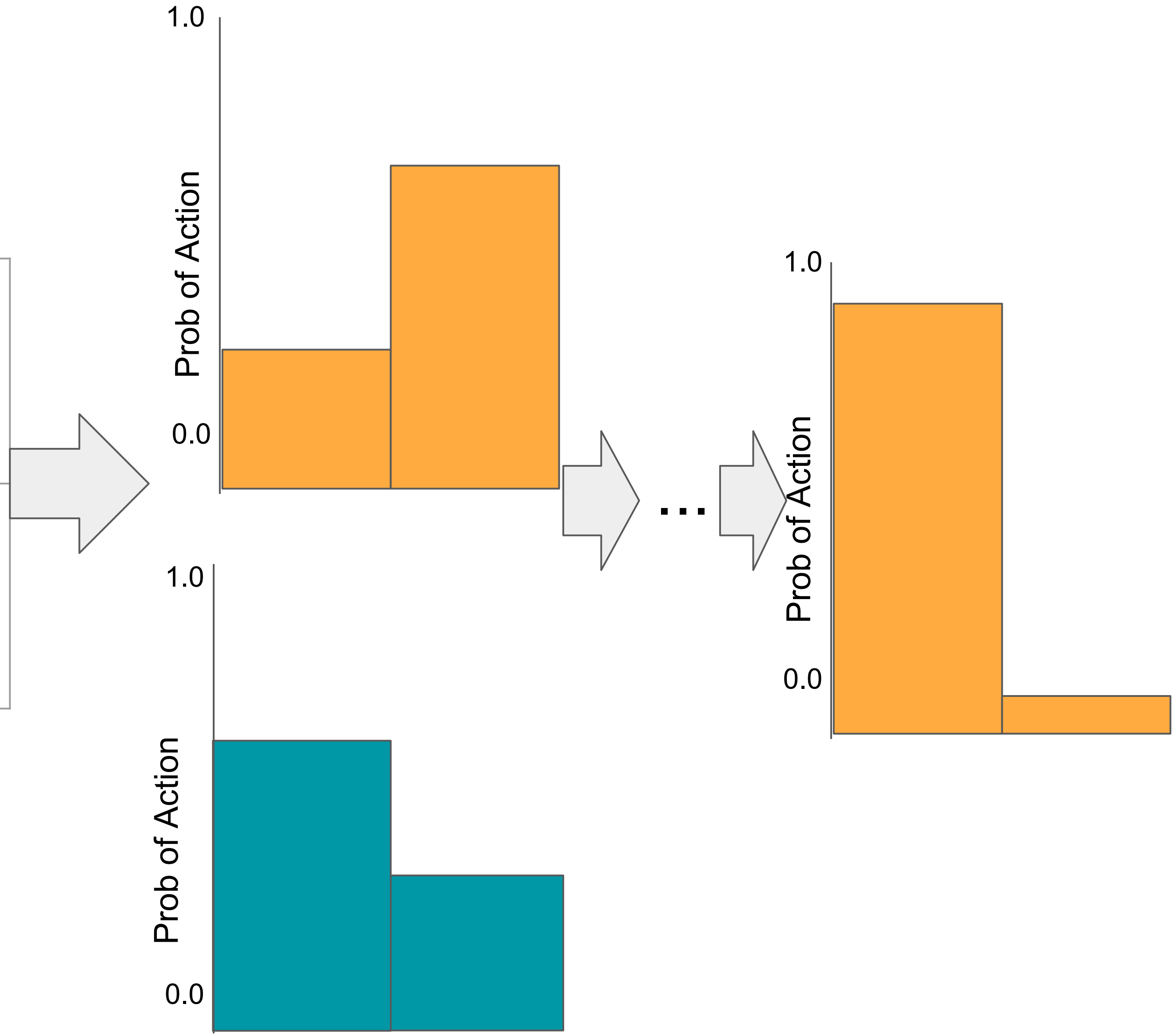
But Neil may think of that and change his action...



	Left	*Right*
Up	14, *15*	0, *19*
Down	3, *5*	10, *3*

Abstractly...

