Automatic Value Function Refinement and Unrefinement for Relational Reinforcement Learning

Mitchell Keith Bloch

University of Michigan
2260 Hayward Street
Ann Arbor, MI. 48109-2121
bazald@umich.edu

June 9, 2016

Carli ≈ Soar – https://github.com/bazald/carli

What's offered:

- A Soar-like execution cycle

Meaning:

1. `^io.input-link`
2. ~~elaboration cycle~~
3. numeric preferences (and implicit operator proposal)
4. decide
   - ~~impasses~~
5. act

Carli $\neq$ Soar – https://github.com/bazald/carli

What's present:

- Soar-RL-like reinforcement learning support
- Architectural support for efficiently creating more specific RL-rules over time – a generative model for a value function

What's missing or different:

- Manipulating WMEs from the RHS has not been tested
- Operators (as you know them) and impasses do not exist
- Chunking, SMem, EpMem, and SVS do not exist

What's New?

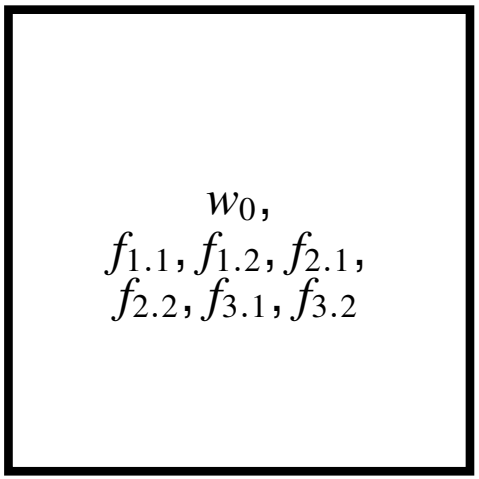
A Sneak Peak:

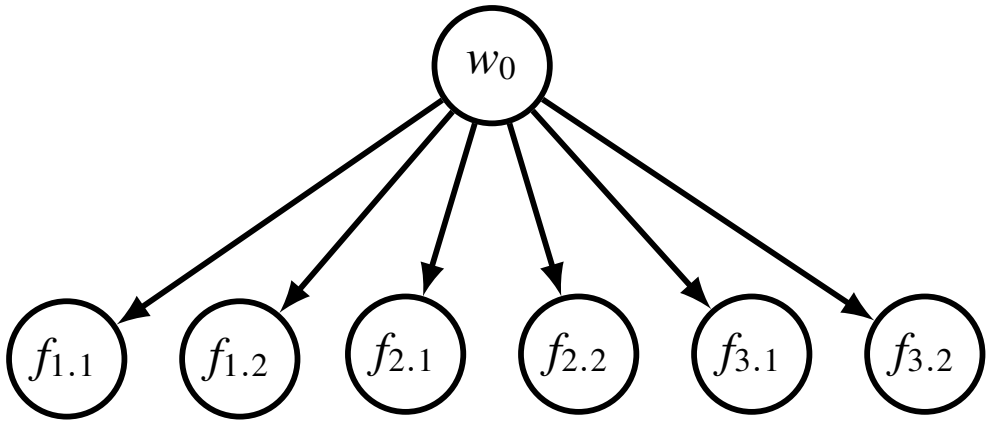
- All refinement criteria have corresponding unrefinement criteria.
- Results demonstrating that rerefinement:
  - Can save CPU time
  - Can reduce regret
  - Can improve the terminal policy

Reinforcement Learning

- Goals:
  - Approximate $\pi(s, a)$, the optimal target policy
  - Minimize regret to the extent possible
- How:
  - Build up the set of features, $\phi(i)$
    - Rules are specialized/despecialized
  - Update the weights, $\theta(i)$ using GQ(λ) during ϵ-greedy exploration
    - The value function is refined/unrefined
  - Use incremental algorithms for efficient, stable computation times

