Learned Prioritization for Trading Off Accuracy and Speed

Jiarong Jiang* Adam Teichert** Hal Daumé III* Jason Eisner**
*University of Maryland College Park **Johns Hopkins University

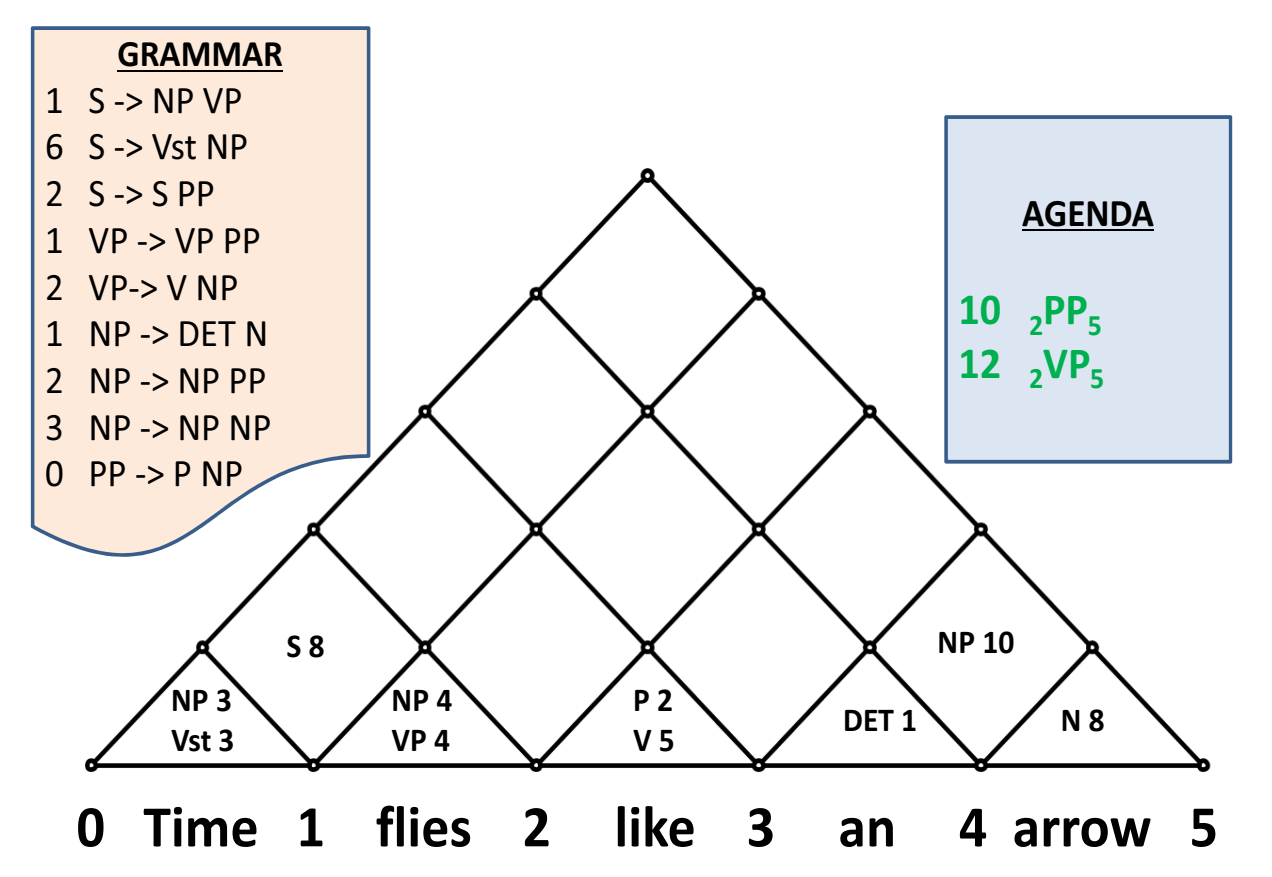
Take Home Summary

- **Main Objective:** fast and accurate structured prediction (search)
- **Search Method:** agenda-based dynamic programming
- **Knob To Tune:** prioritization heuristic
  - **Bad:** try different known heuristics by hand :(
  - **Good:** **learn** a heuristic for your input distribution, grammar, and speed/accuracy needs
- **How?:** **hybrid reinforcement/apprenticeship learning!**