14, *15*	0, *19*
3, *5*	10, *3*



Data

Experimental Data

Data from 9 behavioural economics experiments on human subjects

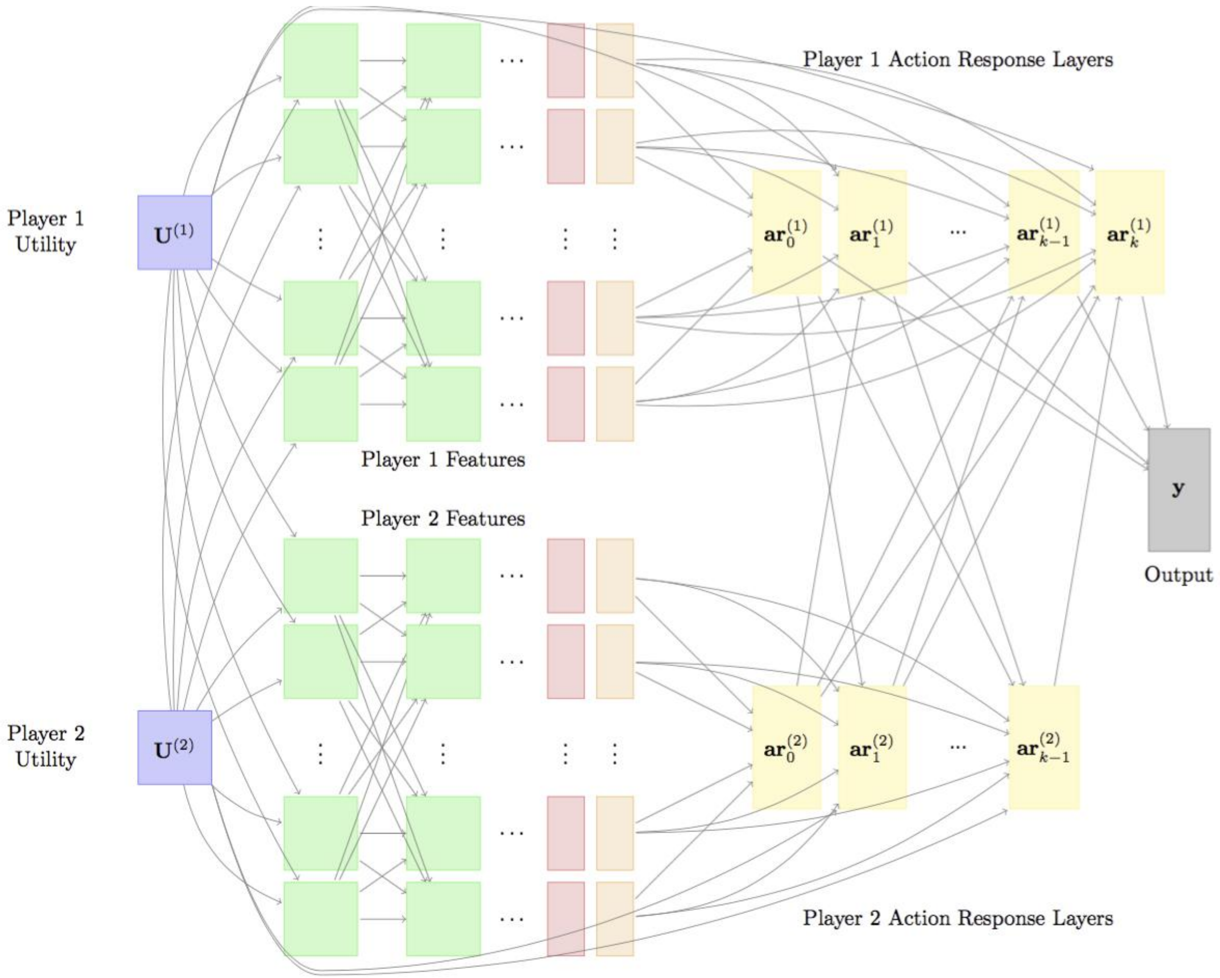
