Combining Inductive and Analytical Learning

- Why combine inductive and analytical learning?
- KBANN: prior knowledge to initialize the hypothesis
- TangentProp, EBNN: prior knowledge alters search objective
- FOCL: prior knowledge alters search operators

Inductive and Analytical Learning

Inductive learning	**Analytical learning**
Hypothesis fits data	Hypothesis fits domain theory
Statistical inference	Deductive inference
Requires little prior knowledge	Learns from scarce data
Syntactic inductive bias	Bias is domain theory

What We Would Like

Inductive learning	Analytical learning
Plentiful data	Scarce data
No prior knowledge	Perfect prior knowledge

- General purpose learning method:
- No domain theory → learn as well as inductive methods
- Perfect domain theory → learn as well as PROLOG-EBG
- Accommodate arbitrary and unknown errors in domain theory
- Accommodate arbitrary and unknown errors in training data

Domain Theory

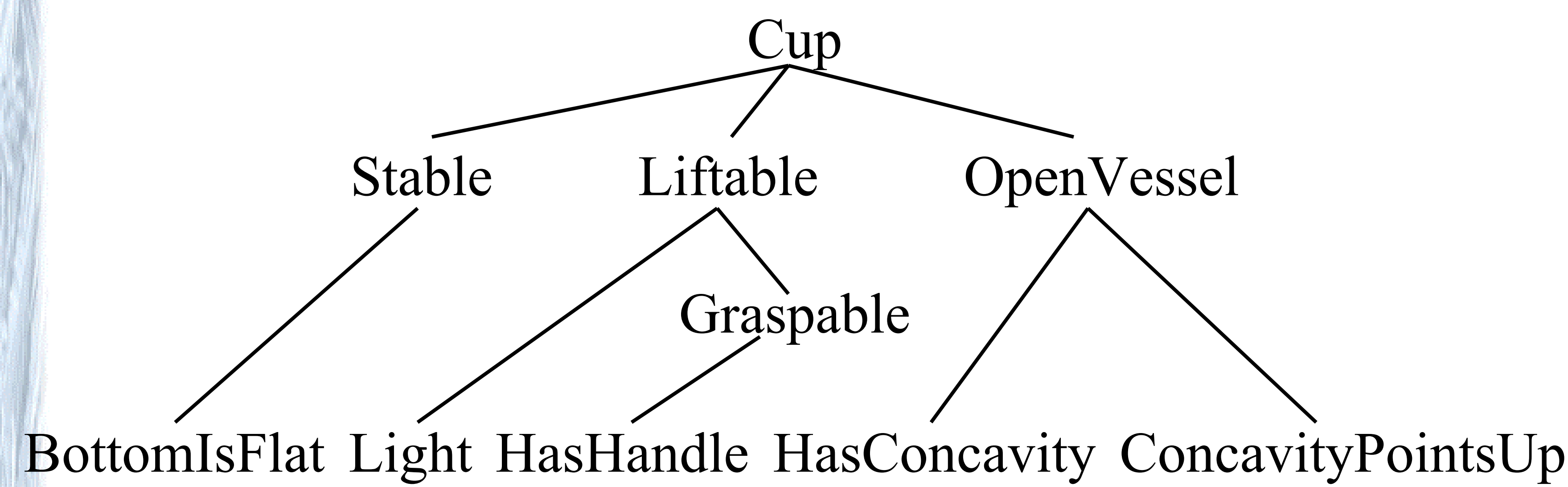
Cup ← Stable, Liftable, OpenVessel

Stable ← BottomIsFlat

Liftable ← Graspable, Light

Graspable ← HasHandle

OpenVessel ← HasConcavity, ConcavityPointsUp



Training Examples

	Cups				Non-Cups					
BottomIsFlat	√	√	√	√	√	√	√			√
ConcavityPointsUp	√	√	√	√	√		√	√		
Expensive	√		√				√		√	
Fragile	√	√			√	√		√		√
HandleOnTop					√		√			
HandleOnSide	√			√					√	
HasConcavity	√	√	√	√	√		√	√	√	√
HasHandle	√			√	√		√		√	
Light	√	√	√	√	√	√	√		√	
MadeOfCeramic	√				√		√	√		
MadeOfPaper				√					√	
MadeOfStyroForm		√	√			√				√