Agenda Based Parsing

- **Goal**: find most likely parse w.r.t. a grammar
- Extend already built partial parses
- Reuse work via dynamic programming
- Extend **most promising** partial solutions first via agenda

```
1. dequeue some update according to its priority from the agenda, say $(Y, i, j) \leftarrow 75$
2. update chart$[Y, i, j]$ to 75
3. for each constituent adjacent to $(Y, i, j)$, such as $(Z, j, k)$
4.    for each grammar rule $X \rightarrow YZ$ that can combine $(Y, i, j)$ with $(Z, j, k)$
5.       let $new \leftarrow chart[Y, i, j] + chart[Z, j, k] + score(X \rightarrow YZ)$
6.       if $new > chart[X, i, k]$
7.          then enqueue $(X, i, k) \leftarrow$ priority function$([X, i, k])$ on the agenda
```



- An A* heuristic would be **exact** but **too slow**:

learn to trade a little accuracy for speed!

Agenda Based Parsing as a Markov Decision Process

- **State Space**: current chart and agenda
- **Action**: choose a partial parse from agenda
- **Transitions**: given the chosen action, deterministically updates chart and builds and pushes other partial (or full) parses to agenda
- **Reward**: $\mathbf{accuracy} - \lambda \cdot \mathbf{time}$
  e.g. Accuracy = labeled span recall, Time = # of pops from agenda
- **Policy**: deterministically pops highest-**priority** available action: $\pi_\theta(s) = \arg\max_a \theta \cdot \phi(a, s)$

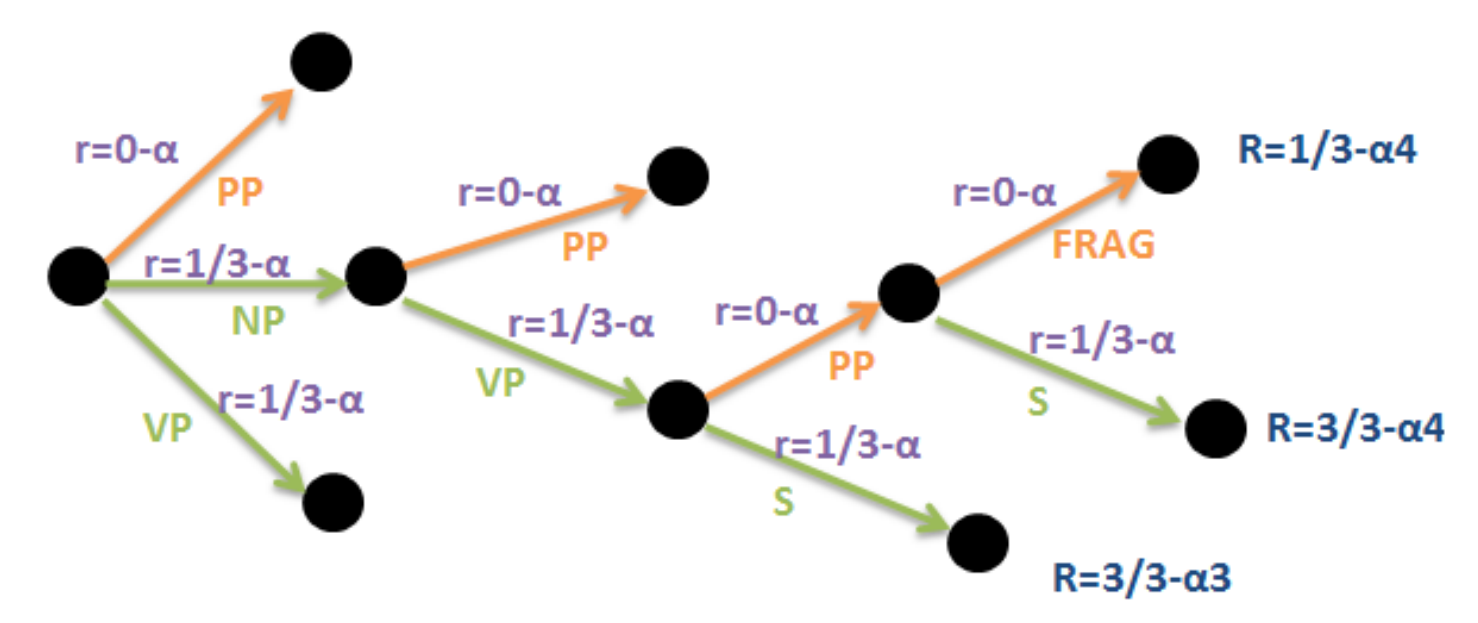
learning a policy = learning the priority function

