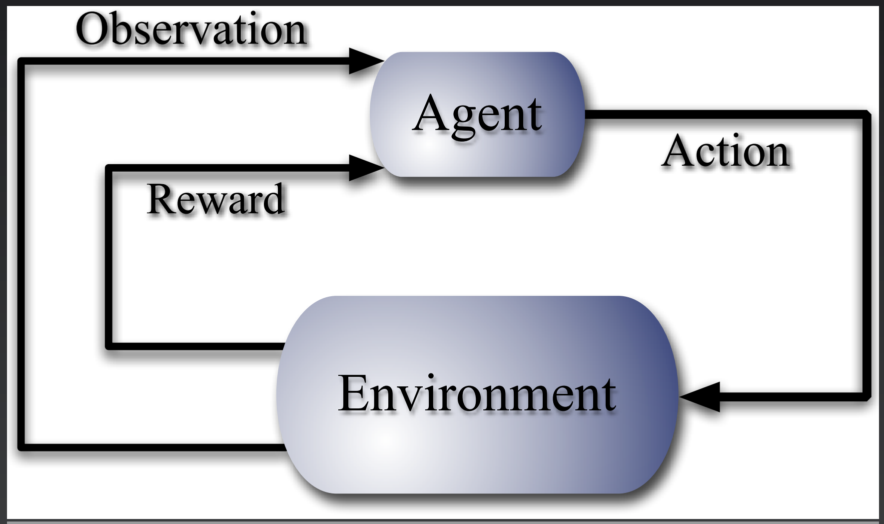
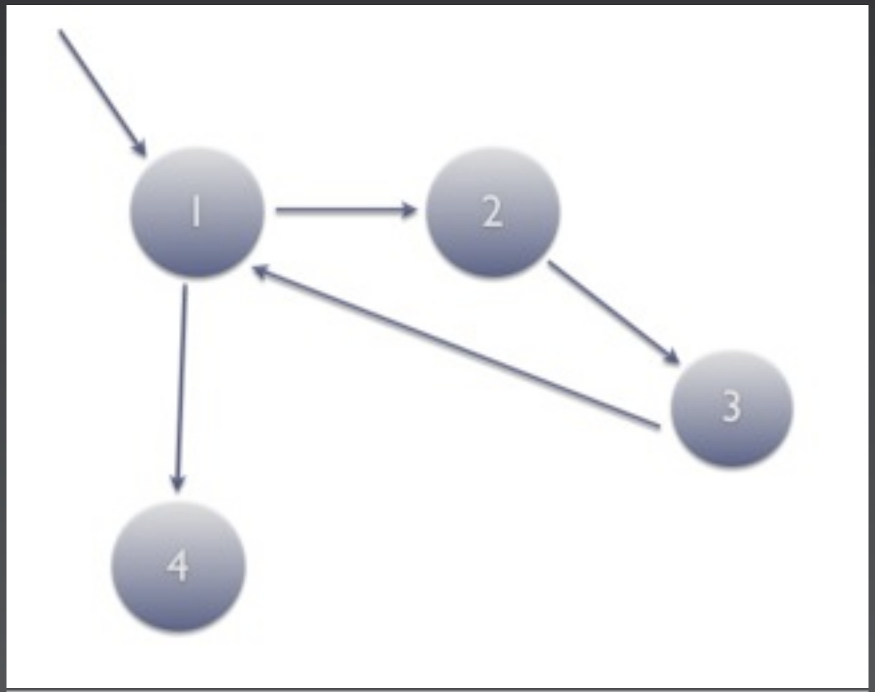
Incremental Least-Squares Temporal Difference Learning

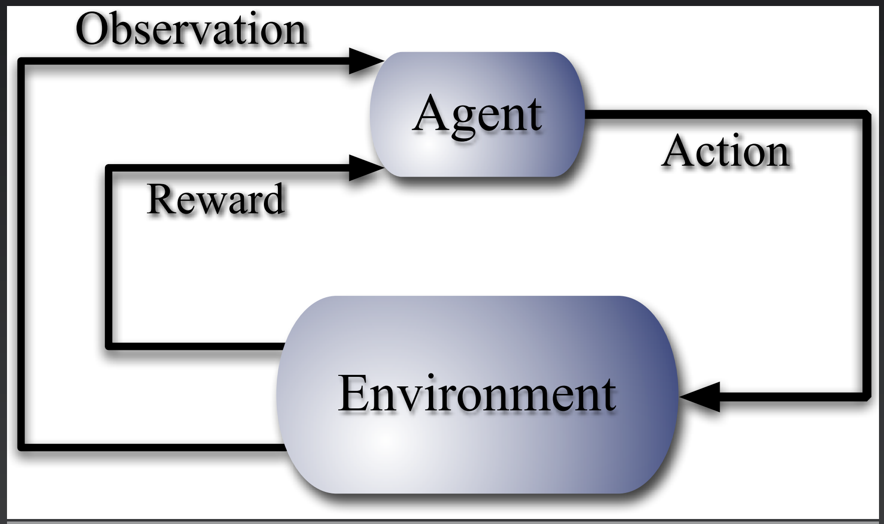
Alborz Geramifard
December, 2006
alborz@cs.ualberta.ca

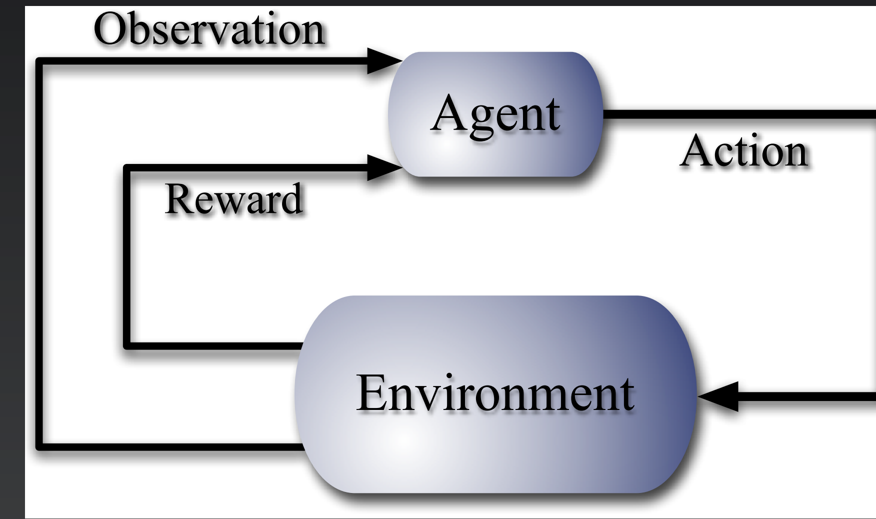
Incremental Least-Squares Temporal Difference Learning

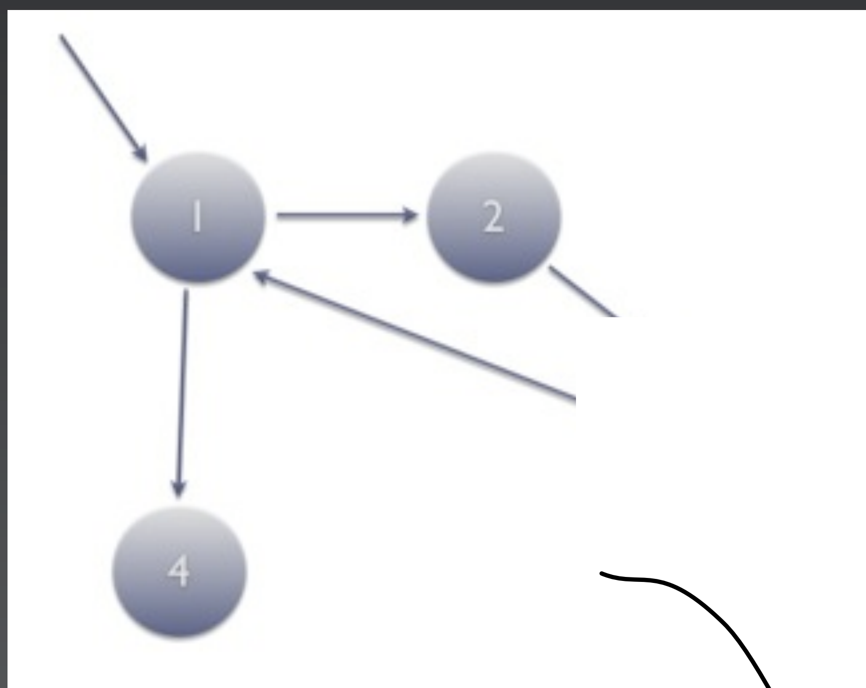
Alborz Geramifard
December, 2006
alborz@cs.ualberta.ca