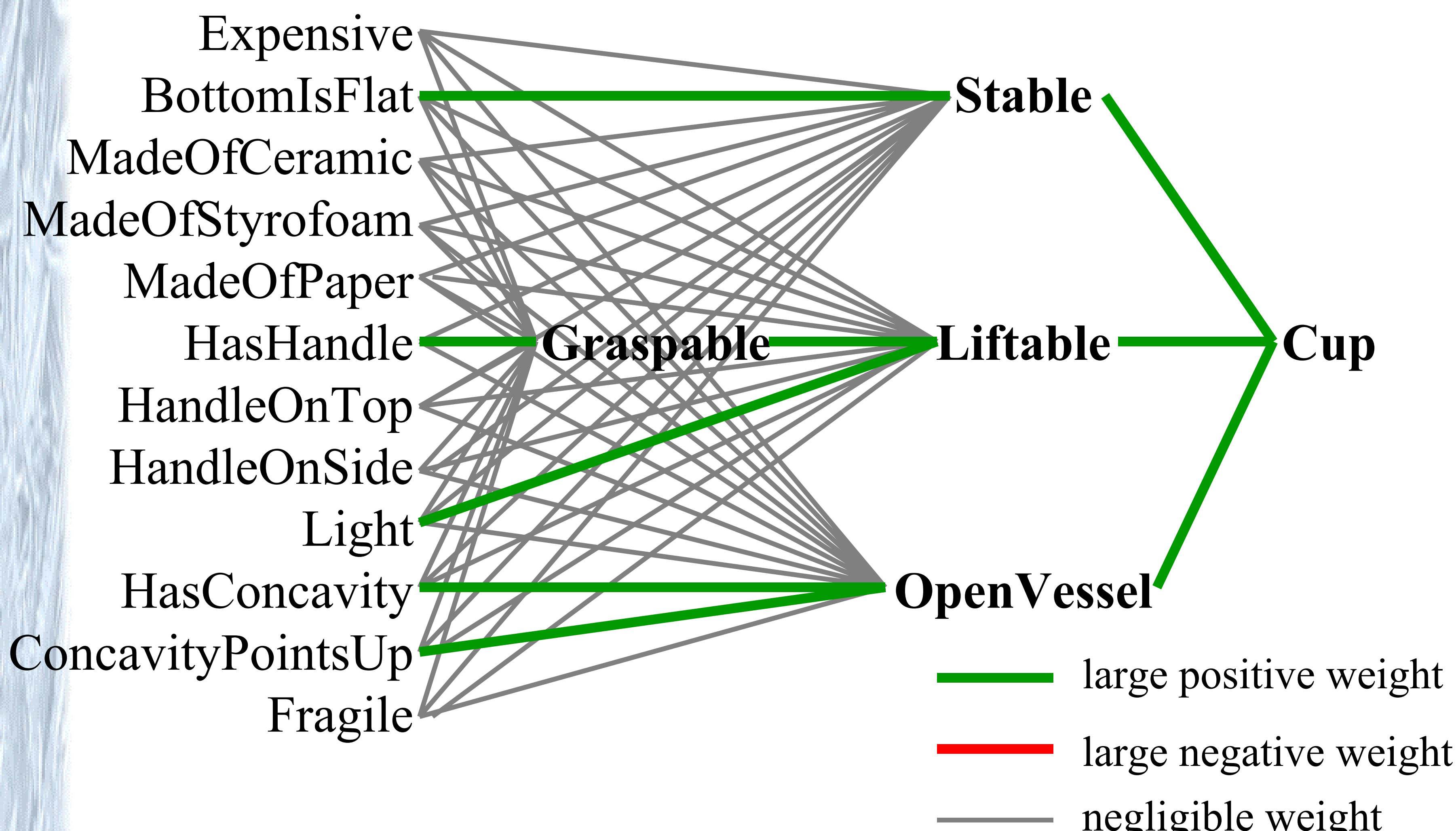
KBANN

Knowledge Based Artificial Neural Networks

KBANN (data *D*, domain theory *B*)