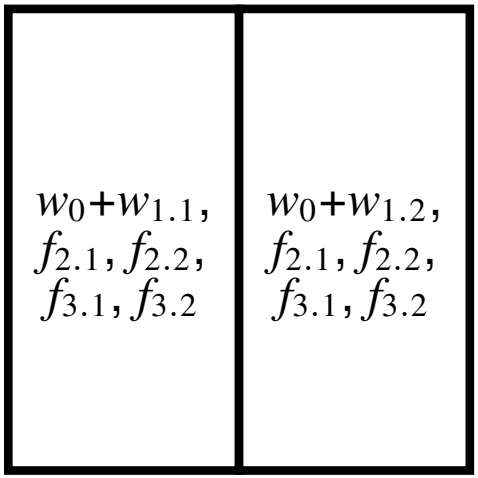
Value Function Refinement Over Time

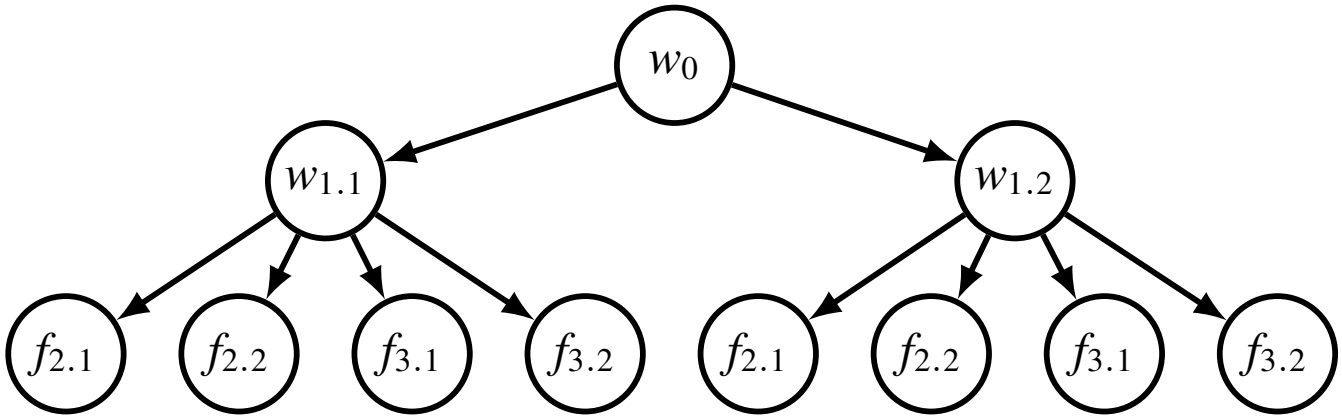




Before any refinement

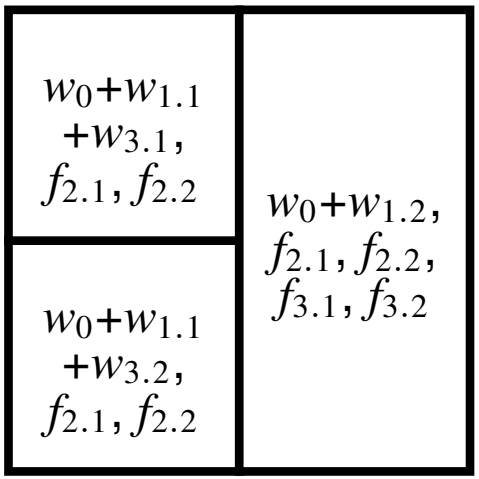
Value Function Refinement Over Time

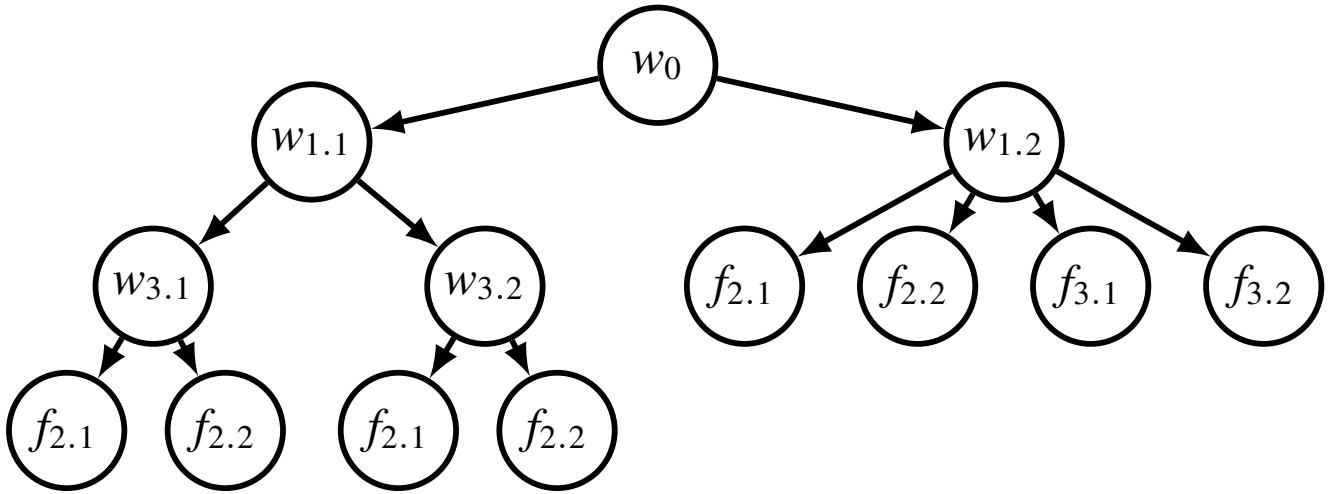




After one refinement

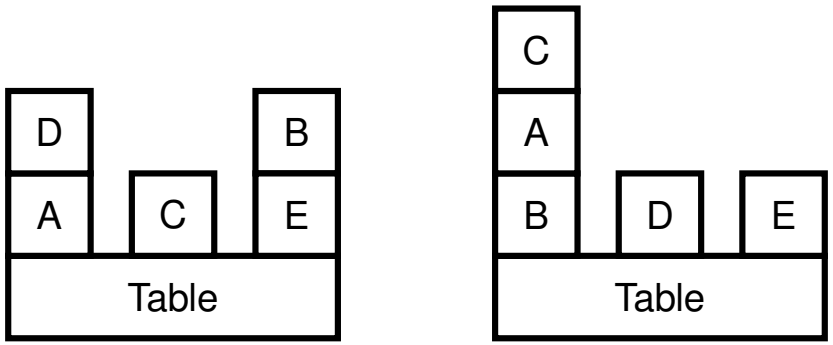
Value Function Refinement Over Time





After two refinements

Relational Blocks World



Blocks Goal

Complete state description visual

- Full representation of the goal presented by the environment
- Allows variable goals and numbers of blocks for different episodes
- Significantly more complex training goal
  - Must test more than one relation

A Carli Agent Rule

```
sp {blocks-world*rl-fringe*u16
   :feature 3 unsplit blocks-world*rl-fringe*s3
   (<s> ^blocks <blocks>)
   (<s> ^goal <goal>)
   ⋮                                  # Rule abbreviated
   -{(<goal> ^stack <goal-stack>)
     (<stack> ^matches <goal-stack>)}


-->
   = 0.32900469057018422 17
}
```

A Successor Carli Agent Rule

```
sp {blocks-world*rl-fringe*u38
   :feature 3 unsplit blocks-world*rl-fringe*s16
   (<s> ^blocks <blocks>)
   (<s> ^goal <goal>)
   ⋮                                  # Rule abbreviated
   -{(<goal> ^stack <goal-stack>)
     (<stack> ^matches <goal-stack>)}
   +{(<goal> ^stack <goal-stack>)
     (<dest-stack> ^matches <goal-stack>)}
-->
   = 0.0
}
```

Dynamic Refinement for RRL

- Each state is described by a set of relations, such as `(<stack> ^top <block>)`
- Each RL-rule / feature in $\phi(i)$ represents a conjunction of any number of such relations
- Given $\phi(i)$, $\theta(i)$, and other metadata, which features are most likely to improve the value function?
- We've explored the following criteria:
  - Cumulative Absolute Temporal Difference Error
  - Policy – Maximal change in $\pi(s, a)$
  - Value – Maximal change in $\theta(i)$
- We've done so with:
  - No unrefinement
  - Unlimited rerefinement
  - Rerefinement with blacklists
  - Rerefinement with boosts for previous choices

Cumulative Absolute Temporal Difference Error

- Focus on regions of high activity and error.
- Track TD error experienced at each leaf node in the value function.
- The nodes with highest error are eligible for specialization when their features match.
- Despecialization:
  - Test whether error experienced at the internal “fringe” node is greater than that accumulated by the child nodes.

