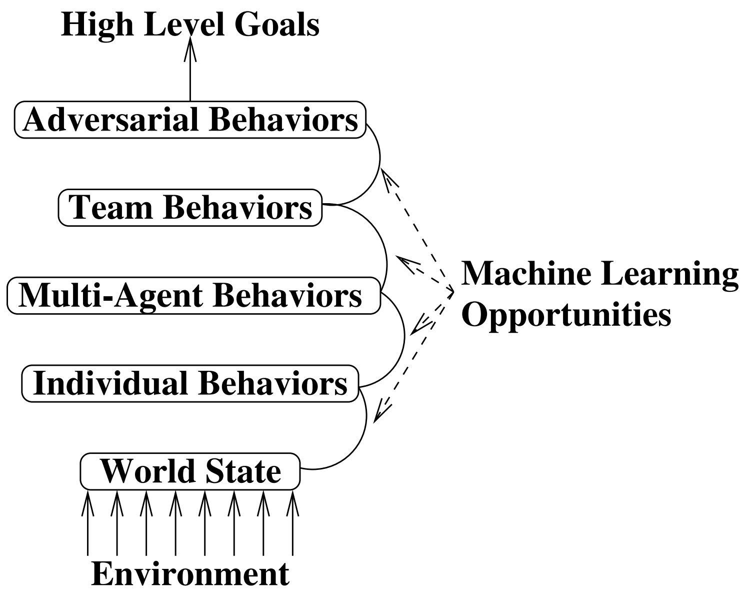
Layered Learning

- Identify h consistent with examples from a class of hypotheses H
- Domains too **complex** for tractably mapping state features $S \longmapsto$ outputs O
- Hierarchical subtask decomposition **given**: $\{L_1, L_2, \ldots, L_n\}$
- Machine learning: **exploit data** to train, adapt
- **Learning in one layer feeds into next layer**



Layer Definition

Each layer is defined as

$$L_i = (\vec{F}_i, O_i, T_i, M_i, h_i)$$

$\vec{F}_i$ – input vector of state features relevant for learning subtask L_i.

$$\vec{F}_i = <F_i^1, F_i^2, \ldots>, \forall j,\ F_1^j \in S.$$

O_i – set of outputs result of subtask L_i. $O_n = O$.

T_i – set of training examples used for learning subtask L_i. Each element of T_i consists of a correspondence between an input feature vector $\vec{f} \in \vec{F}_i$ and $o \in O_i$.

M_i – the ML algorithm used at layer L_i to select a hypothesis mapping $\vec{F}_i \mapsto O_i$ based on T_i.

h_i – result of running M_i on T_i. h_i is a function from $\vec{F}_i$ to O_i.

Affecting the Next Layer

- $\forall i < n$, h_i directly affects L_{i+1} in at least 1 of 3 ways:
  - h_i used to construct features F_{i+1}^k
  - h_i used to construct elements of T_{i+1}
  - h_i used to prune the output set O_{i+1}

- Since F_{i+1} can consist of new features constructed using h_i,

$$\forall i, j,\ F_i^j \in S \cup_{k=1}^{i-1} O_k$$

Layered Learning in Robotic Soccer

	Strategic Level	**Example**
L_1	individual	ball interception
L_2	multi-agent	pass evaluation
L_3	team	pass selection
L_4	team	strategic positioning
L_5	adversarial	strategic adaptation