1. Create a feedforward network *h* equivalent to *B*
2. Use BACKPROP to tune *h* to fit *D*

Neural Net Equivalent to Domain Theory



Creating Network Equivalent to Domain Theory

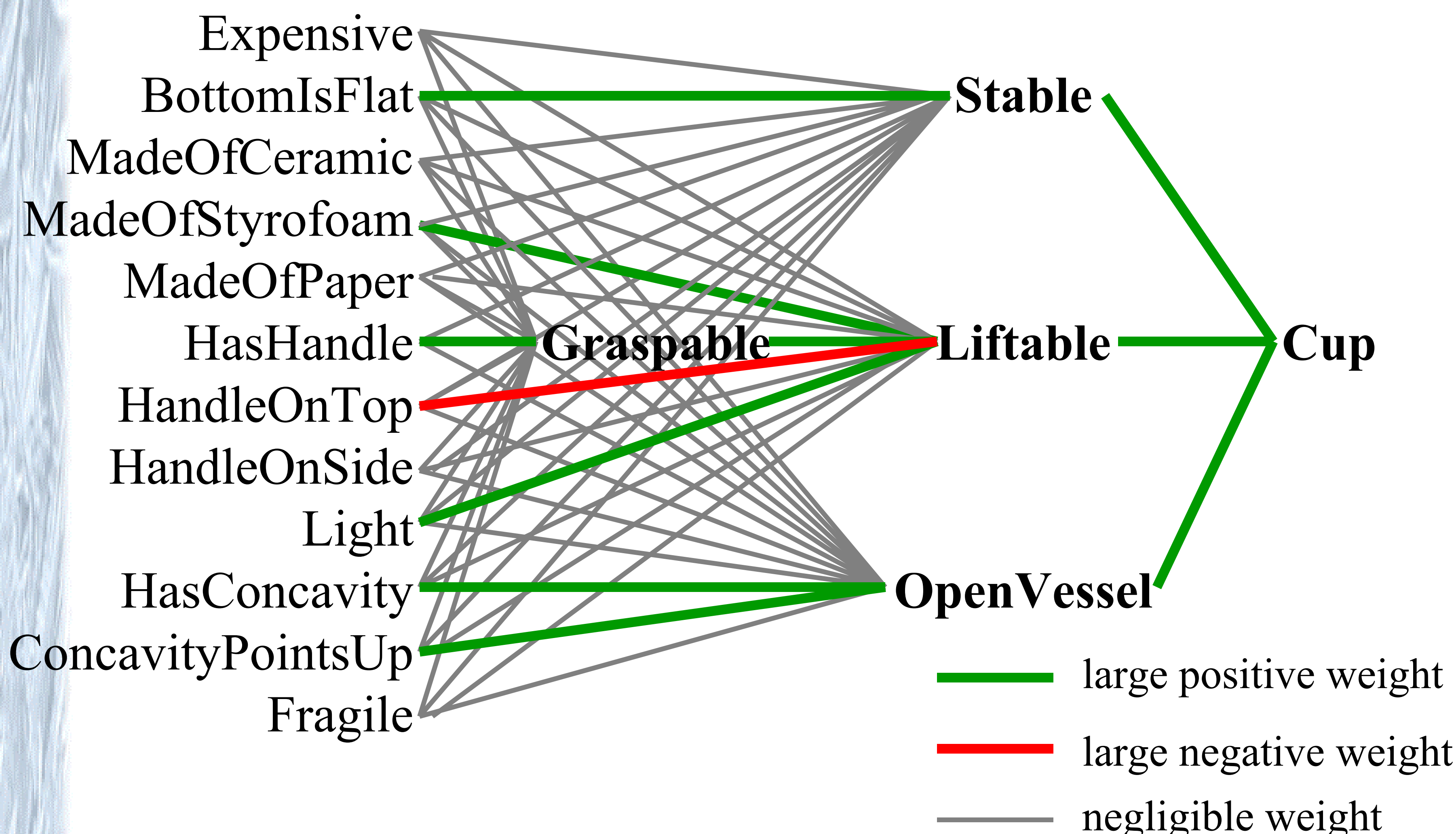
Create one unit per horn clause rule (an AND unit)