Policy Criterion

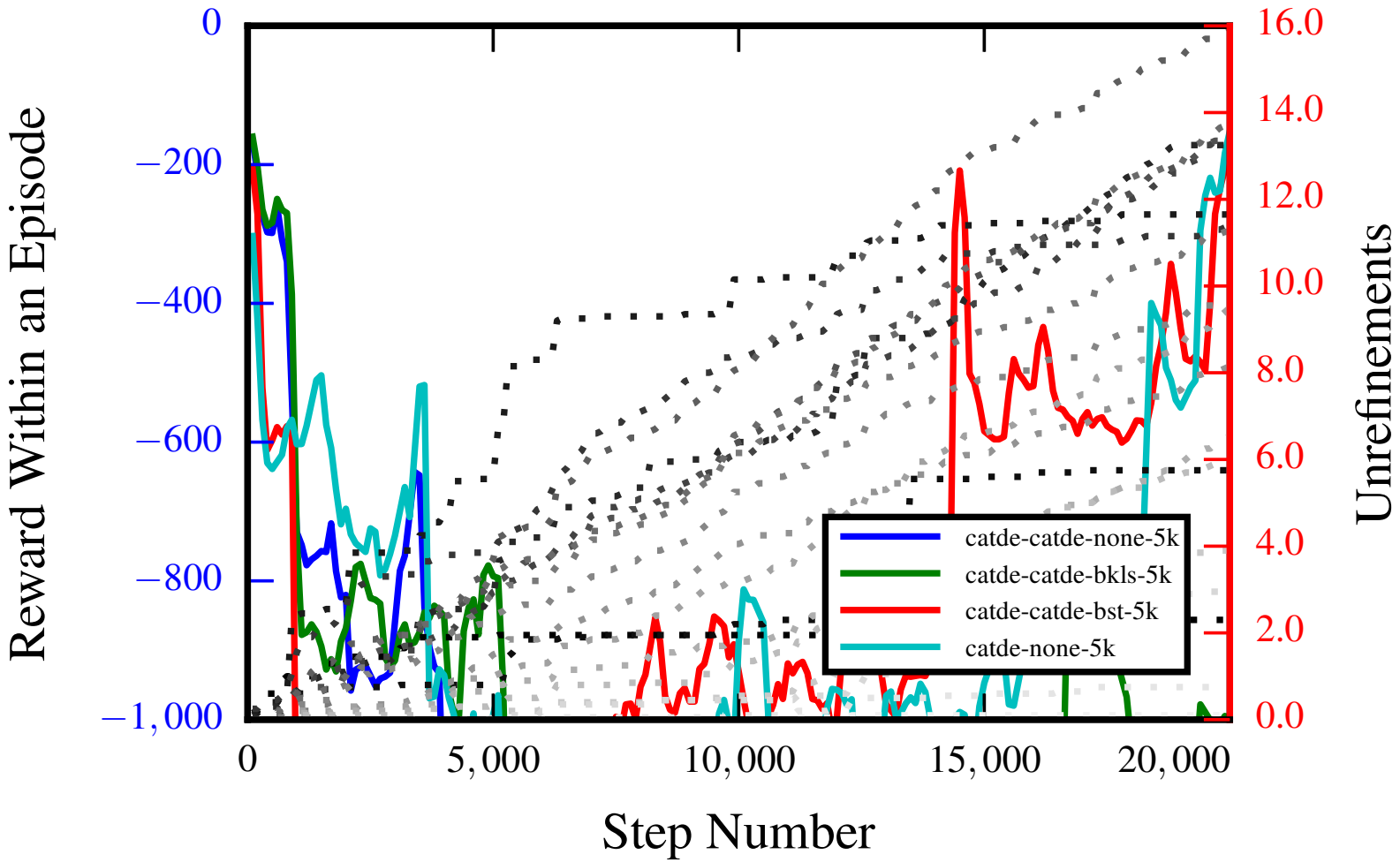
- Focus on modifying policy (Whiteson 2007)
- Choose features which maximize the change in the greedy set of actions.
- Despecialization:
  - Simulate criterion as through no further refinement had been done.
  - Evaluate whether a feature that was not chosen results in a larger change in the greedy set of actions.

Value Criterion

- Focus on improving value estimates (Whiteson 2007)
- Choose features which maximize value spread on specialization.
- Despecialization:
  - Simulate criterion as through no further refinement had been done.
  - Evaluate whether a feature that was not chosen creates a wider spread than the actual value function.