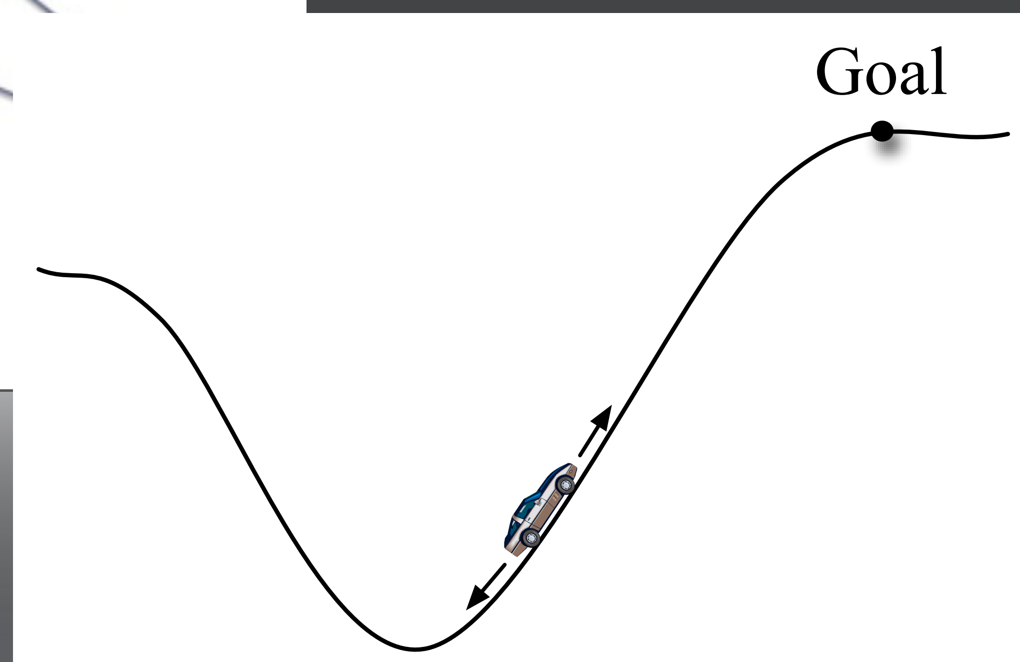

Observation
Agent
Action
Reward
Environment

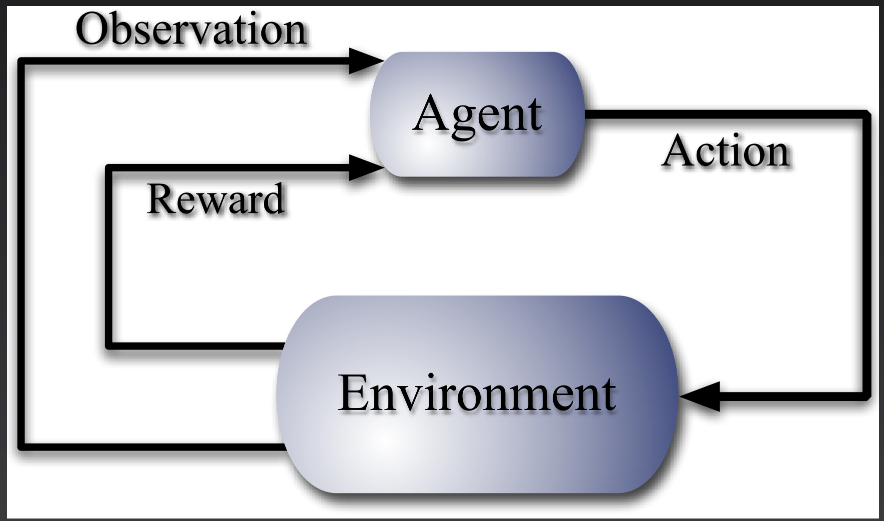

1
2
3
4

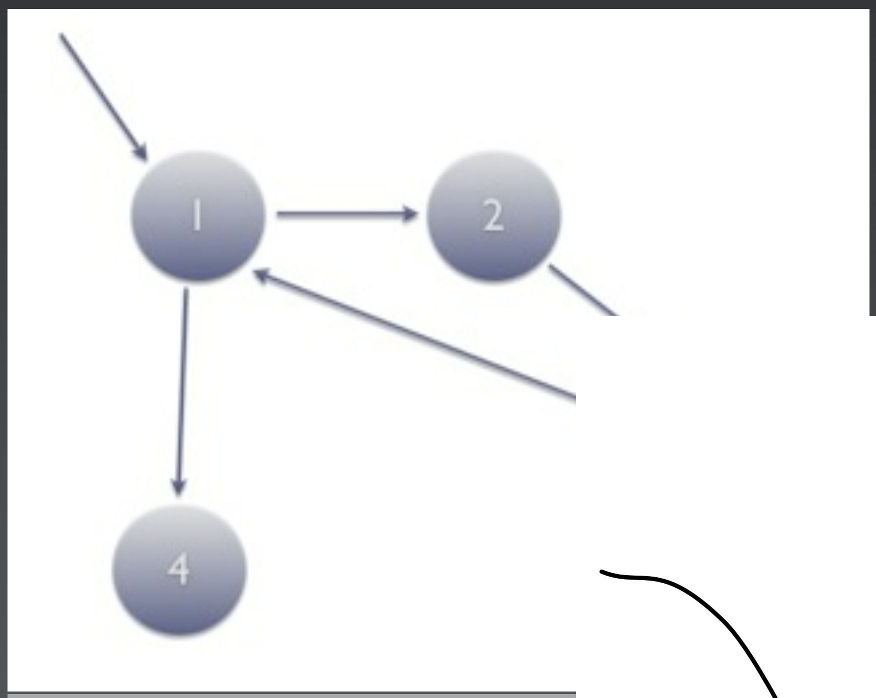

Observation
Agent
Action
Reward
Environment

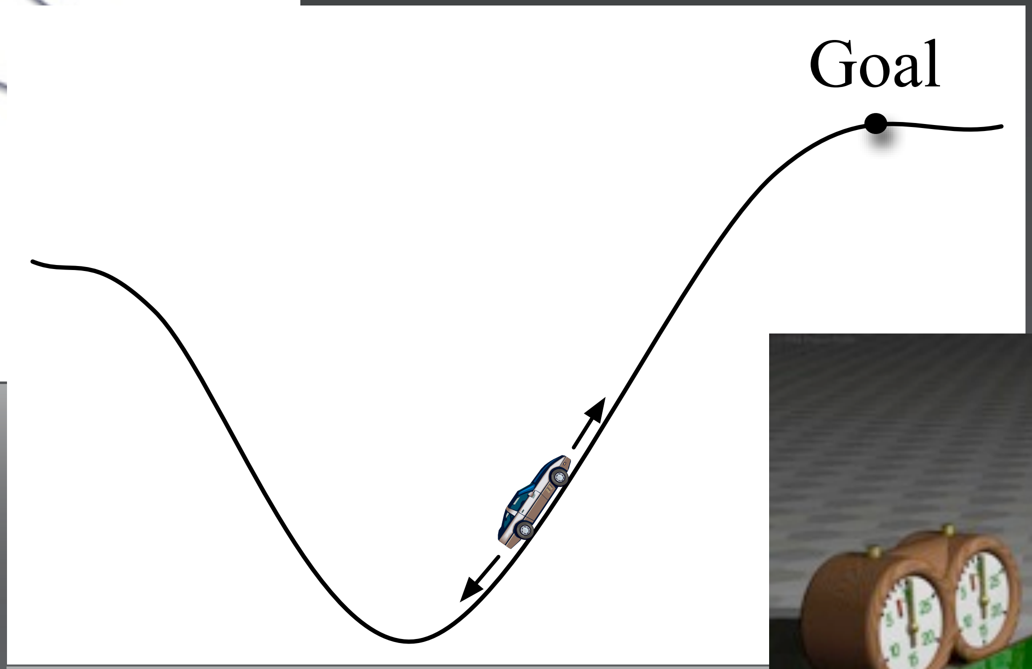

Observation
Agent
Action
Reward
Environment

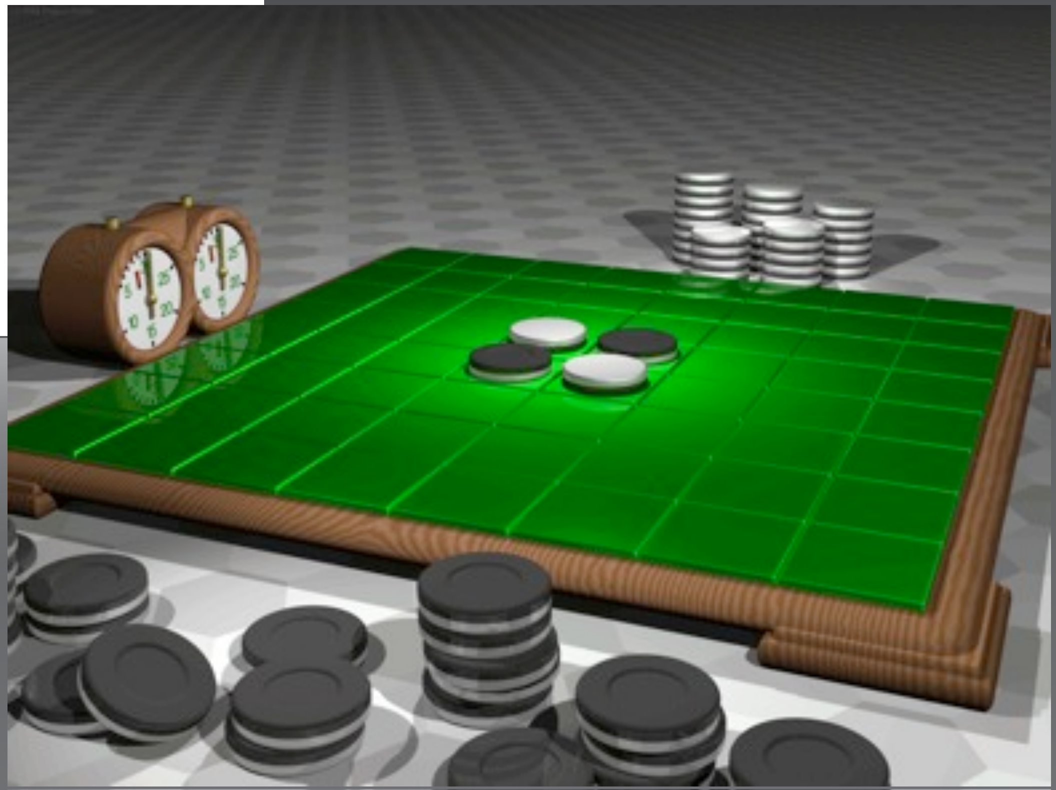

1
2
4

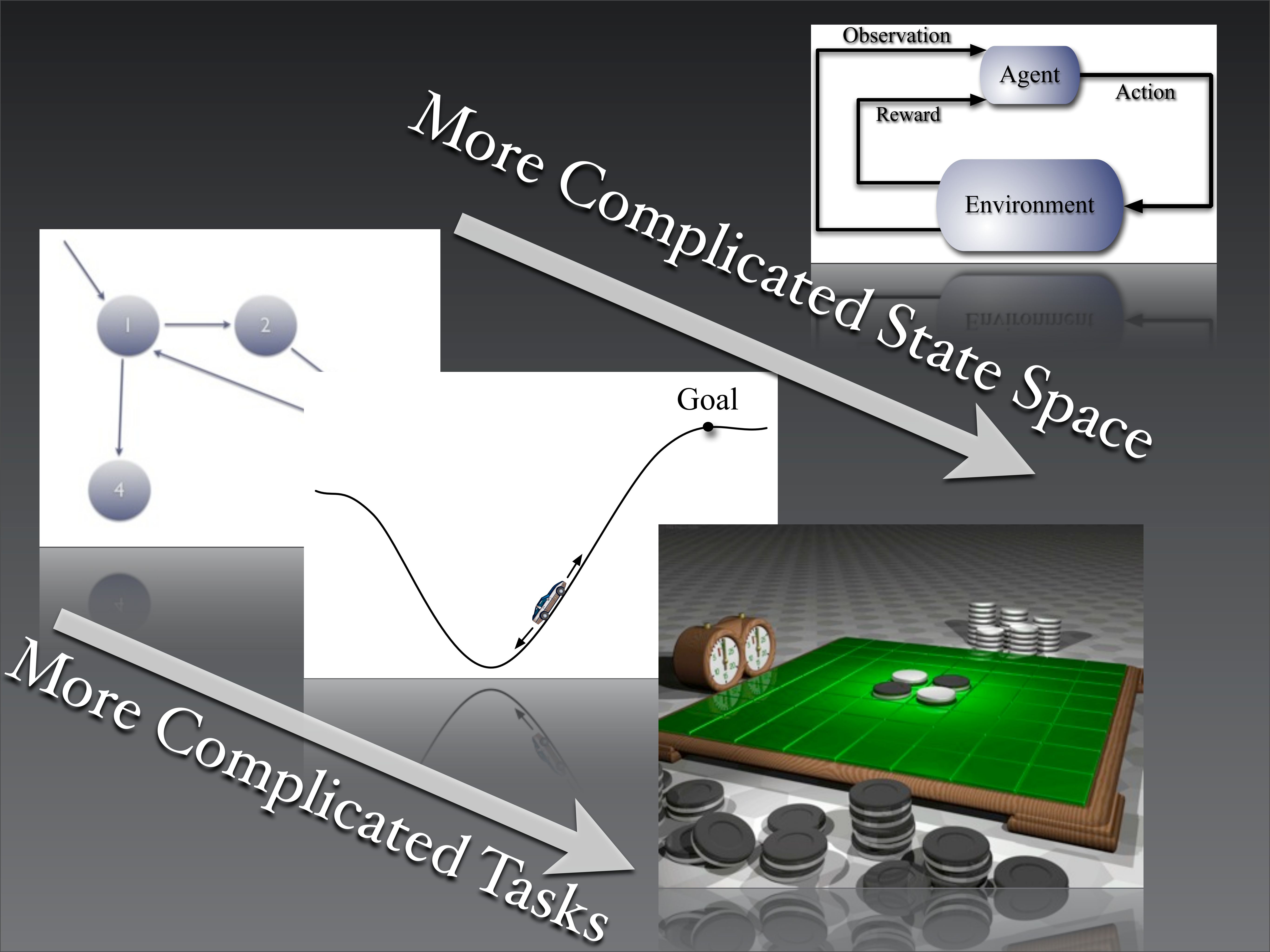

Goal


Observation
Agent
Action
Reward
Environment

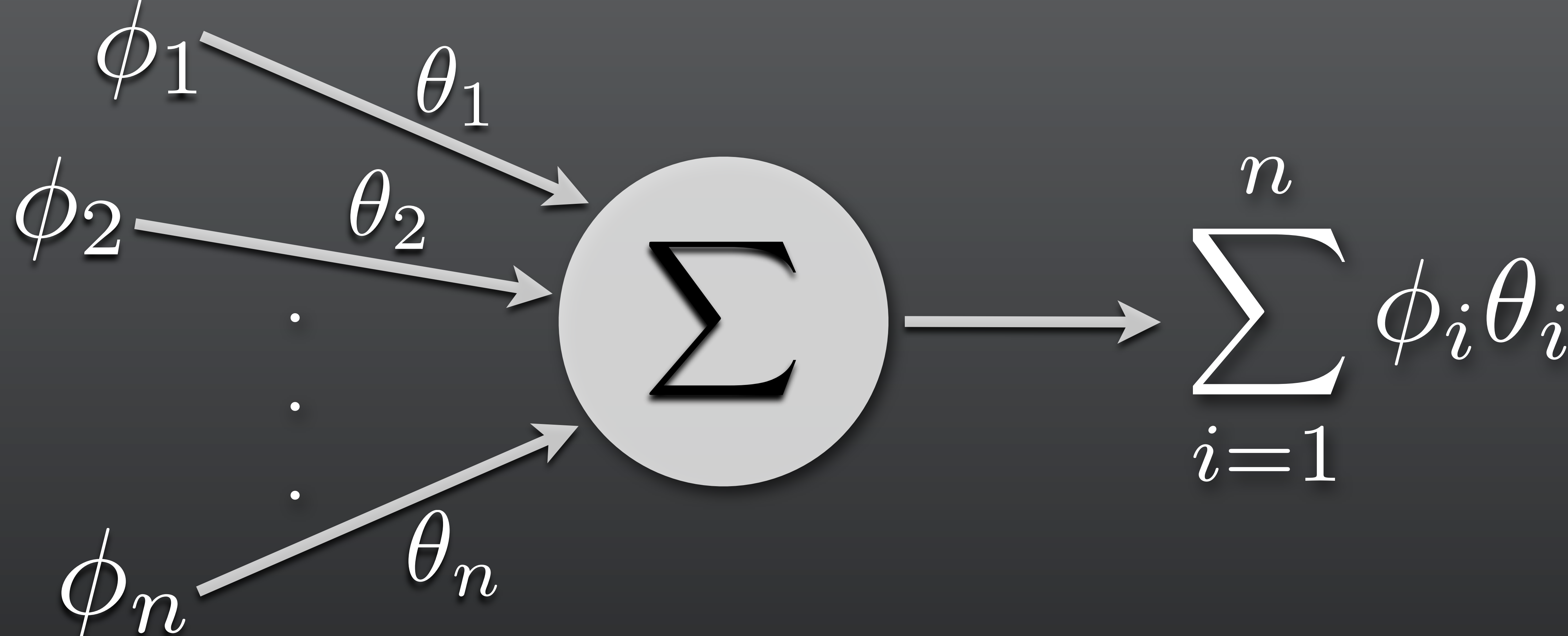

1
2
4


Goal




More Complicated State Space
More Complicated Tasks
Observation
Agent
Action
Reward
Environment
1
2
4
Goal

Linear Function Approximation



Sparsity of features

- Sparsity: Only k features are active at any given moment.

$$k \ll n$$

Sparsity of features

- Sparsity: Only k features are active at any given moment.

$$k \ll n$$

- Simulated leg [Lin, Kim 94]: 350 << 40000

Sparsity of features

- Sparsity: Only k features are active at any given moment.

$$k \ll n$$

  - Simulated leg [Lin, Kim 94]: 350 << 40000
  - Card game [Bowling *et al.* 02]: 3 << 10^6

Sparsity of features

- Sparsity: Only k features are active at any given moment.

$$k \ll n$$

  - Simulated leg [Lin, Kim 94]: 350 << 40000
  - Card game [Bowling *et al.* 02]: 3 << 10^6
  - Keep away soccer [Stone *et al.* 05]: 416 << 10^4

Sparsity of features

- Sparsity: Only k features are active at any given moment.

$$k \ll n$$

  - Simulated leg [Lin, Kim 94]: 350 << 40000
  - Card game [Bowling *et al.* 02]: 3 << 10^6
  - Keep away soccer [Stone *et al.* 05]: 416 << 10^4
  - Go [Silver *et al.* 07]: ~200 << ~10^6

Policy Evaluation

Policy Evaluation

Policy Improvement

Policy Evaluation

Policy Improvement

Policy Evaluation

Policy Improvement

Clarification

Clarification

- Feature, Dimension, Component

Clarification

- Feature, Dimension, Component
- Time, Running time, Speed, Computation, Computer time