128 unique games with 12 071 plays

Model Data

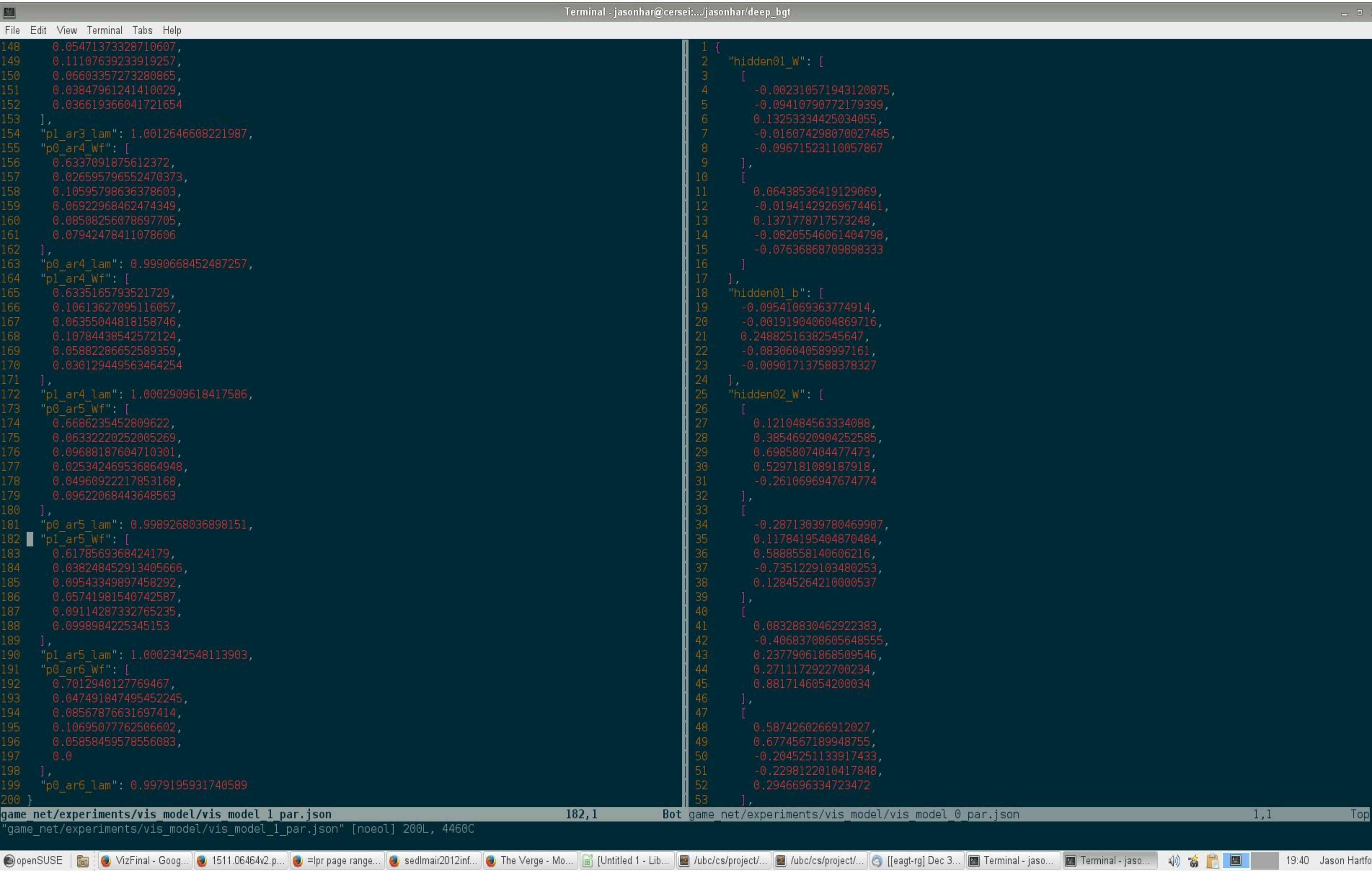
Set of numbers that parameterise the model

Output at each intermediate stage of computation.