- Connect unit inputs to corresponding clause antecedents
- For each non-negated antecedent, corresponding input weight $w \leftarrow W$, where W is some constant
- For each negated antecedent, weight $w \leftarrow -W$
- Threshold weight $w_0 \leftarrow -(n - .5)\ W$, where n is number of non-negated antecedents

Finally, add additional connections with near-zero weights

$$Liftable \leftarrow Graspable, \neg Heavy$$

Result of Refining the Network



KBANN Results

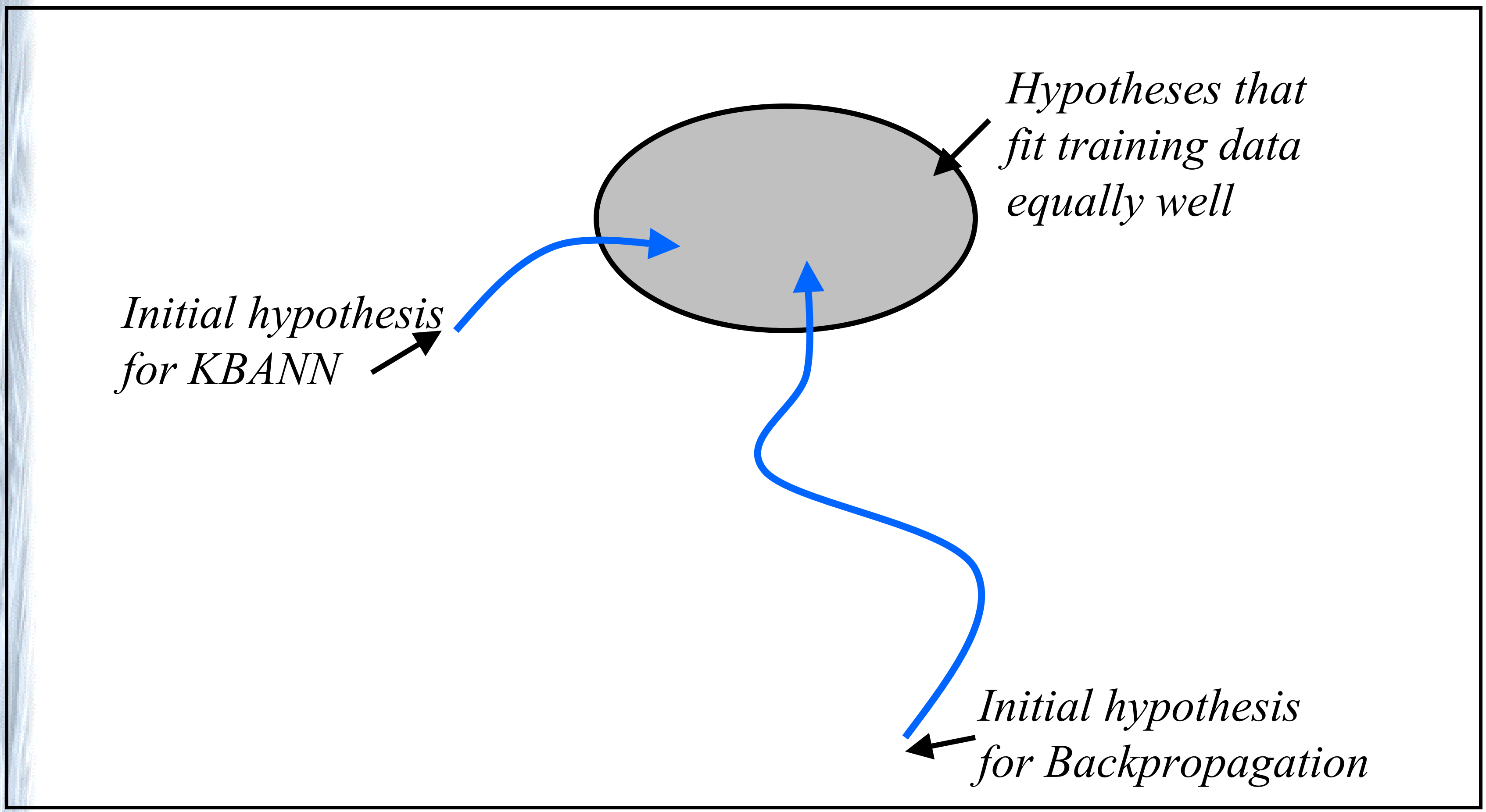
Classifying promoter regions in DNA (leave one out testing):