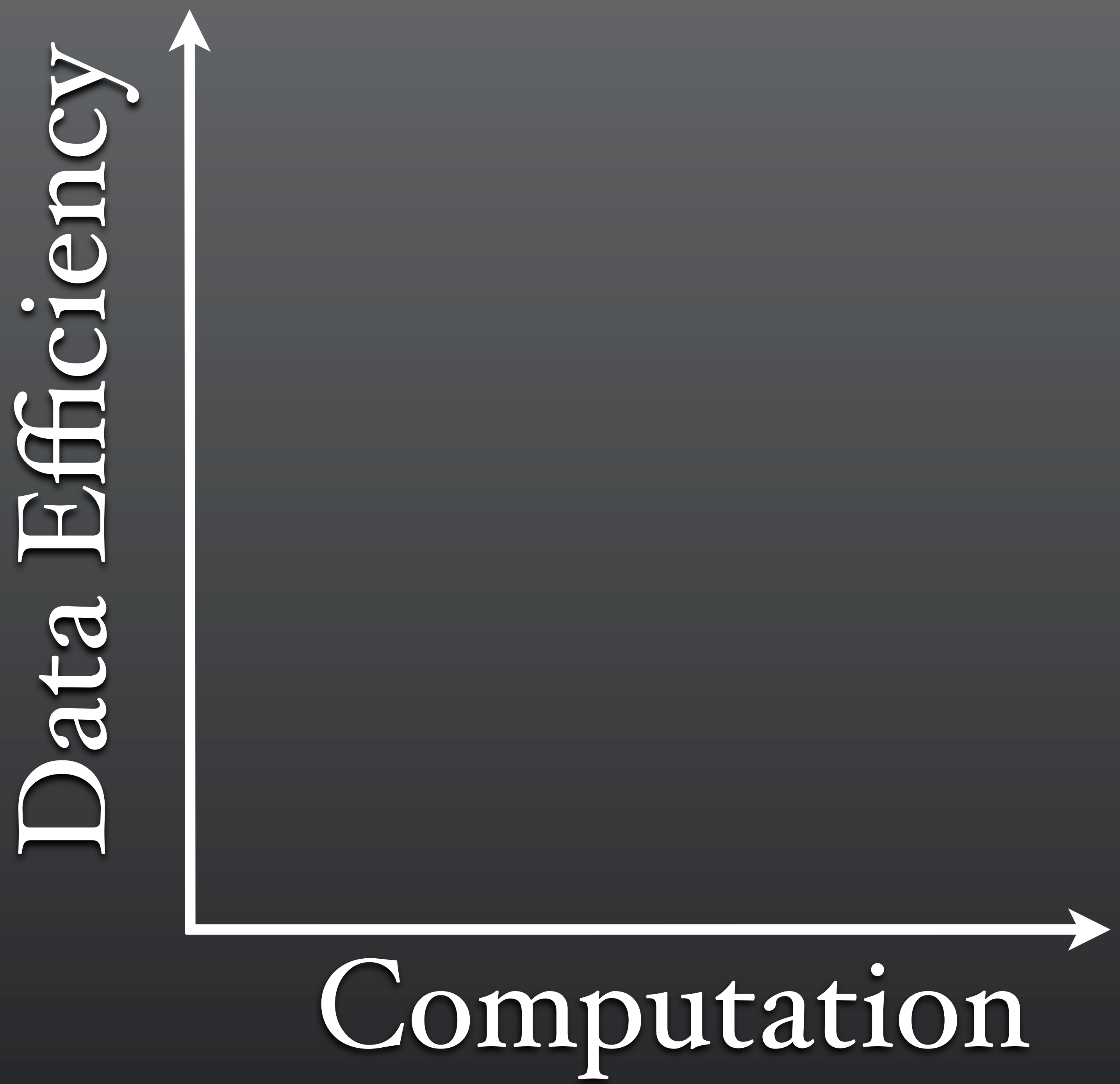
Clarification

- Feature, Dimension, Component
- Time, Running time, Speed, Computation, Computer time
- Data, Data Efficiency

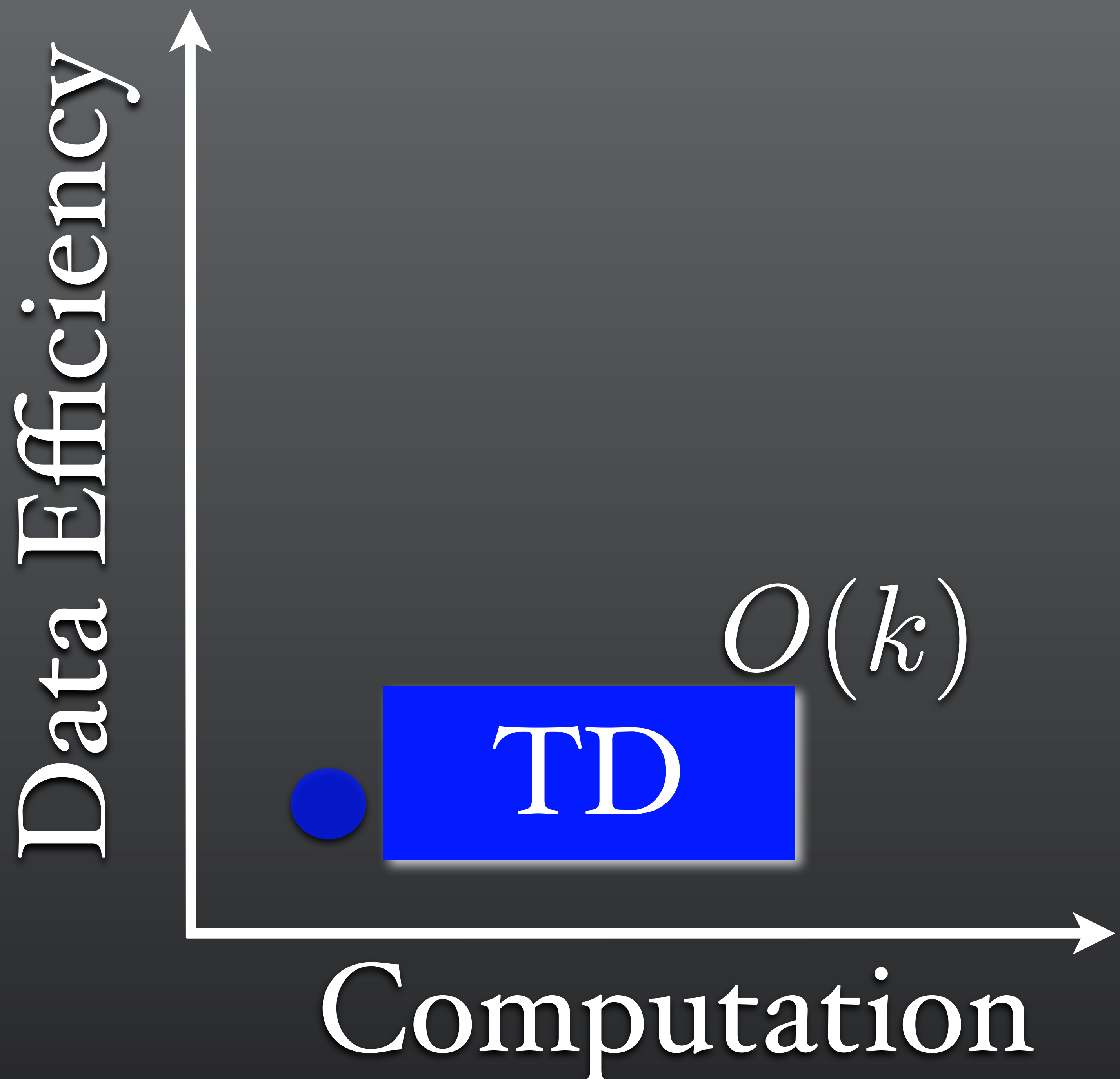
Clarification

- Feature, Dimension, Component
- Time, Running time, Speed, Computation, Computer time
- Data, Data Efficiency