Implementation – M_i

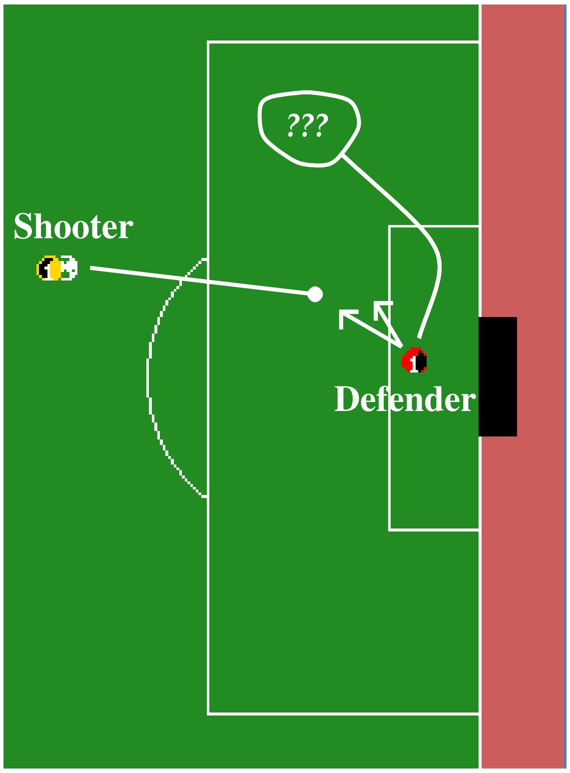
L_1**:** off-line learning using a neural network

L_2**:** off-line learning using a decision tree

L_3**:** a new multi-agent, on-line ML method

L_4, L_5**:** defined within flexible team structure

Ball Interception: A Robust Individual Skill



- Defender moves straight to ball $\Longrightarrow$ miss
- Learns to intercept ball's path
- Neural network (M_1) with 4 inputs ($\vec{F}_1$):
  - Ball angle and distance
  - Ball velocity (speed and direction)

Ball Interception

- $F_1 = \{BallDist_t, BallAng_t, BallDist_{t-1}\}$
- $O_1 = \{TurnAng\}$
- T_1 = randomly generated labeled training examples from shots
- M_1 = a neural network:
  - Fully-connected with 4 sigmoid hidden units
  - Learning rate $= 10^{-6}$
  - 3000 epochs
- h_1 = a trained interception behavior

Intercepting the Ball (T_1)

BD = Ball's distance
BA = Ball's angle
TA = *Turn angle*

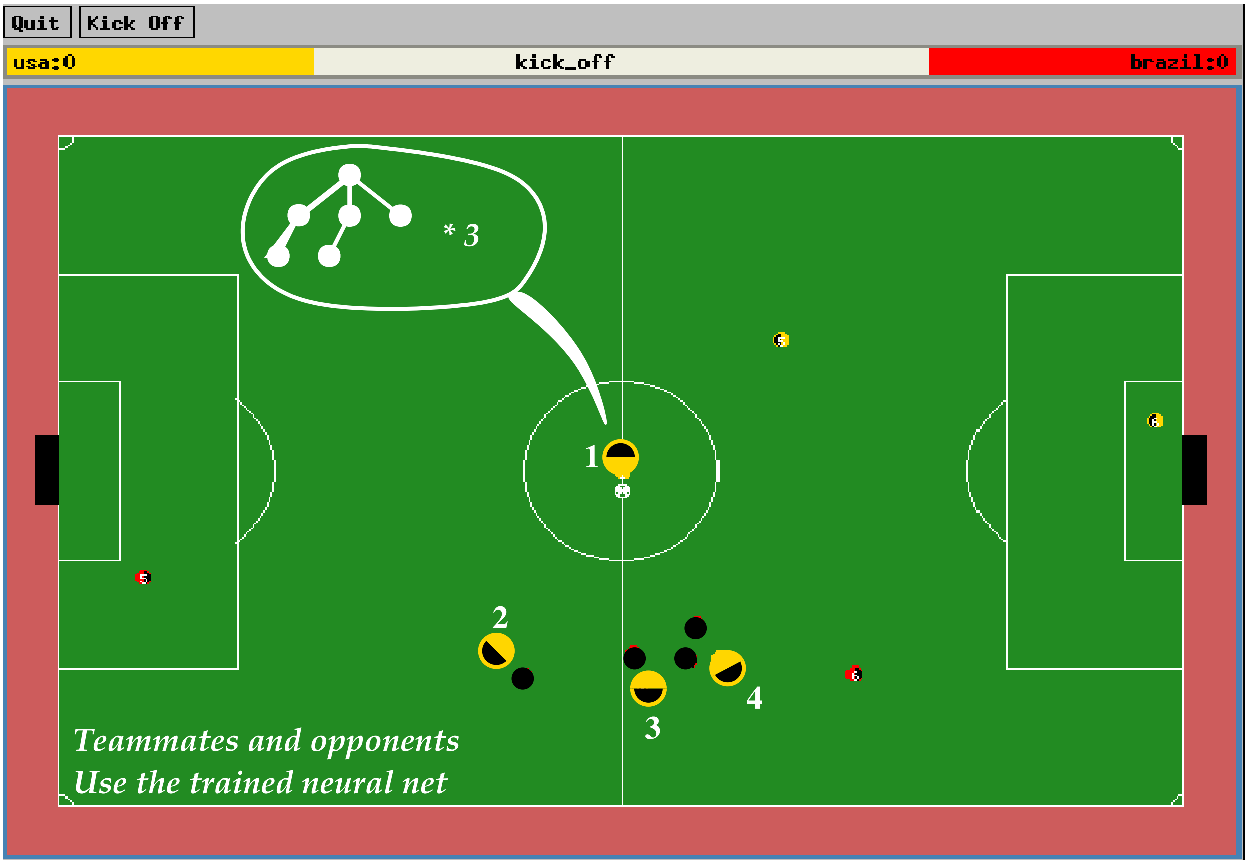
- While BD $>$ 14, TURN(BA)
- When BD $\leq$ 14, set TA = Random Angle between -45 and 45
- Record BD, BA, previous BD, ($\vec{F_1}$) and TA (O_1)
- TURN(BA + TA)
- DASH()
- Record result (from coach)