- Backpropagation: error rate 8/106
- KBANN: 4/106

Similar improvements on other classification, control tasks.

Hypothesis Space Search in KBANN

Hypothesis Space



EBNN

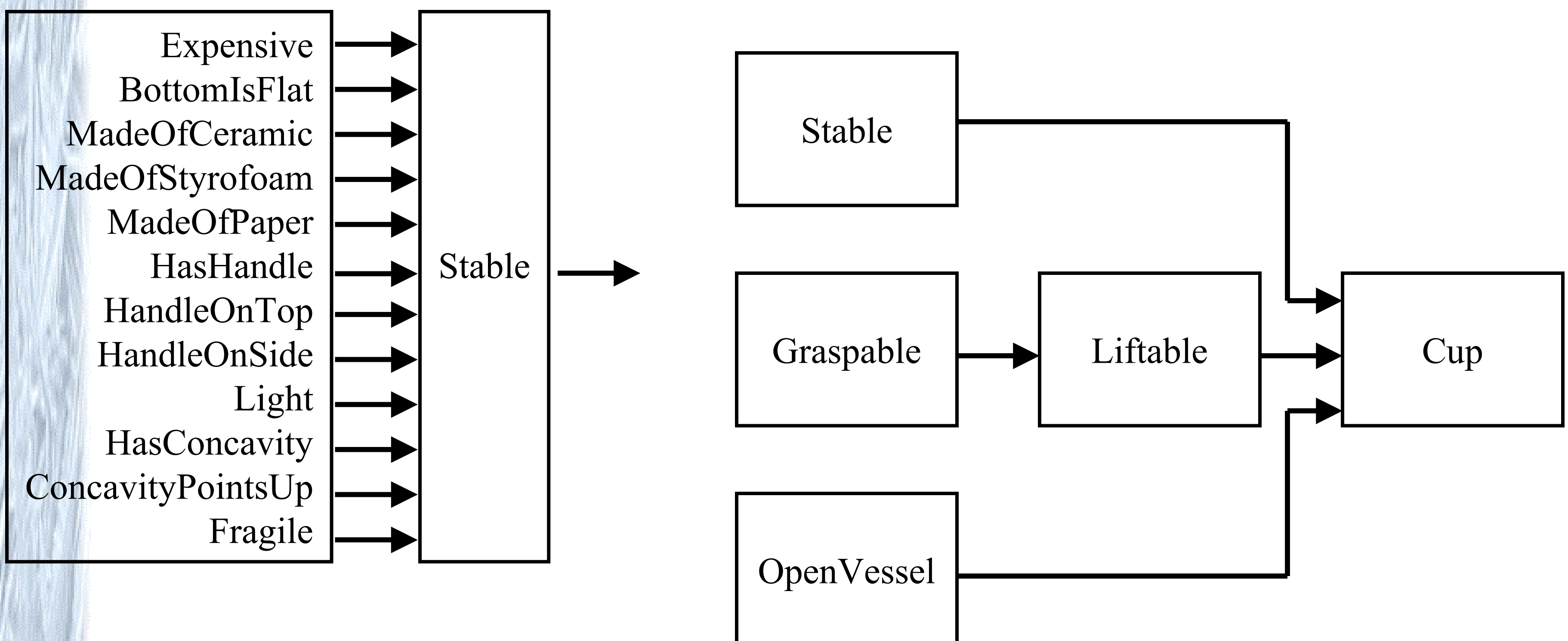
Explanation Based Neural Network

Key idea:

- Previously learned approximate domain theory
- Domain theory represented by collection of neural networks
- Learn target function as another neural network

Explanation in Terms of Domain Theory