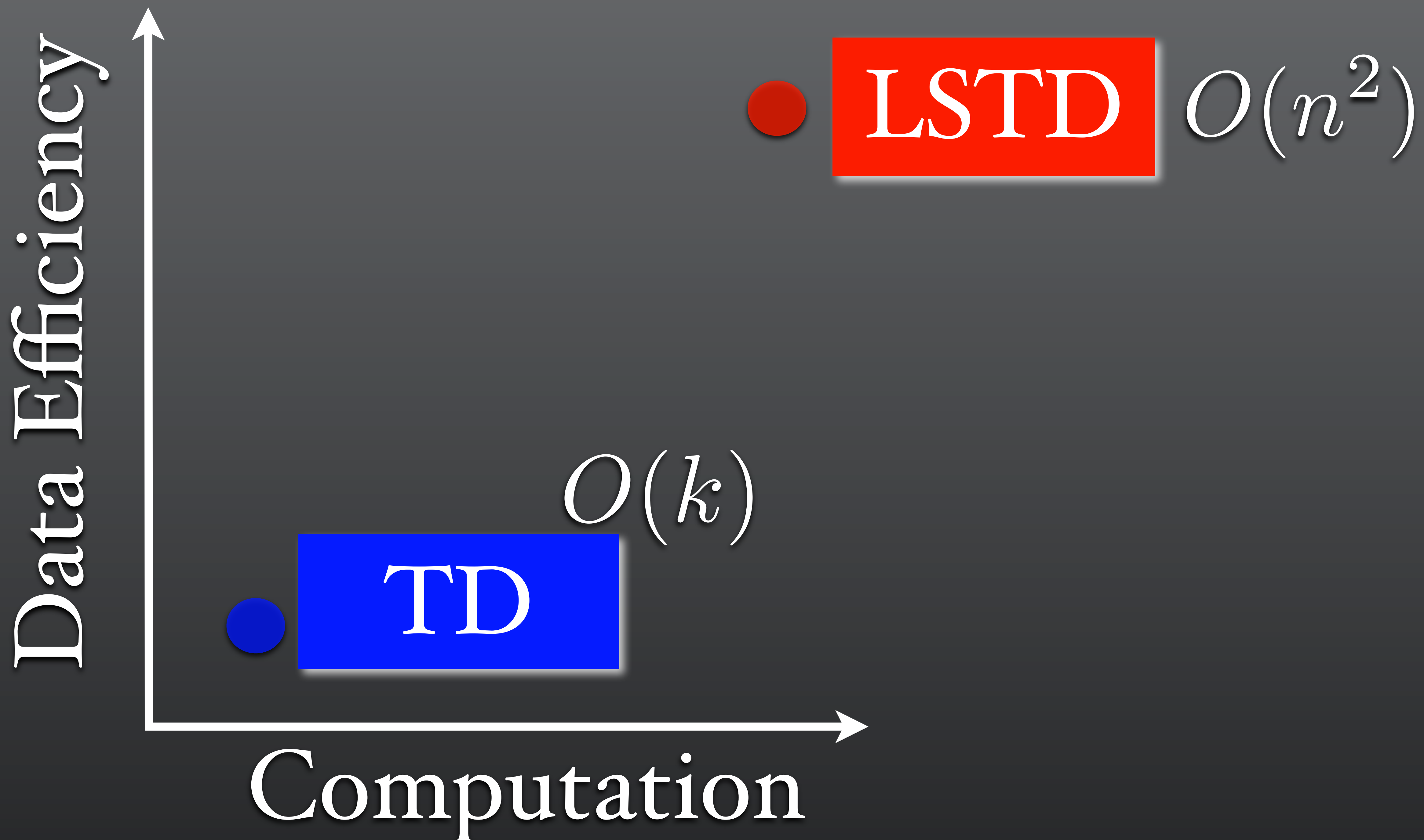
Summary



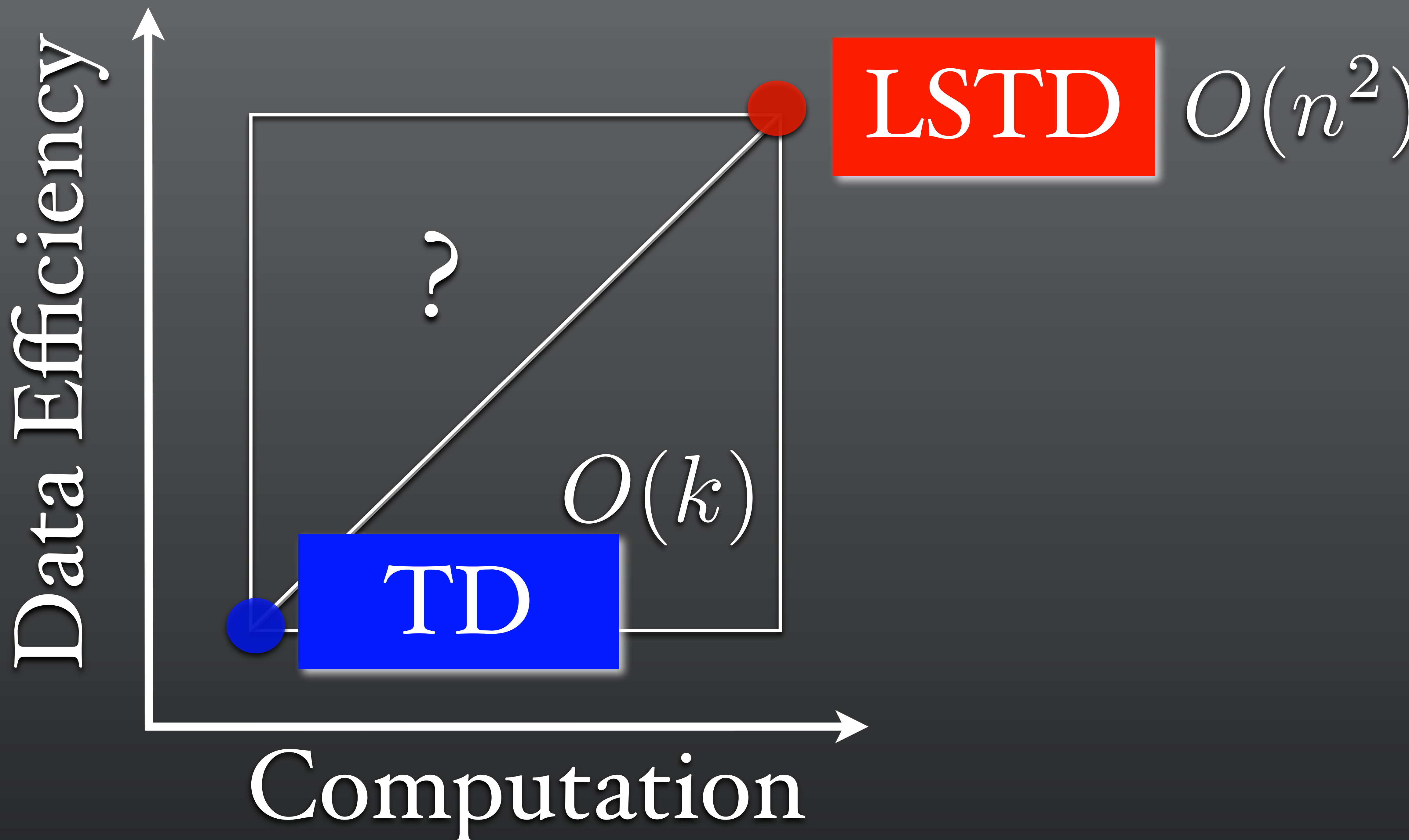
Summary



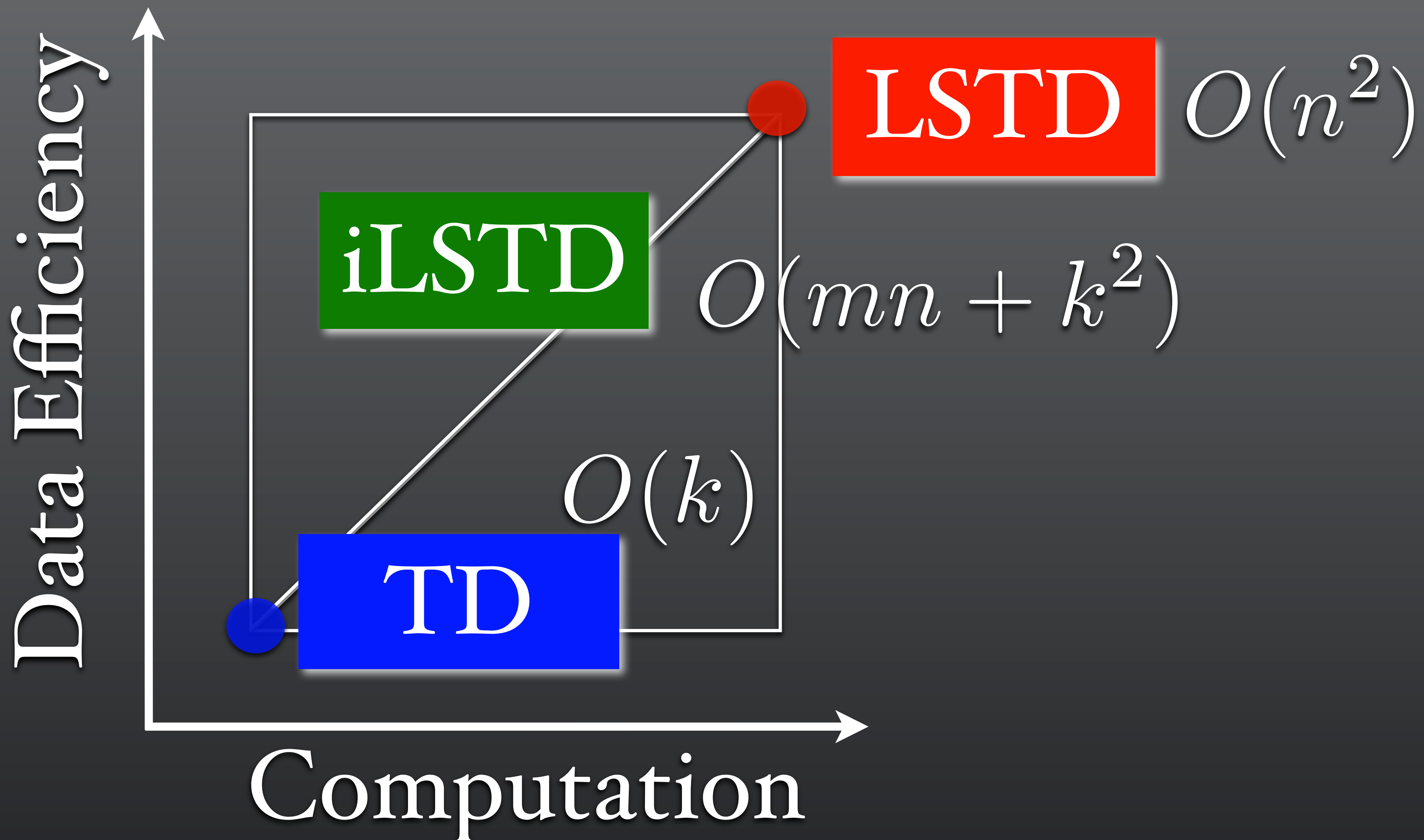
Summary



Summary



Summary



Contributions

Contributions

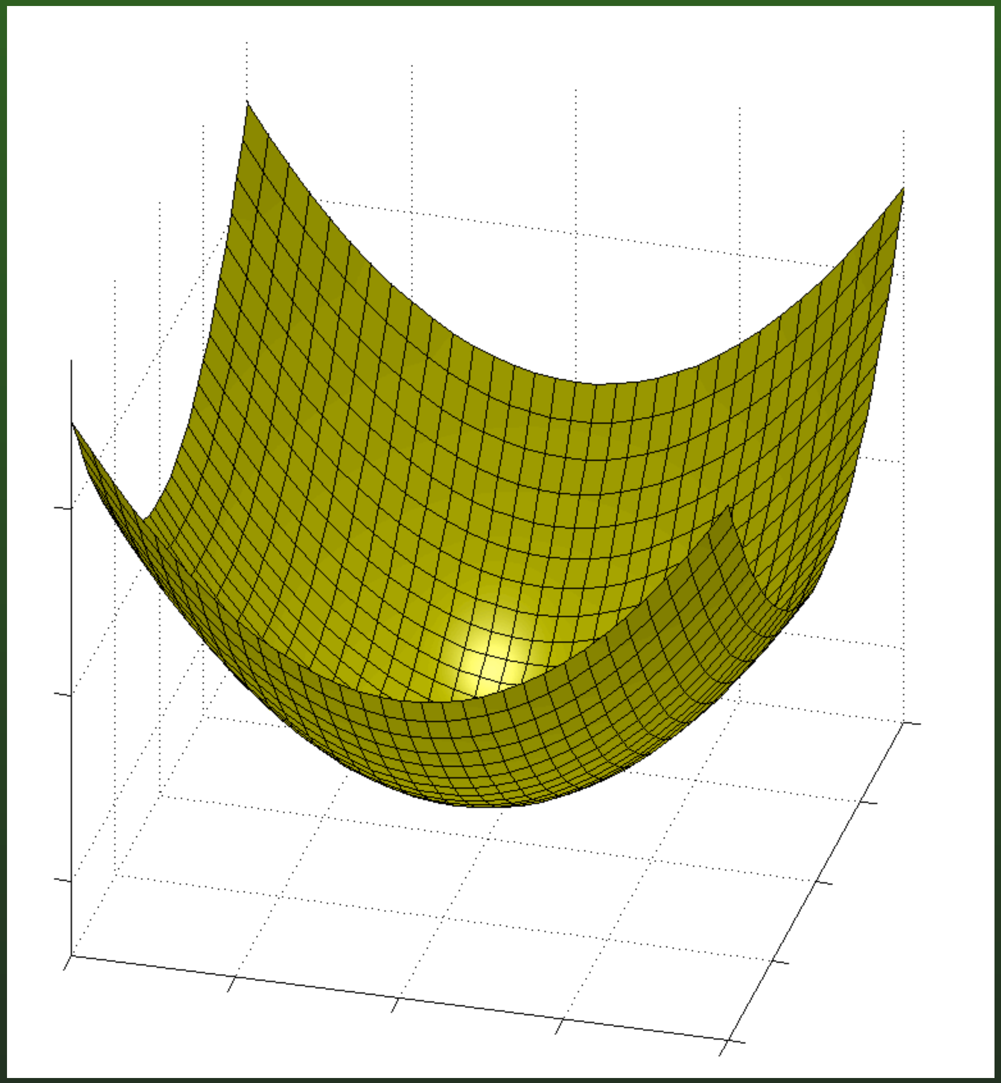
- iLSTD: A new policy evaluation algorithm
- Computation analysis