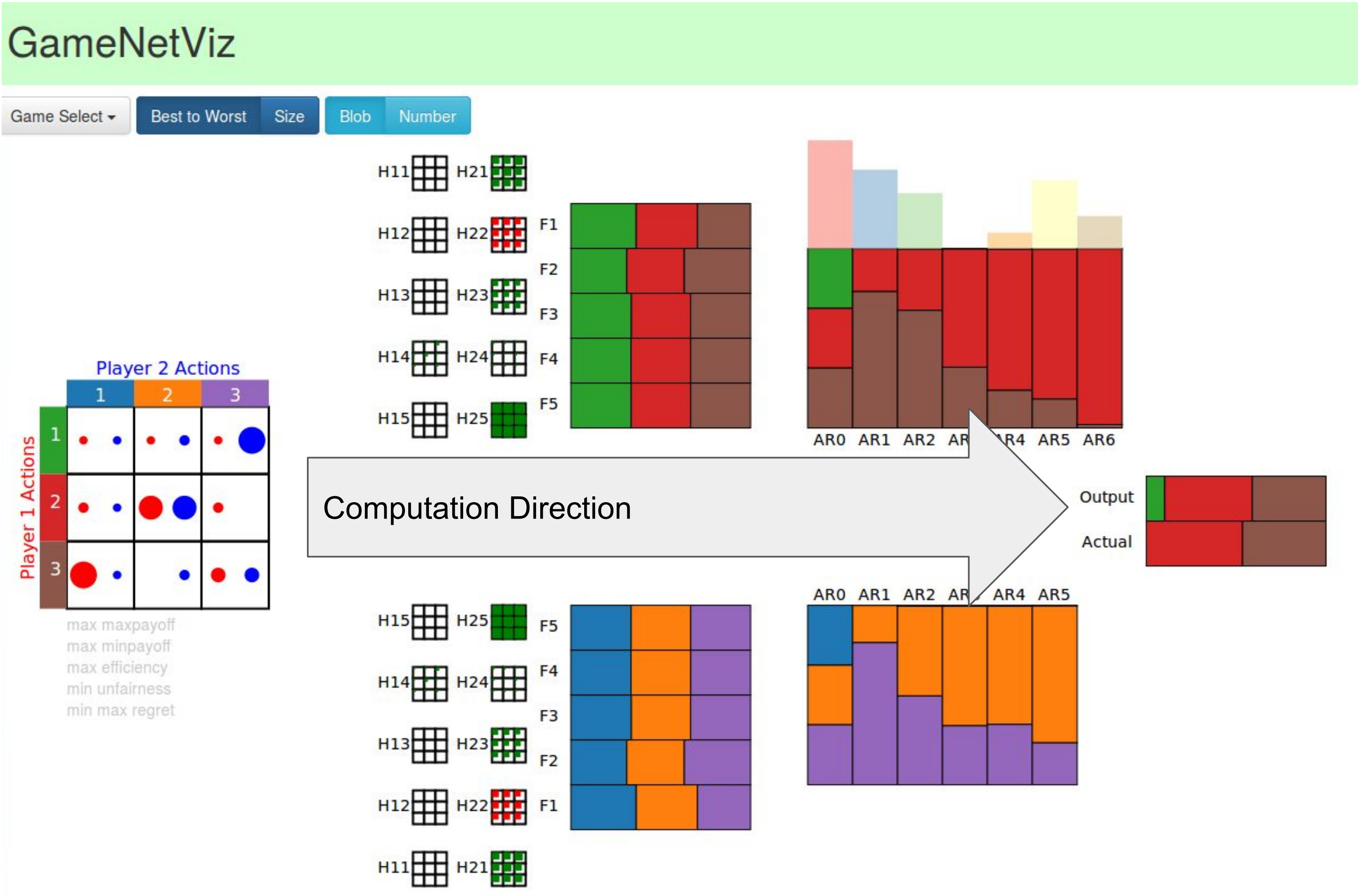
Model…



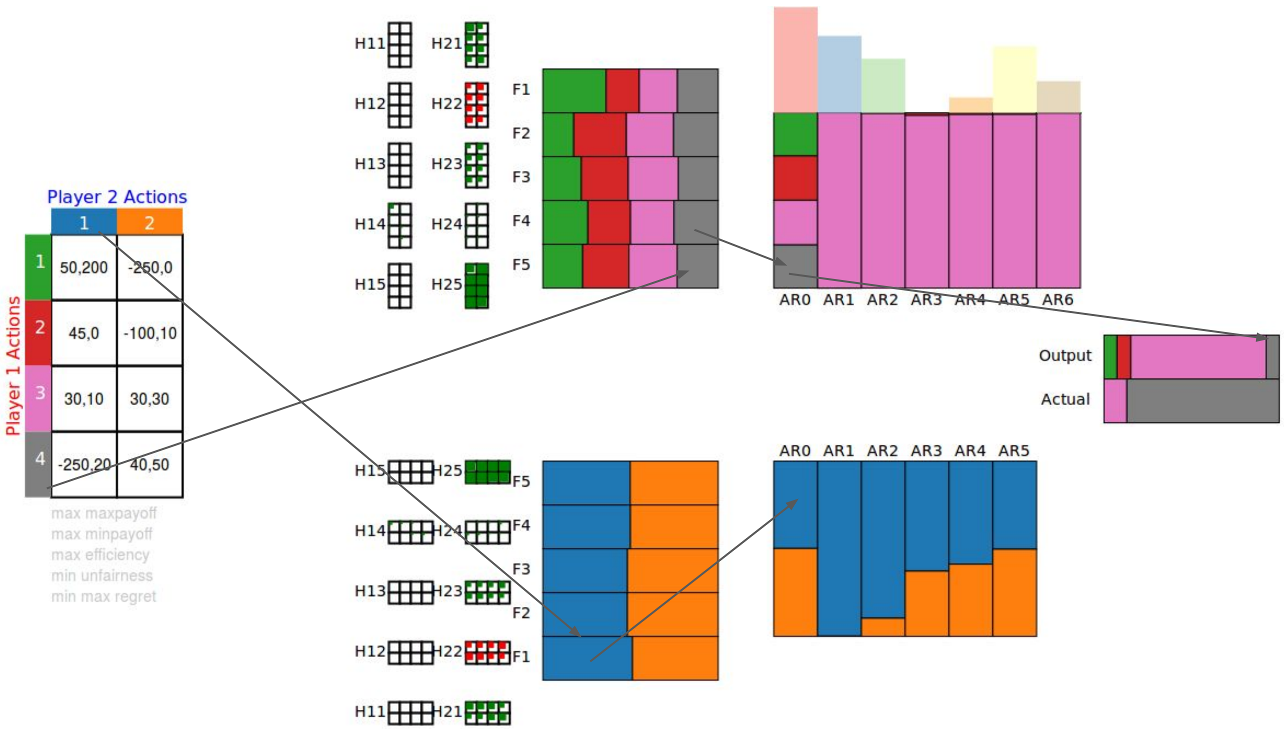
- Many parameters
- Difficult to visualise intermediate computation
- Difficult to identify design flaws / poor optimization fits