Prior learned networks for useful concepts combined into a single target network



TangetProp

Assume x, $f(x)$ and $\left.\frac{\partial f(x)}{\partial x}\right|_{x_i}$ provided as input

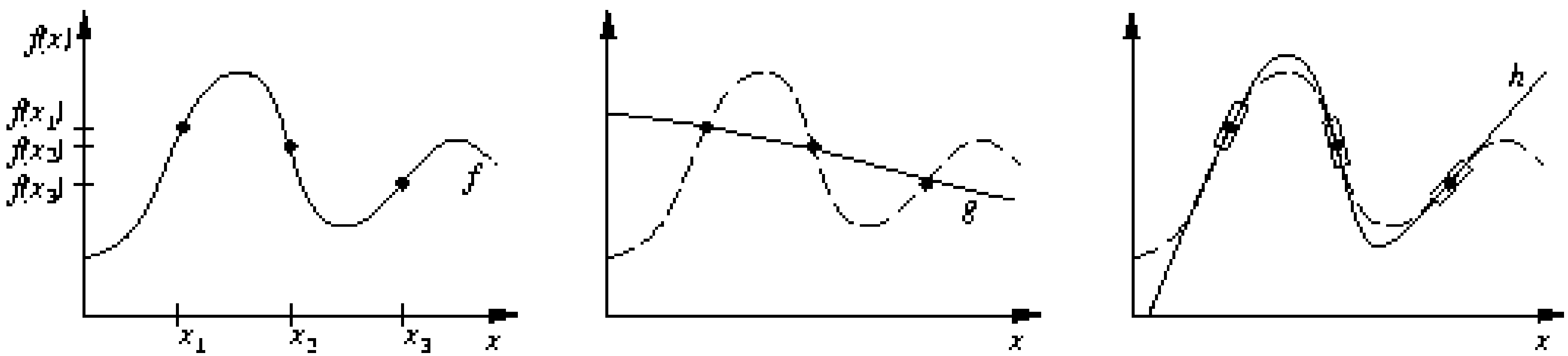
Modified objective for gradient descent :

$$E = \sum_i \left[(f(x_i) - \hat{f}(x_i))^2 + \mu_i \sum_j \left(\frac{\partial A(x)}{\partial x^j} - \frac{\partial \hat{f}(x)}{\partial x^j} \right)^2_{(x=x_i)} \right]$$