Speed and Accuracy in Agenda Based Parsing

- All experiments on Penn Treebank WSJ (sentence length ≤ 15)
- **Development data**:
  - Grammar: Berkeley latent variable PCFG trained on sections 2-20
  - Training (if any): 100 sentences from section 21
  - Test: Evaluated on same 100 sentences
- **Results on development data for baselines**:
  - **Exhaustive Search (CKY order): Recall = 93.3, Relative # of pops = 3.0x**
  - **A*parser with a 0 heuristic function: Recall = 93.3, Relative # of pops = 1.0x**
  - **A*parser with a 0 heuristic function with pruning (A^*_0): Recall = 92.0, Relative # of pops = 0.33x**

Policy Gradient with Reward Shaping

- Weakness of vanilla **policy gradient with Boltzmann exploration**:
  No attempt to determine which actions were **responsible** for a trajectory's reward



- **Reward Shaping** $\Rightarrow$ fast convergence

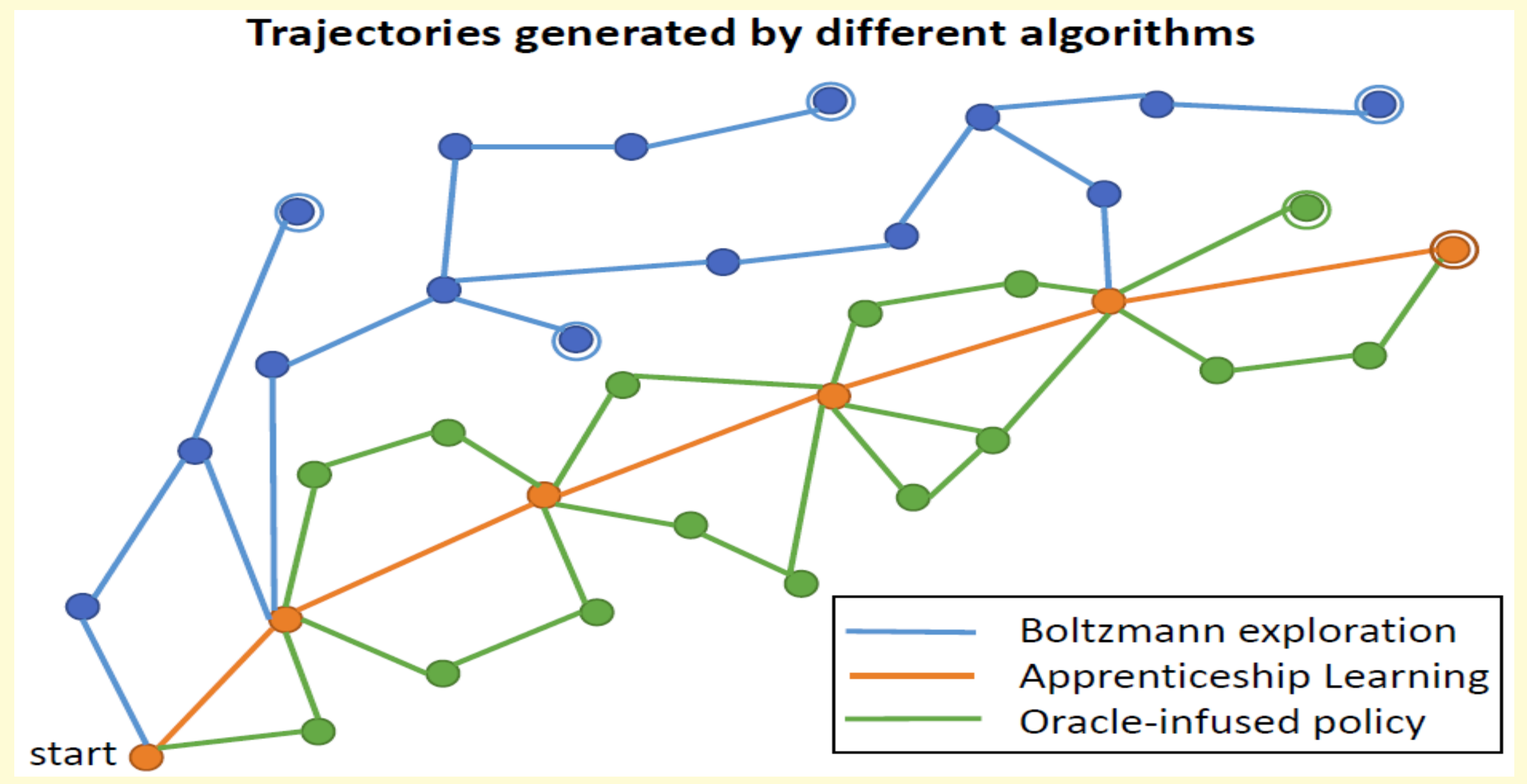
$$\tilde{r}(s, a) = \begin{cases} \delta(a) - \lambda & \text{if } a \text{ is a full parse tree} \\ 1 - \lambda & \text{if } a \text{ is in the true parse} \\ -\lambda & \text{otherwise} \end{cases}$$

$\delta(s)$: a negative reward for actions which received early reward for constituents that were not in the final parse.

Result on development data: Recall = 56.4, Relative # of pops = 0.46x

state space $>>$ # of reasonable trajectories

Solution: Oracle-Infused Policy Gradient



- **Oracle action**: action that leads to a maximum-reward tree
- **Apprenticeship learning via classification:**
  following oracle trajectories = training a supervised log-linear classifier
- **Result on development data: Recall = 84.2, Relative # of pops = 0.85x**

too hard to imitate oracle with our features

- **Oracle-infused policy**

$$\pi^+_\delta(a \mid s) = \delta\pi^*(a \mid s) + (1 - \delta)\pi(a \mid s)$$

  - $\delta = 0.8^{\text{epoch}}$
  - epoch: the current number of passes made through the training set
- **Result on development data: Recall = 91.2, Relative # of pops = 0.46x**