Terminal - jasonhar@cersei:.../jasonhar/deep_bgt
File Edit View Terminal Tabs Help
0.05471373328710607,
0.11107639233919257,
0.06603357273280865,
0.03847961241410029,
0.03661936604172165
],
"p1_ar3_lam": 1.0012646608221987,
"p0_ar4_Wf": [
0.6337091875612372,
0.026595796552470373,
0.10595798636378603,
0.06922968462474349,
0.08508256078697705,
0.07942478411078606
],
"p0_ar4_lam": 0.9990668452487257,
"p1_ar4_Wf": [
0.6335165793521729,
0.10613627095116057,
0.06355044818158746,
0.10784438542572124,
0.05882286652589359,
0.030129449563464254
],
"p1_ar4_lam": 1.0002909618417586,
"p0_ar5_Wf": [
0.6686235452809622,
0.06332220252005269,
0.09688187604710301,
0.025342469536864948,
0.04960922217853168,
0.09622068443648563
],
"p0_ar5_lam": 0.9989268036898151,
"p1_ar5_Wf": [
0.6178569368424179,
0.038248452913405666,
0.09543349897458292,
0.05741981540742587,
0.09114287332765235,
0.0998984225345153
],
"p1_ar5_lam": 1.0002342548113903,
"p0_ar6_Wf": [
0.7012940127769467,
0.047491847495452245,
0.08567876631697414,
0.10695077762506602,
0.05858459578556083,
0.0
],
"p0_ar6_lam": 0.9979195931740589
}
game_net/experiments/vis_model/vis_model_1_par.json 182,1 Bot
"game_net/experiments/vis_model/vis_model_1_par.json" [noeol] 200L, 4460C
{
"hidden01_W": [
[
-0.002310571943120875,
-0.09410790772179399,
0.13253334425034055,
-0.016074298070027485,
-0.09671523110057867
],
[
0.06438536419129069,
-0.01941429269674461,
0.1371778717573248,
-0.08205546061404798,
-0.07636868709898333
]
],
"hidden01_b": [
-0.09541069363774914,
-0.001919040604869716,
0.24882516382545647,
-0.08306040589997161,
-0.009017137588378327
],
"hidden02_W": [
[
0.1210484563334088,
0.38546920904252585,
0.6985807404477473,
0.5297181089187918,
-0.2610696947674774
],
[
-0.28713039780469907,
0.11784195404870484,
0.5888558140606216,
-0.7351229103480253,
0.12845264210000537
],
[
0.08328830462922383,
-0.40683708605648555,
0.23779061868509546,
0.2711172922700234,
0.8817146054200034
],
[
0.5874260266912027,
0.6774567189948755,
-0.2045251133917433,
-0.2298122010417848,
0.2946696334723472
],
game_net/experiments/vis_model/vis_model_0_par.json 1,1 Top
openSUSE
VizFinal - Goog...
1511.06464v2.p...
=lpr page range...
sedlmair2012inf...
The Verge - Mo...
[Untitled 1 - Lib...
/ubc/cs/project/...
/ubc/cs/project/...
[[eagt-rg] Dec 3...
Terminal - jaso...
Terminal - jaso...
19:40
Jason Hartford