- Extension with eligibility traces
- Proof of convergence
- Component selection methods
- Empirical results

Contributions

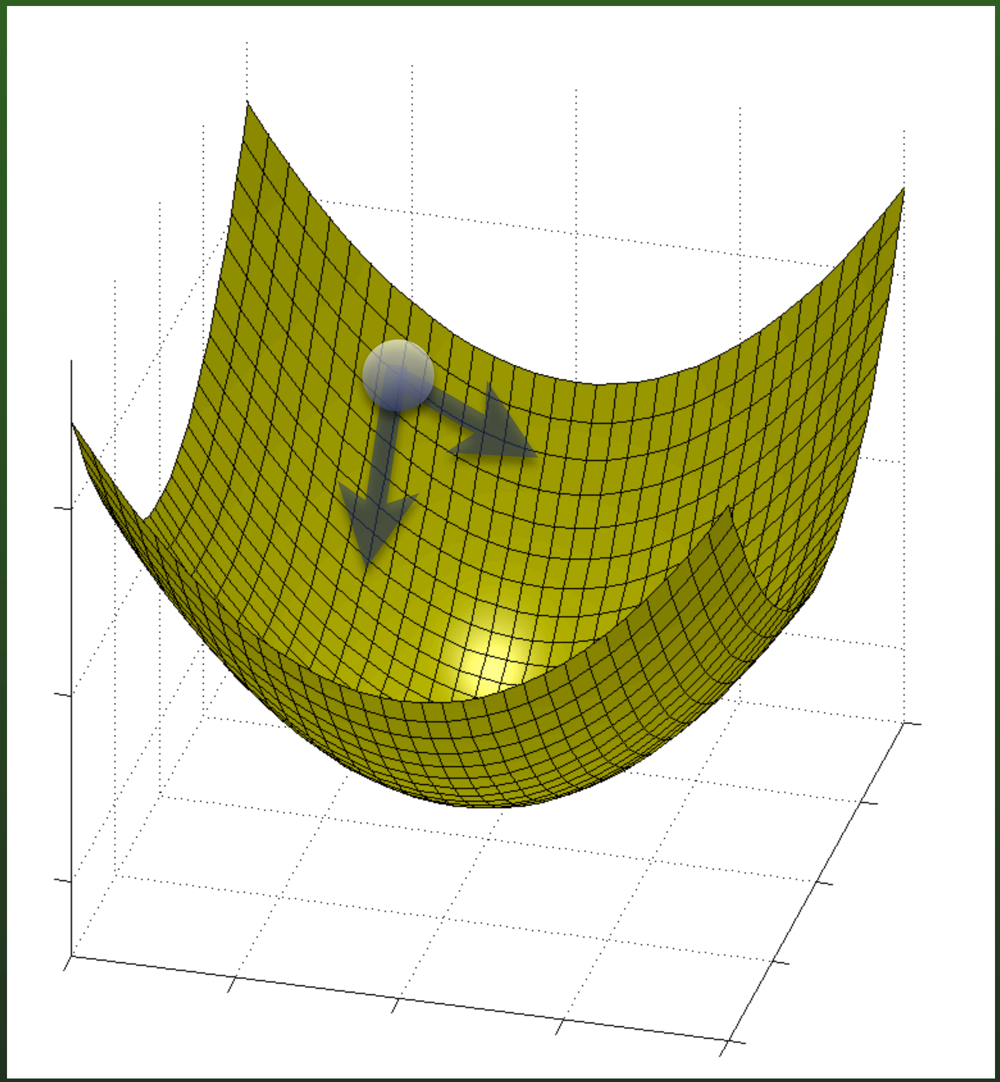
- iLSTD: A new policy evaluation algorithm
- Computation analysis
- Extension with eligibility traces
- Proof of convergence
- Component selection methods
- Empirical results