Comparing with other Methods

- A lookup table from $BallAng_t$ to $TurnAng$ based on the output of the trained NN.
- An analytic method based on the ball's vector motion.

Defender Behavior	Saves(%)	Goals(%)	Saves/(Goals+Saves) (%)
NN	**86**	9	**91**
Lookup Table	83	8	91
Analytic	82	13	86

Pass Evaluation



$\vec{F}_2$: 174 continuous state features, missing values

T_2: uses h_1

M_2: off-line using C4.5

O_2: success/failure prediction, confidence

h_2: success confidence $\geq .8 \Longrightarrow 80\%$ success

The DT Attributes

- Distance and Angle to the receiver (2);
- Distance and Angle to other teammates (up to 9) sorted by angle from the receiver (18);
- Distance and Angle to opponents (up to 11) sorted by angle from the receiver (22);
- Counts of teammates, opponents, and players within given distances and angles of the receiver (45);
- Distance and Angle from receiver to teammates (up to 10) sorted by distance (20);
- Distance and Angle from receiver to opponents (up to 11) sorted by distance (22);
- Counts of teammates, opponents, and players within given distances and angles of the passer from the receiver's perspective (45);

```
C4.5 [release 8] decision tree interpreter      Tue Mar 19 13:49:09 1996
------------------------------------------
Decision Tree:

passer opponents ang6 > 0 : F (420.0/131.1)
passer opponents ang6 <= 0 :
|   passer receiver distance <= 18.5 :
|   |   passer teammates dist4 ang8 <= 1 :
|   |   |   passer teammates dist12 ang8 <= 1 : S (320.0/73.7)
|   |   |   passer teammates dist12 ang8 > 1 :
|   |   |   |   passer opponent3 distance <= 17.6 : F (5.0/2.3)
|   |   |   |   passer opponent3 distance > 17.6 : S (14.0/1.3)
|   |   passer teammates dist4 ang8 > 1 :
|   |   |   passer opponents dist4 ang4 <= 0 : S (12.0/2.5)
|   |   |   passer opponents dist4 ang4 > 0 :
|   |   |   |   passer opponent1 distance <= 19.1 : F (2.0/1.0)
|   |   |   |   passer opponent1 distance > 19.1 : M (2.0/1.0)
|   passer receiver distance > 18.5 :
|   |   passer opponent1 angle <= 14 :
|   |   |   passer opponents dist12 ang4 <= 0 :
|   |   |   |   receiver players dist12 ang8 > 1 : F (16.7/7.6)
|   |   |   |   receiver players dist12 ang8 <= 1 :
|   |   |   |   |   passer opponent2 angle <= 22 : F (258.3/106.9)
|   |   |   |   |   passer opponent2 angle > 22 : S (65.1/32.8)
|   |   |   passer opponents dist12 ang4 > 0 :
|   |   |   |   receiver opponent3 distance <= 6.9 : F (25.6/13.8)
|   |   |   |   receiver opponent3 distance > 6.9 : S (69.4/32.0)
|   |   passer opponent1 angle > 14 :
|   |   |   receiver teammate1 distance <= 19.7 :
|   |   |   |   passer opponent1 angle <= 22 : S (85.0/28.9)
|   |   |   |   passer opponent1 angle > 22 :
|   |   |   |   |   receiver players dist8 ang12 > 1 : M (2.0/1.0)
```