Our Solution

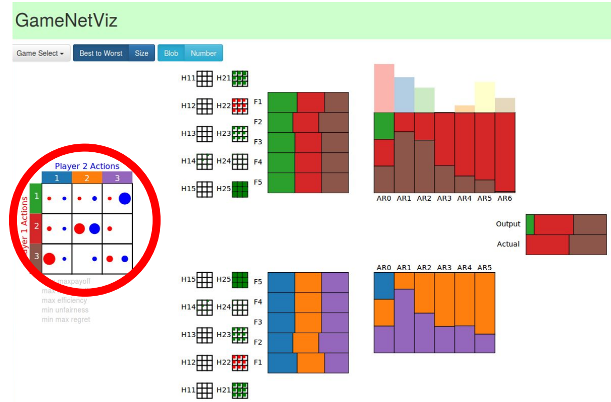

GameNetViz
Game Select
Best to Worst
Size
Blob
Number
H11 H21
H12 H22
H13 H23
H14 H24
H15 H25
F1
F2
F3
F4
F5
AR0 AR1 AR2 AR4 AR5 AR6
Player 2 Actions
1 2 3
Player 1 Actions
1
2
3
max maxpayoff
max minpayoff
max efficiency
min unfairness
min max regret
Computation Direction
Output
Actual
AR0 AR1 AR2 AR4 AR5
H15 H25
H14 H24
H13 H23
H12 H22
H11 H21

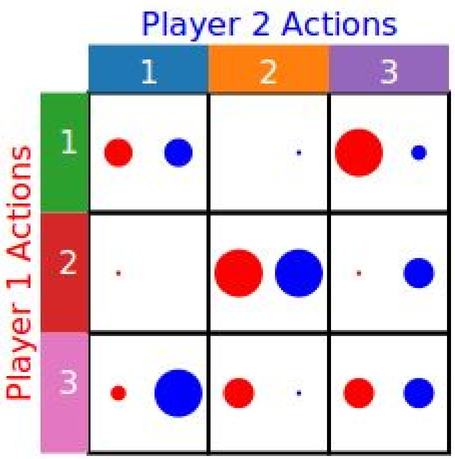
Common Colours



- The same colour corresponds to the same action throughout the viz

Payoff Matrix Viz



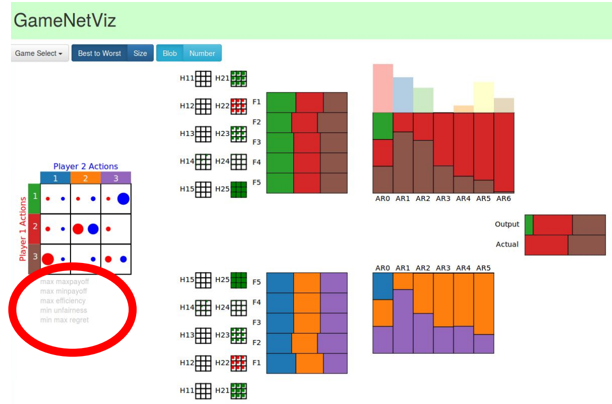


Player 2 Actions

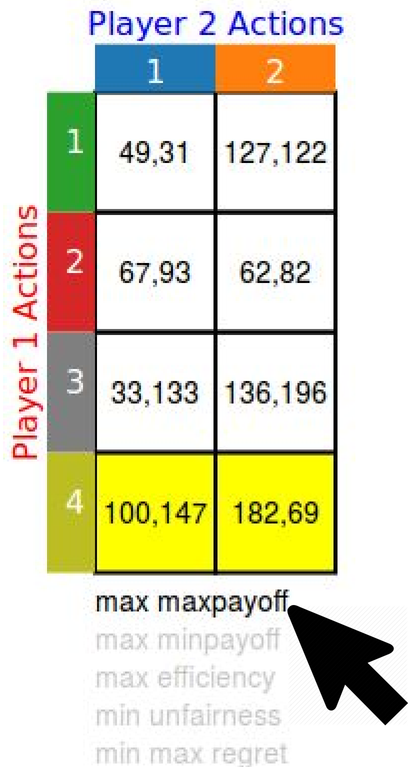
Player 1 Actions	1	2	3
1	35,35	0,1	100,10
2	1,0	100,100	1,40
3	10,100	40,1	40,40