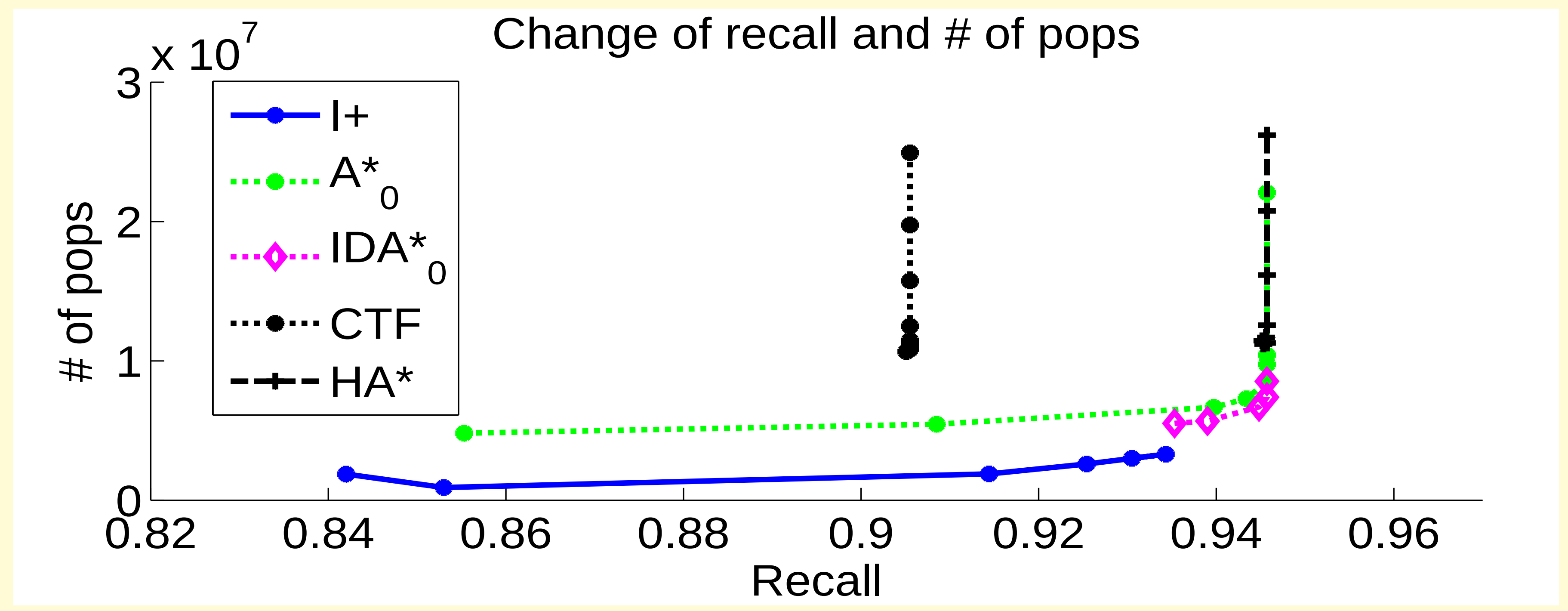
Solution: explore near oracle $\xrightarrow{slowly}$ explore near learned policy

Features

1. Viterbi inside score
2. Touches start of sentence?
3. Touches end of sentence?
4. Width of partial parse
5. Ratio of width to sentence length
6. log p(label | prev POS) and log p(label | next POS)
7. Case pattern of {preceding, following, initial} word in partial parse
8. Punctuation pattern in partial parse (five most frequent)

Final Experiments

- Data:
  - Grammar: Berkeley latent variable PCFG trained on sections 2-21
  - Evaluation: on section 23
- Baselines:
  - **(HA*)** a Hierarchical A*parser [3] with same pruning threshold at each level
  - **(A^*_0)** an A*parser with a 0 heuristic function and pruning
  - **(IDA^*_0)** an iterative deepening A^*_0 algorithm
  - **(CTF)** an agenda-based coarse-to-fine parser [4].
- Our method: Oracle-infused policy gradient with immediate reward **(I+)** trained on section 22



Related Work

1. H. Daumé III, J. Langford, and D. Marcu. 2009. Search-based structured prediction. Machine Learning, 75(3):297—C325.
2. V. Gullapalli and A. G. Barto. 1992. Shaping as a method for accelerating reinforcement learning. In Proceedings of the IEEE International Symposium on Intelligent Control.
3. A. Pauls and D. Klein. 2009. Hierarchical search for parsing. In NAACL/HLT.
4. S. Petrov and D. Klein. 2007. Improved inference for unlexicalized parsing. In NAACL/HLT.
5. S. Ross, G. J. Gordon, and J. A. Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In AI-Stats.