Some Results

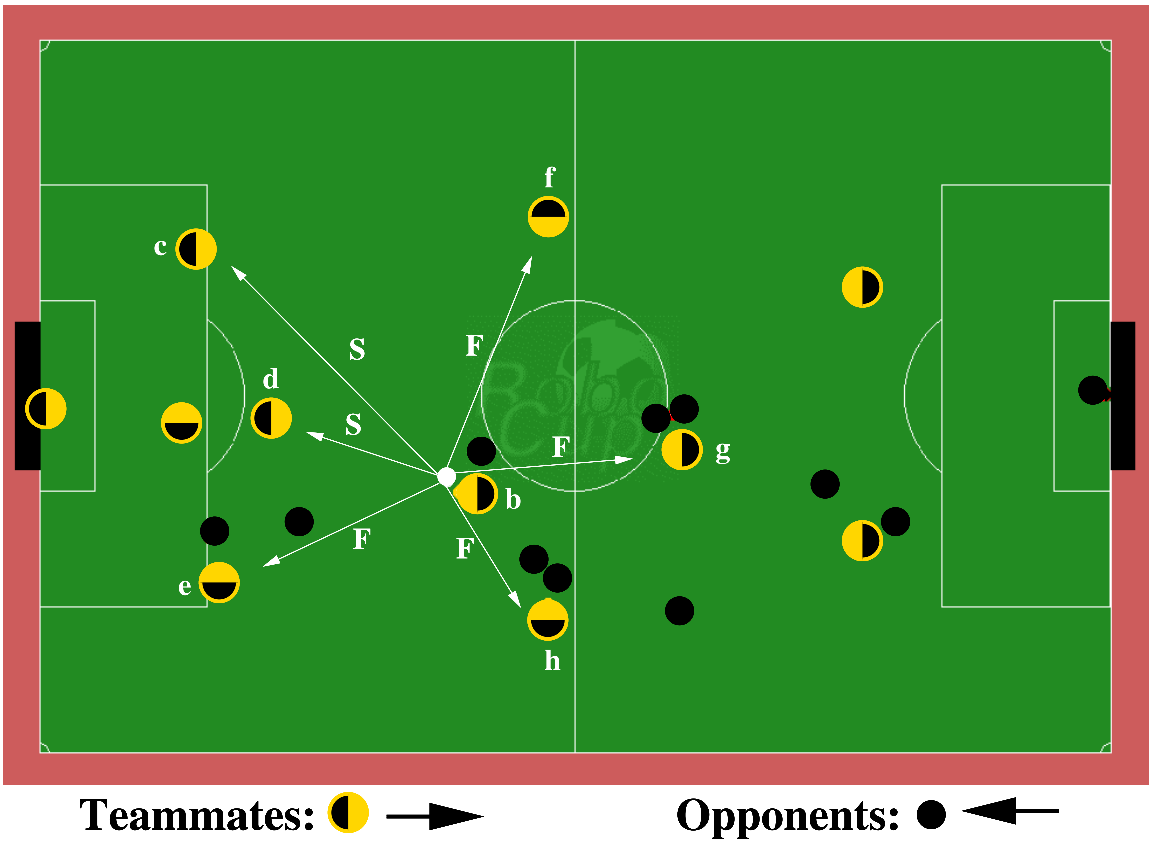
- Neural networks for learning ball-interception:

Training Examples	Saves(%)	Goals(%)	$\frac{\text{Saves}}{\text{Goals+Saves}}$(%)
100	57	33	63
200	73	18	80
300	81	13	86
400	81	13	86
500	84	10	89
750	**86**	9	**91**
1000	83	10	89
4773	84	9	90

- Decision trees for learning pass confidence:

Result	Overall	Success Confidence: .8–.9	.7–.8	.6–.7
(Number)	(5000)	(1050)	(3485)	(185)
SUCCESS (%)	65	**79**	63	58
FAILURE (%)	26	15	29	31
MISS (%)	8	5	8	10

Pass Selection: A Collaborative and Adversarial Policy



- Local information not good enough
- Need goal-oriented strategic information
- Depends on teammates, adversaries
- Necessarily on-line

Pass Selection

- $F_3 = \{PlayerPosition, O_2, O_2, O_2, \ldots\}$
  - 1 coarse geographical component
  - 1 action-dependent feature for each possible pass.
  - **The action-dependent features are from h_2**
- $O_3 = \{shoot\} \cup \{Teammates\}$
- T_3 = training examples gathered on-line by individual team members during real games
- M_3 = TPOT-RL: Team-Partitioned, Opaque-Transition RL
- h_3 = a distributed pass-selection policy

A General Learning Problem

- Multiple (n) agents in a **team**
  - common long-term goal
  - State space S, action space A
- **Opaque transitions**
  - No knowledge of state from teammates' perspectives
- Traditional RL intractable
  - Large state space/limited training
- Target concept may change
- **Action-dependent** features
  - Depend on current state *and* action

e.g. **robotic soccer** and **network routing**

Team-Partitioned, Opaque-Transition Reinforcement Learning: TPOT-RL

- Abstract S to a feature-space V
- Learn $Q : V \times A \mapsto \mathbb{R}$
- Partition state space among agents
  - $P : S \mapsto M$ with $|M| \geq$ # agents
- Use **action-dependent feature** function
  - $e : S \times A \mapsto U$
  - e.g. (soccer) e= DT, U= {Success,Failure}
- Feature space reduces policy domain
  - $f(s) = \langle P(s), e(s, a_0), \ldots, e(s, a_{n-1}) \rangle \in V$
  - $V = M \times U^{|A|} \quad \Longrightarrow \quad |V| = |M| * |U|^{|A|}$
- Assume $Q(f(s), a_i)$ depends only on $P(s)$,$e(s, a_i)$
  - **Each** agent learns $|M|/n * |U|^1 * |A|$ values

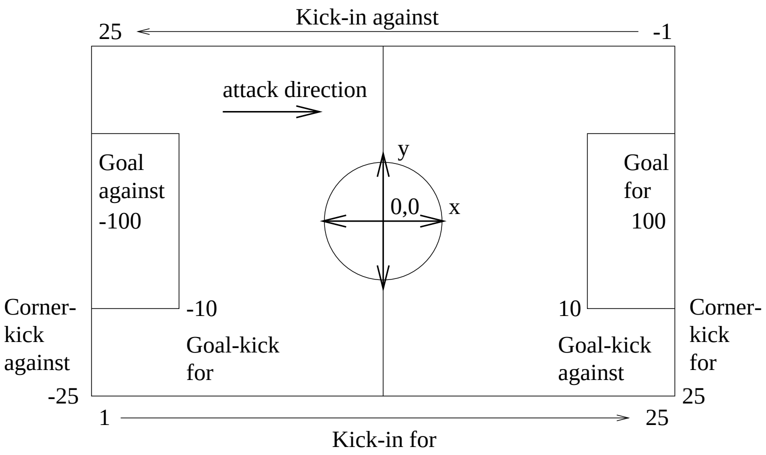
soccer:	$\|M\| = 11$	$n = 11$	$\|U\| = 2$	$\|A\| = 8$:	8 values each
routing:	$\|M\| = 12^2$	$n = 12$	$\|U\| = 2$	$\|A\| = 12$:	72 values each