- Allows for quick summary of a game
- Very easy to spot mismatched payoffs

- Detailed information, for when subtle differences matter

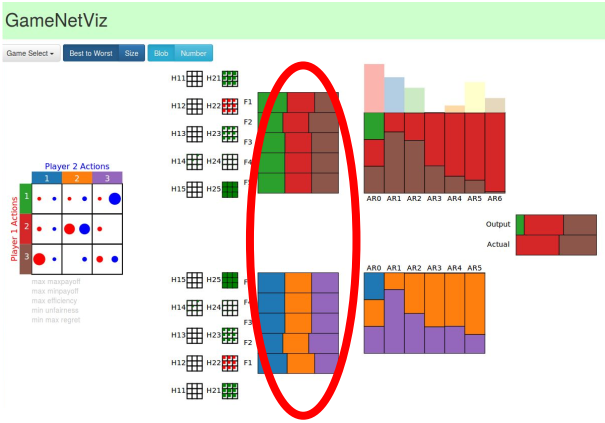


Hand-crafted Features

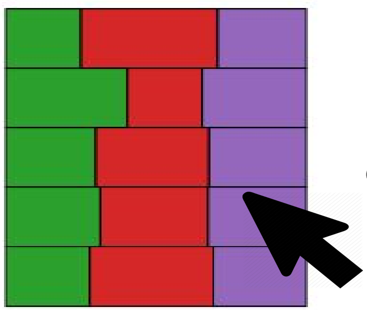


- From previous work, we know about features that players seem to like
- When hovering over a hand-crafted feature, the row(s) corresponding to that feature light up
- In this image, player 1 can achieve the highest payoff by picking action 4 (if player 2 picks action 2)
- Non-hovered features are grayed out, to avoid distraction

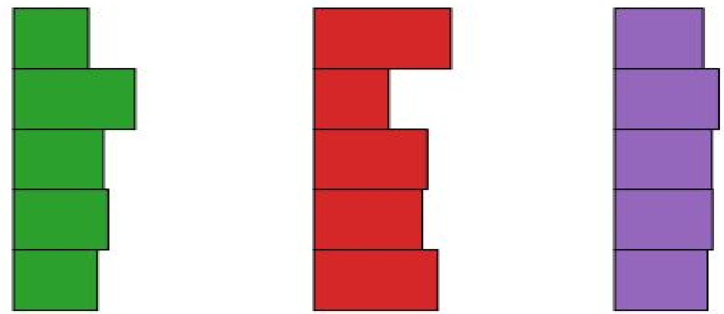
Features

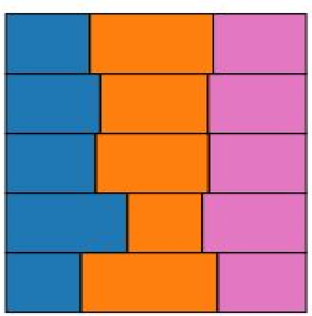


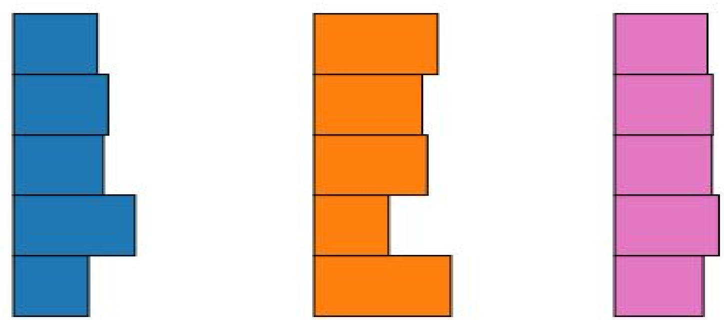
- Each feature outputs a different probability distribution of playing each action



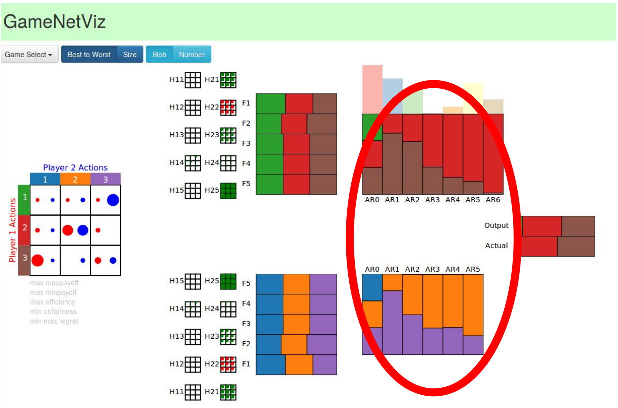
- Clicking on the stacked bar charts splits them into grouped bar charts, so that you can compare an action's distribution across different features