where

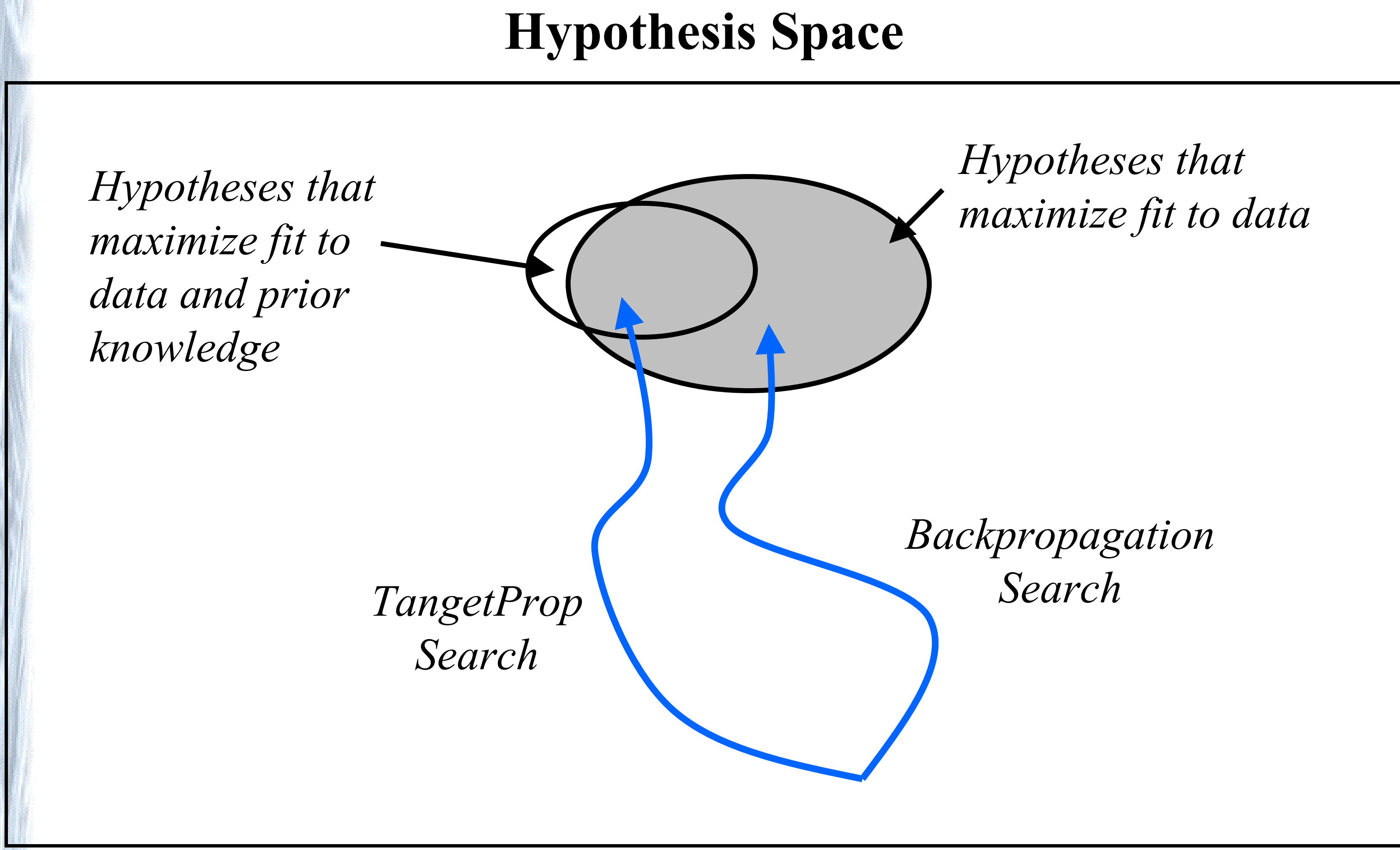
$$\mu_i \equiv 1 - \frac{|A(x_i) - f(x_i)|}{c}$$

- $f(x)$ is target function
- $\hat{f}(x)$ is neural net approximation to $f(x)$
- $A(x)$ is domain theory approximation to $f(x)$