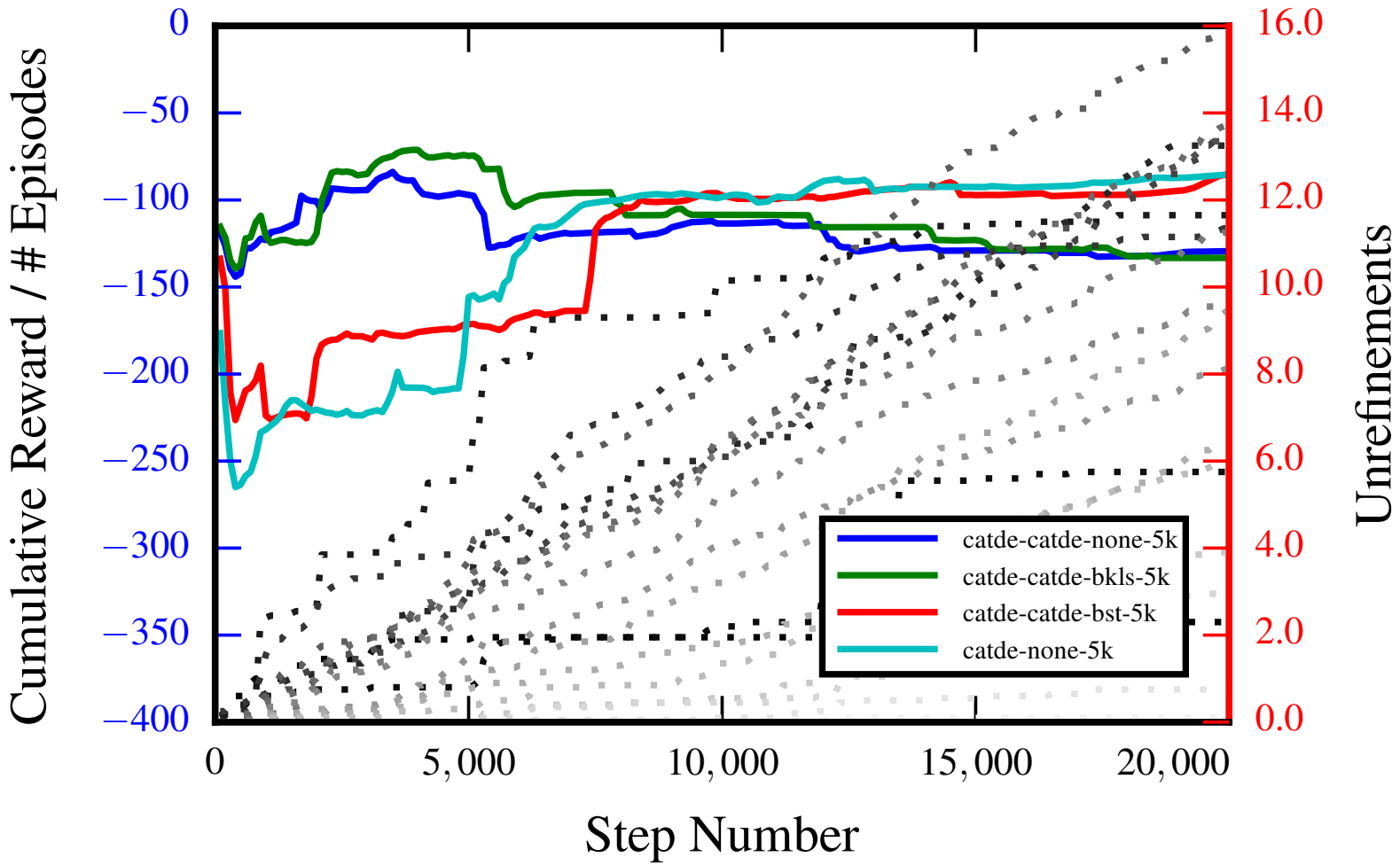
Cumulative Absolute Temporal Difference Error

- Metric focuses on regions of high activity and error
- No unrefinement and boost are best



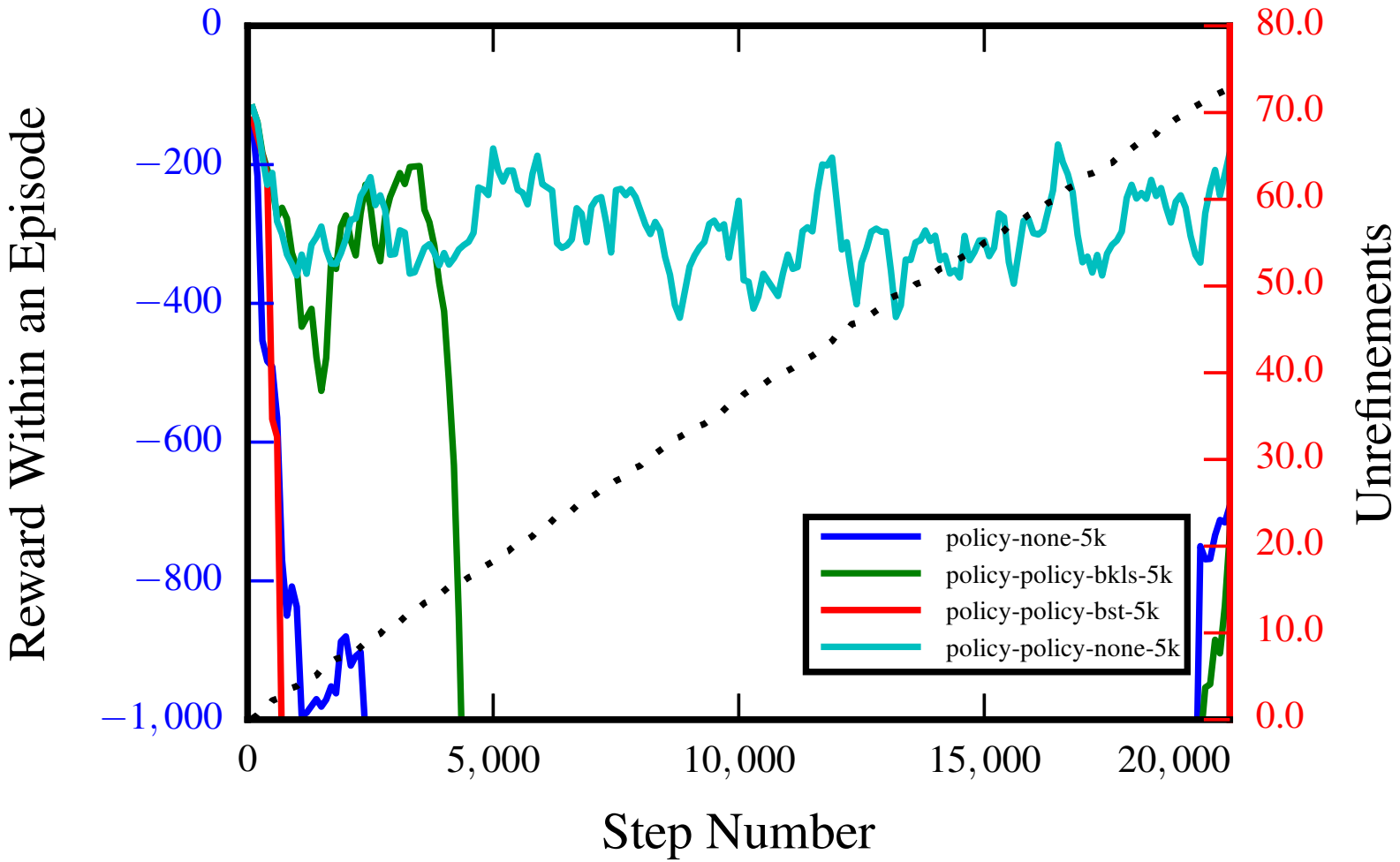
Cumulative Absolute Temporal Difference Error