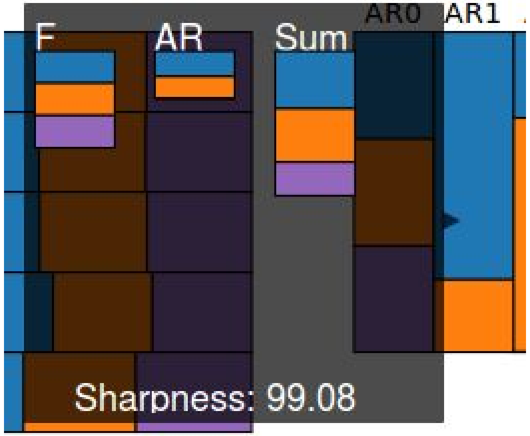






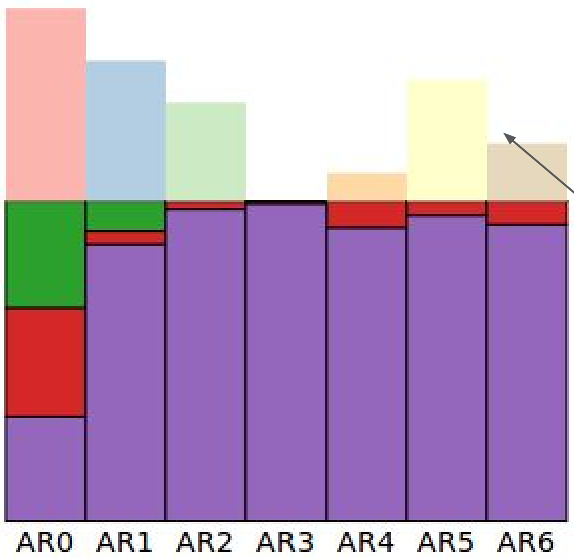
Action Response Layers
Tool tip





- AR layers are weighted sums of the feature units, as well as previous AR layers
- Hovering over an AR unit produces a tool-tip showing a breakdown of how it is composed, before the non-linearity is applied.
- The sharpness parameter of the non-linearity function for that AR layer is also displayed

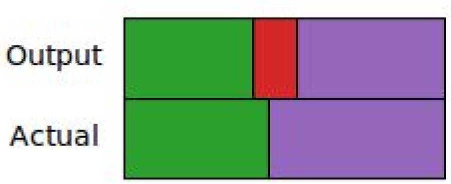
Level distribution



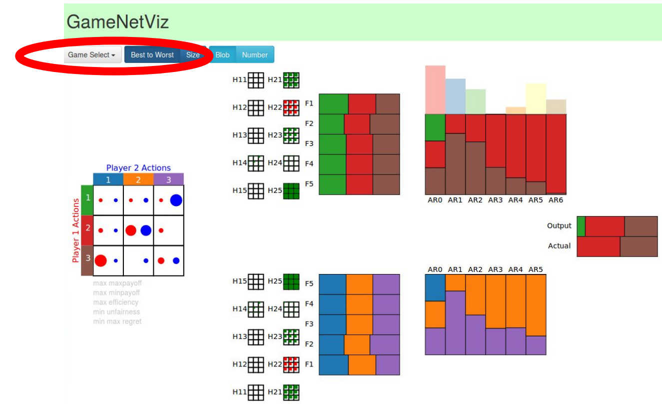
- The output is a weighted sum of all the AR units. The weight associated with each AR is encoded as a bar above the AR.

Output vs Actual Play

- Compare the model's predicted distribution of play against observed play (from experiments with human players)



Game Selector



- Games can be ordered by size, or by a derived difference between the model’s prediction and observed play
- Choosing a game from the selector will render data for that game

DEMO

Critique & Future Work

- The blob payoff matrix encoding is invariant to scaling, so two scaled games look the same. But humans have a non-linear response to payoffs, and maybe we can find an encoding that matches this.
- Hidden layer encoding not offering any insights, could be better
- Handle larger games (e.g. 100 x 100)
- Show even more data! (Parameters, optimization)

Applause