f(x)
f(x1)
f(x2)
f(x3)
x1
x2
x3
x
f
x
g
h
x

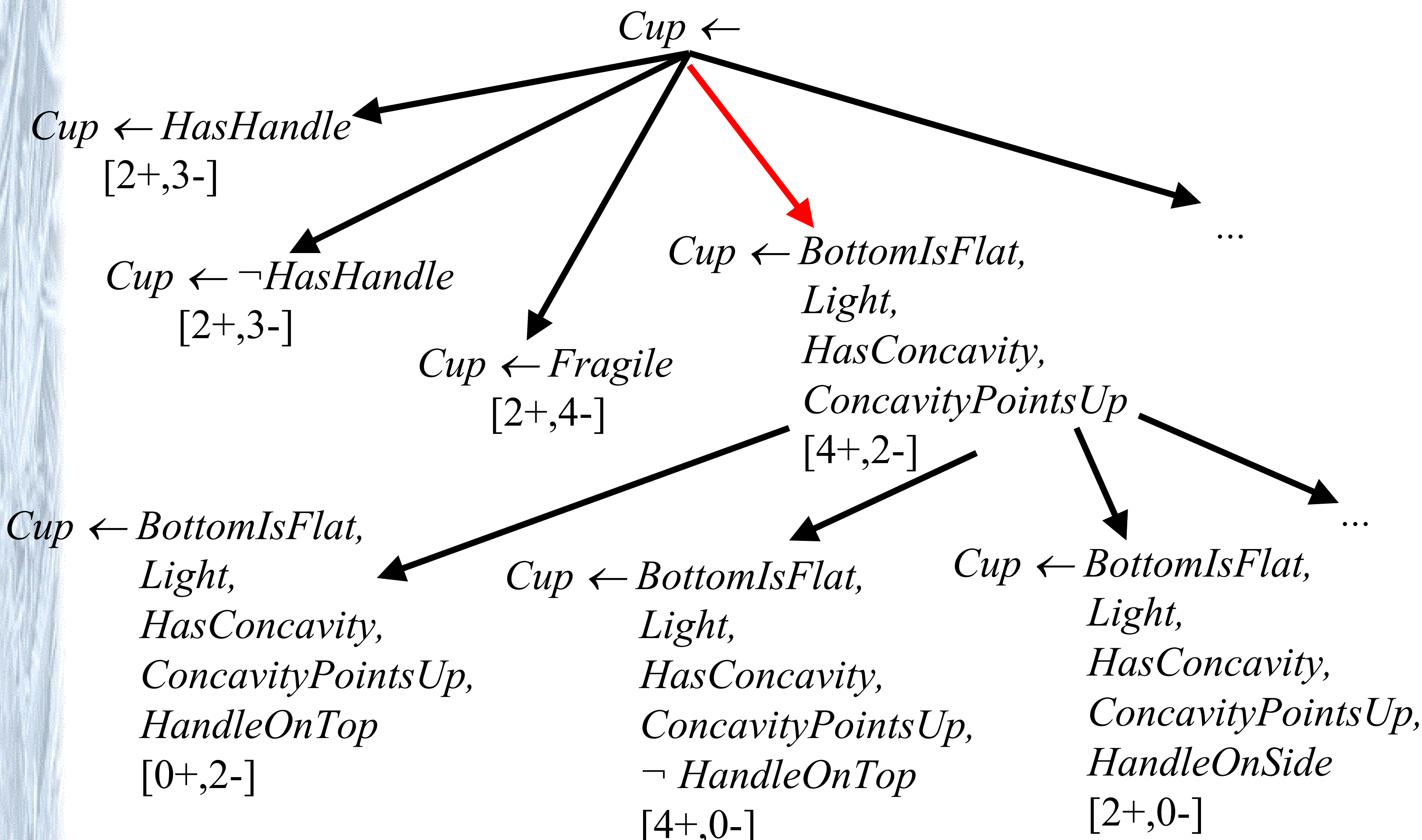
Hypothesis Space Search in TangentProp



FOCL

- Adaptation of FOIL that uses domain theory
- When adding a literal to a rule, not only consider adding single terms, but also think about adding terms from domain theory
- May also prune specializations generated

Search in FOCL



FOCL Results

Recognizing legal chess endgame positions:

- 30 positive, 30 negative examples
- FOIL: 86%
- FOCL: 94% (using domain theory with 76% accuracy)

NYNEX telephone network diagnosis

- 500 training examples
- FOIL: 90%
- FOCL: 98% (using domain theory with 95% accuracy)