Main Idea of iLSTD

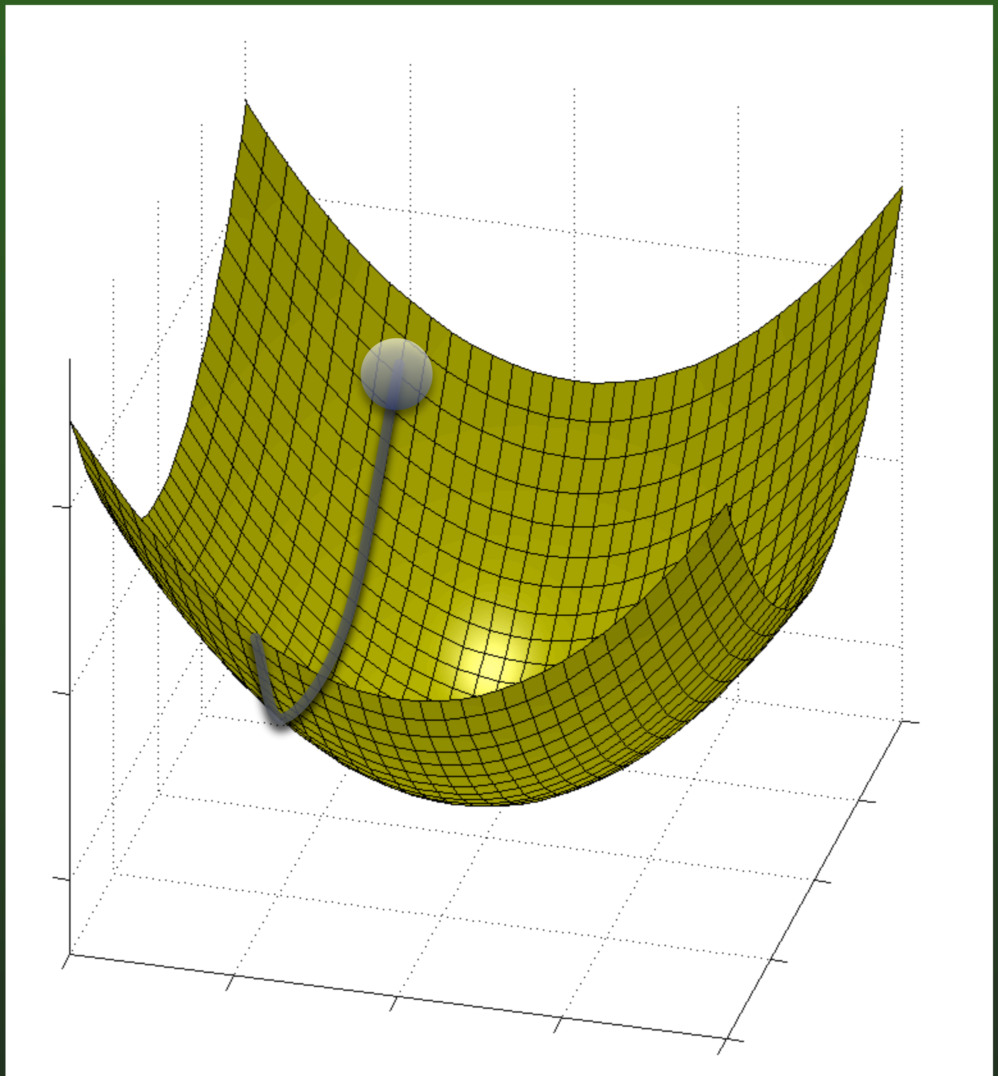
Main Idea of iLSTD



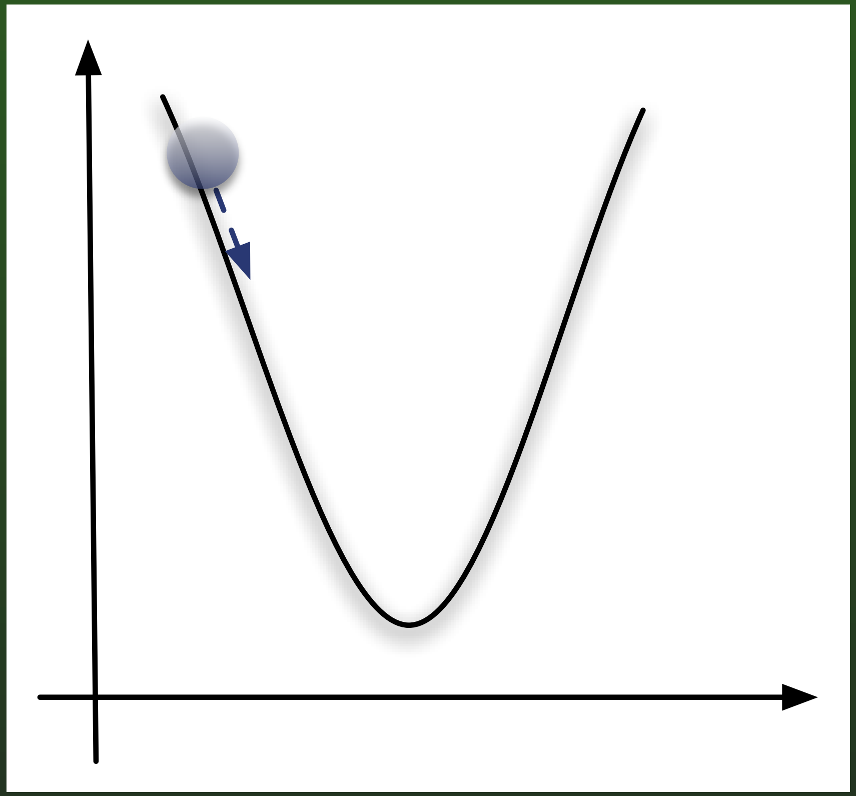
Main Idea of iLSTD



Main Idea of iLSTD



Main Idea of iLSTD



Eligibility Traces

Eligibility Traces

TD

$$O(k) \rightarrow O(lk)$$

Eligibility Traces

TD

$$O(k) \rightarrow O(lk)$$

$$l = \log_{\lambda}^{\xi}$$

Eligibility Traces

TD $O(k) \rightarrow O(lk)$

iLSTD $O(mn + k^2) \rightarrow O(mn + lk^2)$

$$l = \log_{\lambda}^{\xi}$$

Eligibility Traces

TD $O(k) \rightarrow O(lk)$

iLSTD $O(mn + k^2) \rightarrow O(mn + lk^2)$

LSTD $O(n^2)$

$$l = \log_{\lambda}^{\xi}$$

Proof of Convergence

- *Theorem* : iLSTD converges with probability one to the same solution as TD, under the usual step-size conditions, for any component selection method such that all components for which μ_t is non-zero are selected in the limit an infinite number of times.

Contributions

Contributions

- iLSTD: A new policy evaluation algorithm
- Component analysis
- Extension with eligibility traces
- Proof of convergence
- Component selection methods
- Empirical results

Contributions

- iLSTD: A new policy evaluation algorithm
- Component analysis
- Extension with eligibility traces
- Proof of convergence
- Component selection methods
- Empirical results

Greedy Component Selection

- Pick the component for which the descent is steepest

Greedy Component Selection

- Pick the component for which the descent is steepest
  - Different from steepest descent method!

Greedy Component Selection

- Pick the component for which the descent is steepest
  - Different from steepest descent method!
- Not proven to converge.

ε-Greedy Component Selection

- ε : Non-Zero Random
- (1-ε) : Greedy

ε-Greedy Component Selection

- ε : Non-Zero Random
- (1-ε) : Greedy
- Convergence proof applies.

Boltzmann Component Selection

- Boltzmann Distribution + Non-Zero Random
- Convergence proof applies.

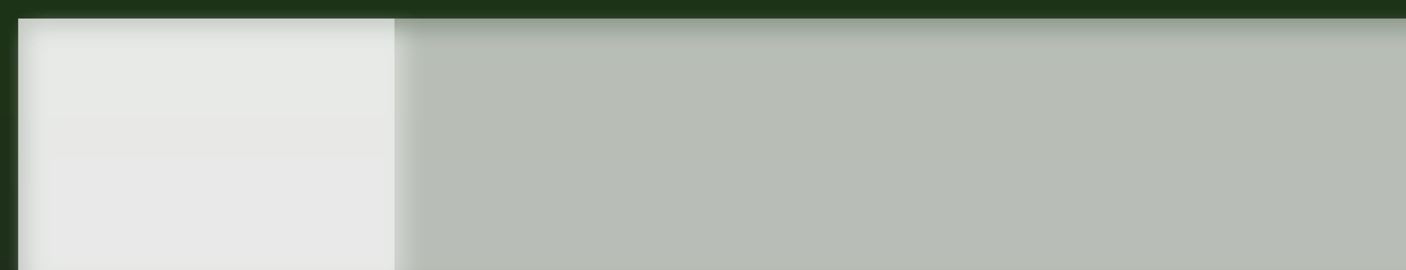
Boltzmann Component Selection

- Boltzmann Distribution + Non-Zero Random

- Convergence proof applies.

Boltzmann Component Selection

- Boltzmann Distribution + Non-Zero Random



$\psi \times m$

- Convergence proof applies.

Boltzmann Component Selection

- Boltzmann Distribution + Non-Zero Random

$\psi \times m$

- Convergence proof applies.

Boltzmann Component Selection

- Boltzmann Distribution + Non-Zero Random

Boltzmann Distribution

$\psi \times m$

- Convergence proof applies.

Conclusions Based on Empirical Results

Conclusions Based on Empirical Results

- iLSTD outperforms TD both data and computation wise.

Conclusions Based on Empirical Results

- iLSTD outperforms TD both data and computation wise.
  - Computations are based on implementation.

Conclusions Based on Empirical Results

- iLSTD outperforms TD both data and computation wise.
  - Computations are based on implementation.
    - Better idea: counting the number of floating point operations

Conclusions Based on Empirical Results

- iLSTD outperforms TD both data and computation wise.
  - Computations are based on implementation.
    - Better idea: counting the number of floating point operations
- If the number of features increases enough, iLSTD outperforms LSTD based on computation.

Conclusions Based on Empirical Results

- If number of features increases enough, LSTD can not outperform TD based on computation.

Conclusions Based on Empirical Results

- If number of features increases enough, LSTD can not outperform TD based on computation.

- Data wise LSTD outperforms TD.