- Metric focuses on regions of high activity and error
- No unrefinement and boost are best



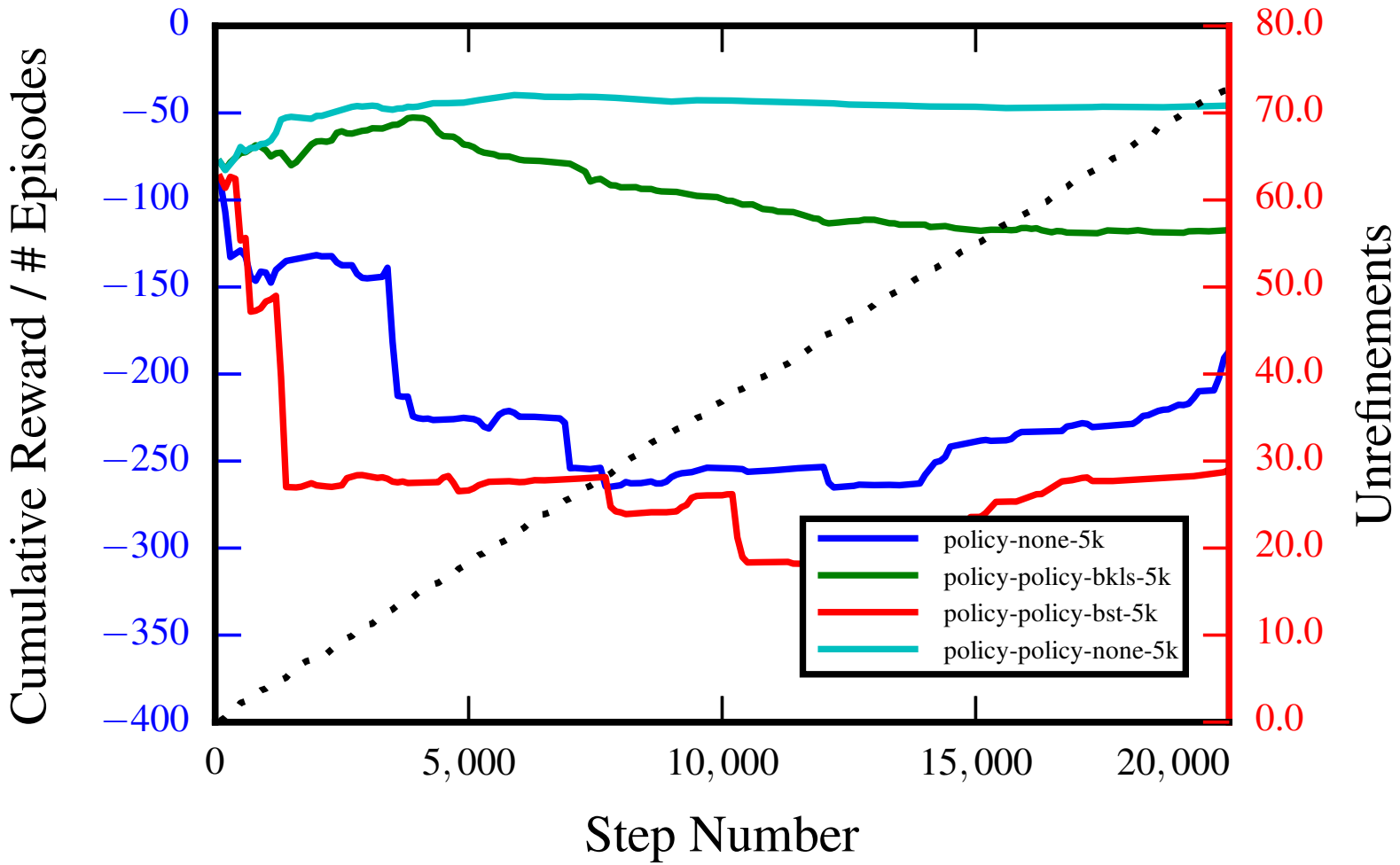
Policy Criterion

- Focus on policy difference (Whiteson 2007)
- Unlimited rerefinement is best



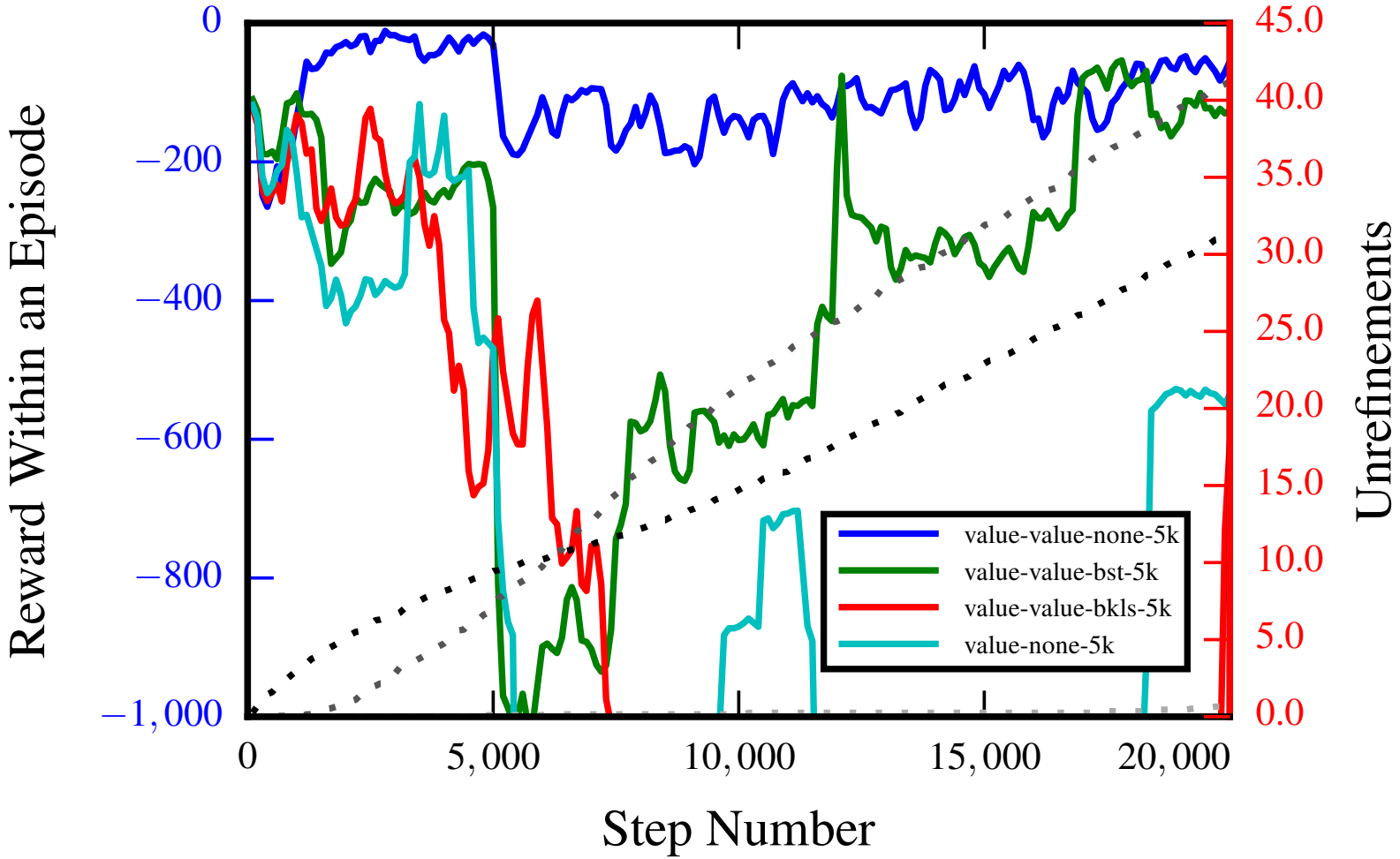
Policy Criterion

- Focus on policy difference (Whiteson 2007)
- Unlimited rerefinement is best



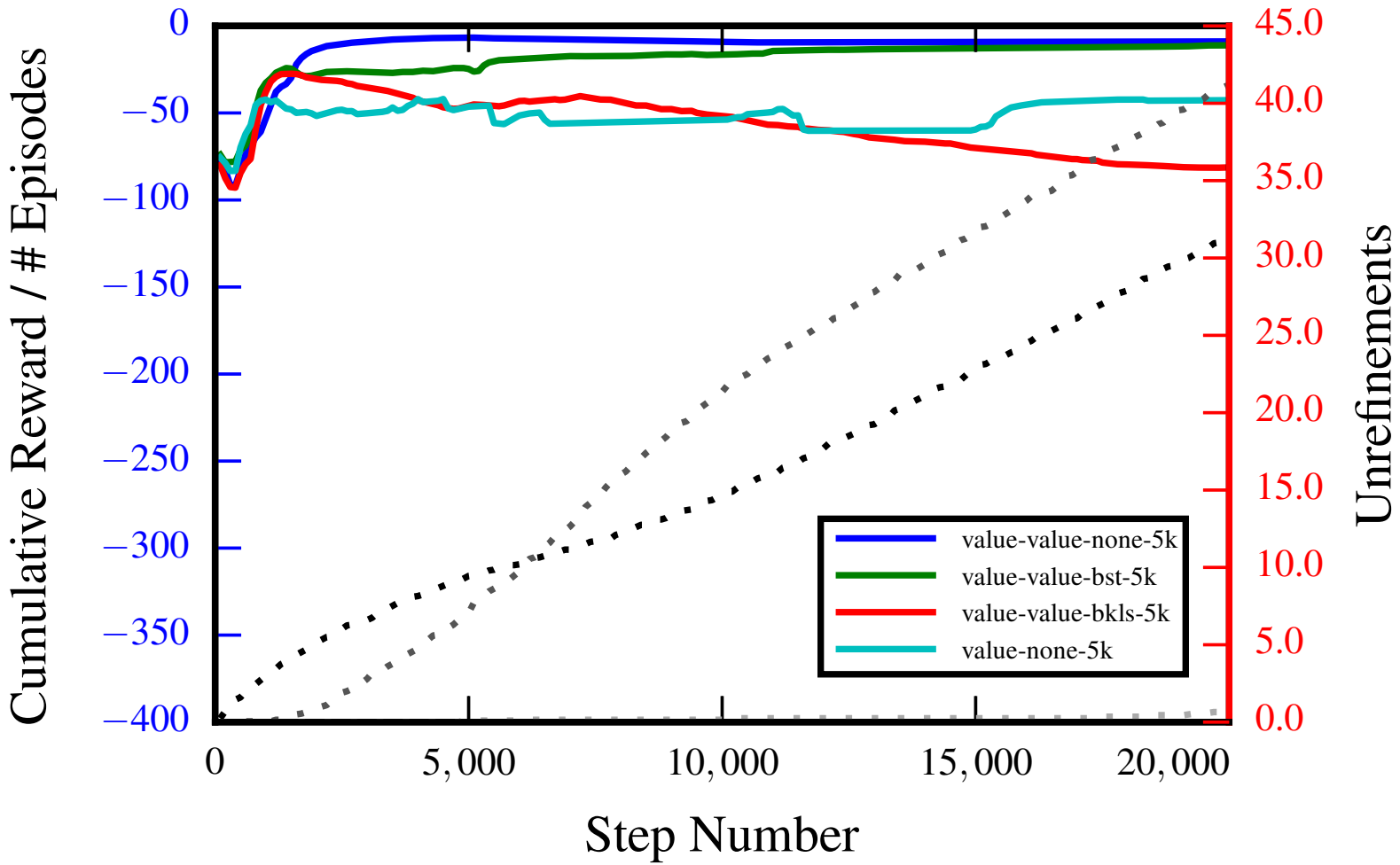
Value Criterion

- Focus on improving value estimates (Whiteson 2007)
- Unlimited rerefinement and boost are best



Value Criterion

- Focus on improving value estimates (Whiteson 2007)
- Unlimited rerefinement and boost are best



Analysis

Unrefinement helps with two of the three criteria.

Boost helps both CATDE and Value.

Boost's hurting Policy casts doubt on it as a criterion. Perhaps the dependency between the number of actions and environmental relations makes it unsuitable for relational reinforcement learning.

Boost is expensive without some new optimization.

CPU Time	No Unrefinement	Unrestricted	Blacklisting	Boosting
CATDE	40.9	148.3	146.3	167.2
Policy	25.0	15.2	21.8	61.9
Value	36.2	20.8	21.6	115.0

Nuggets and Coal

Nuggets:

- Rerefinement is implemented!
- It can save CPU time and improve performance.
- I can finish my thesis now.

Coal:

- Internal “fringe” nodes necessary for policy and value unrefinement criteria hurt performance.
- Boost hurts performance even more without providing additional guarantees about the terminal policy.
- The policy criterion may unsuitable for relational reinforcement learning.