Training

- Medium/long-term reward function
  $R : S^{t_{lim}} \mapsto \mathbb{R}$ — S from agent's perspective
- Action a in state s with $f(s) = v$ at time t_a
- Reward r at time t_r discounted based on $t_r - t_a$
- Learn $Q(v, a)$
  - $Q(v,a) := Q(v,a) + \alpha(r - Q(v,a))$
  - Decay values for shifting concept

Robotic soccer

- **True reward:** goal scored for/against — sparse
- **Intermediate reward:** observed ball trajectory



Soccer Results

Test Opponents

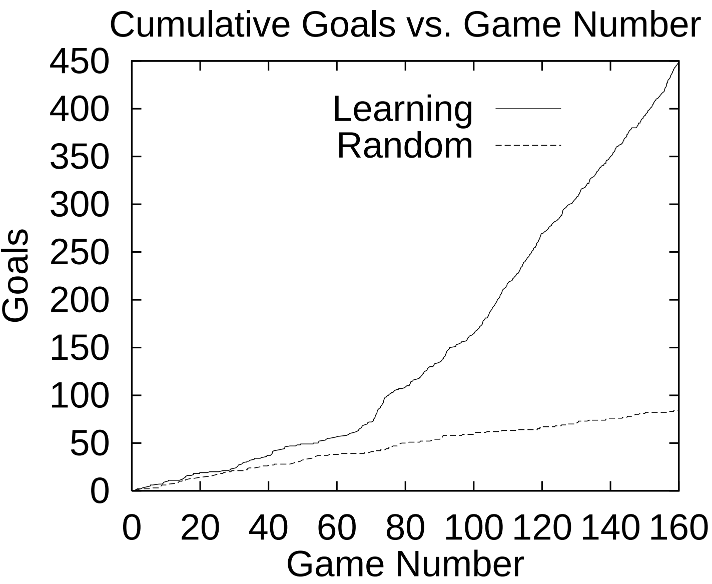
Random: pass to random teammate

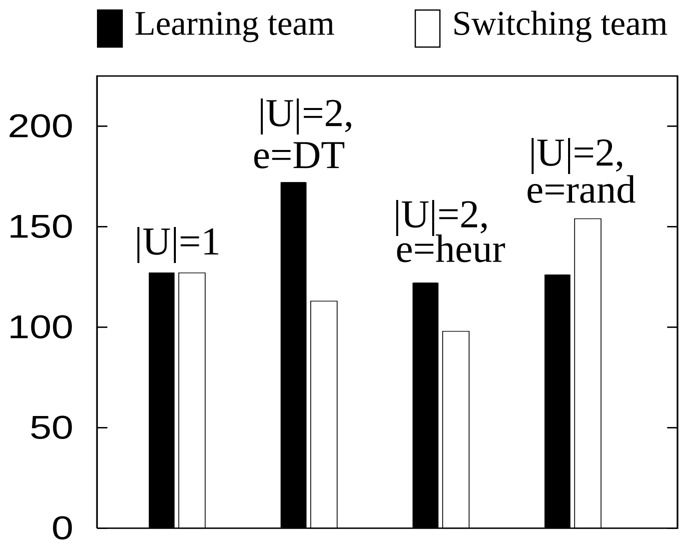
- otherwise identical to learning team

Switching: periodically switch sides of the field

- Fixed pass/shoot policies

Train on-line vs. fixed opponent





Peter Stone