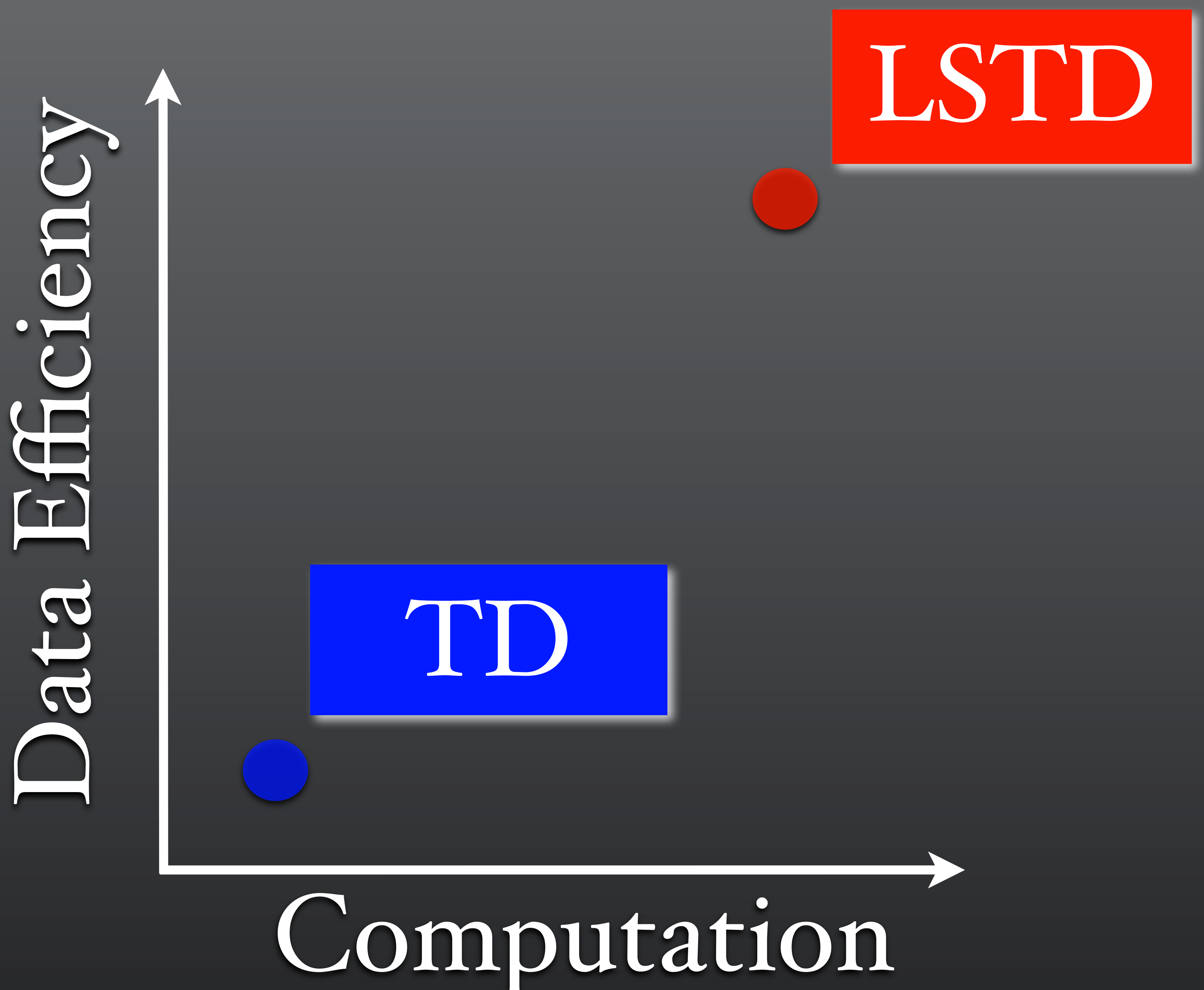
Conclusions Based on Empirical Results

- If number of features increases enough, LSTD can not outperform TD based on computation.
- Data wise LSTD outperforms TD.
- Greedy component selection method outperformed Random, ε-Greedy, and Boltzmann in most cases.

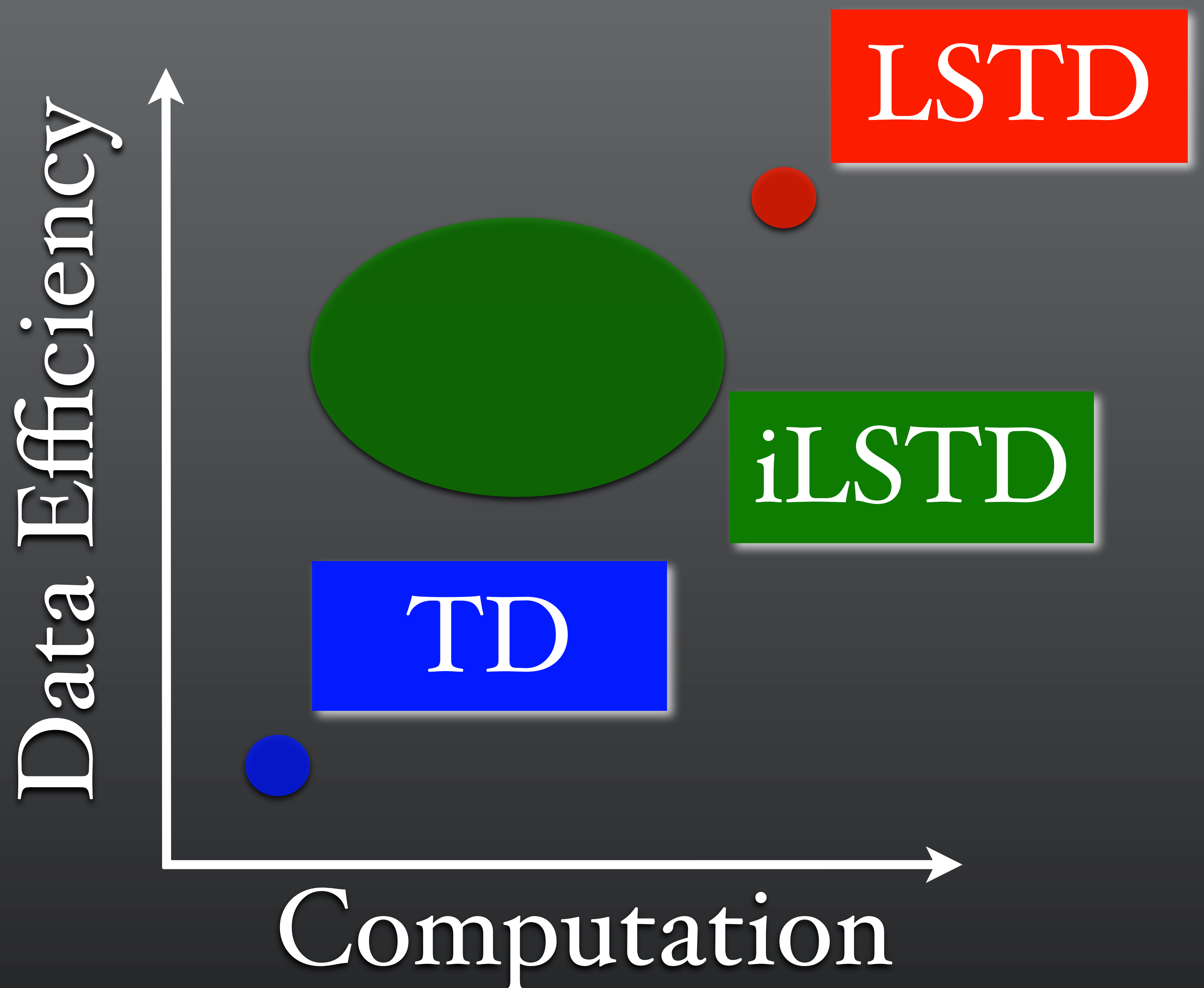
Conclusions Based on Empirical Results

- If number of features increases enough, LSTD can not outperform TD based on computation.
- Data wise LSTD outperforms TD.
- Greedy component selection method outperformed Random, ε-Greedy, and Boltzmann in most cases.
- No convergence proof!

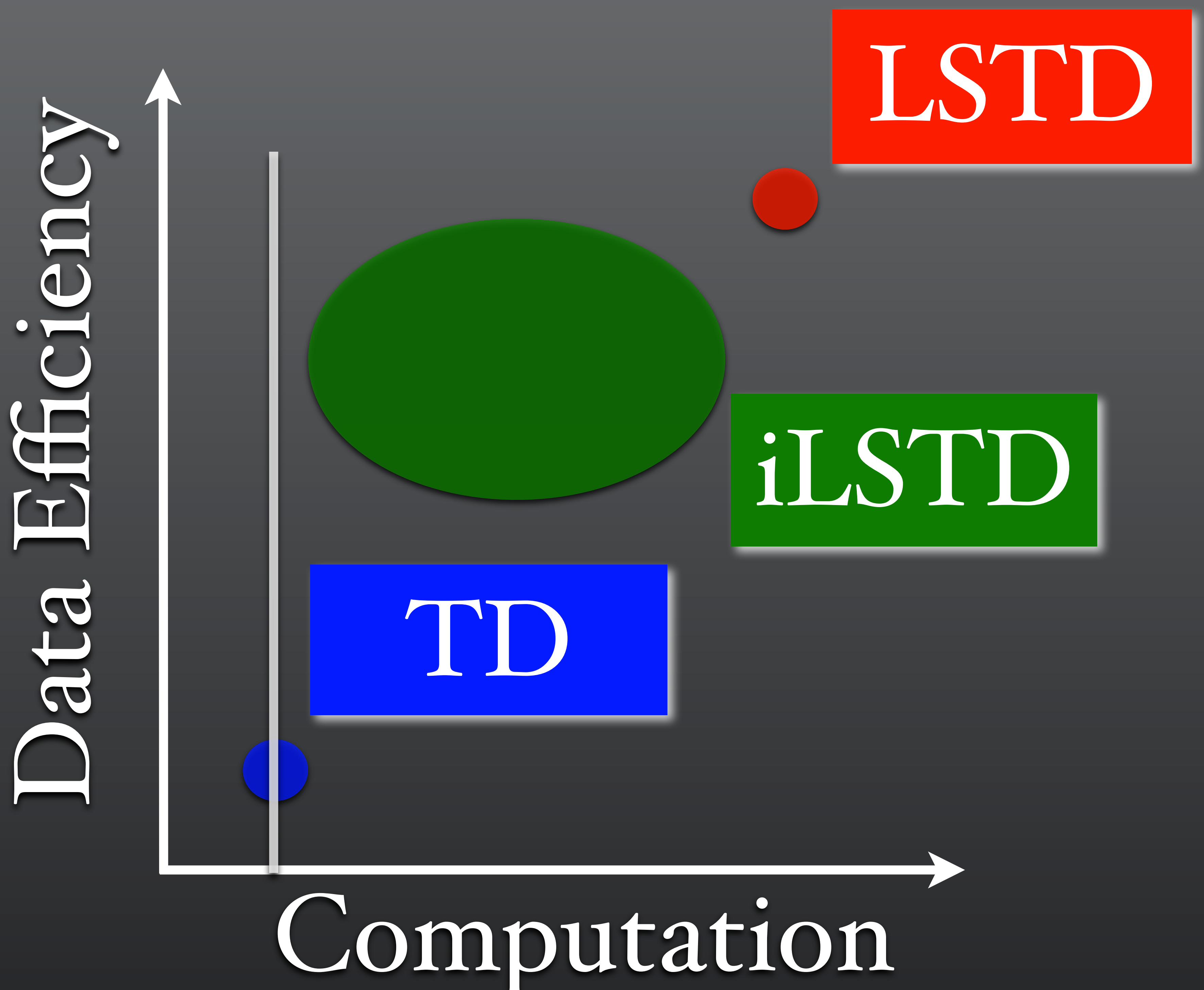
Conclusion



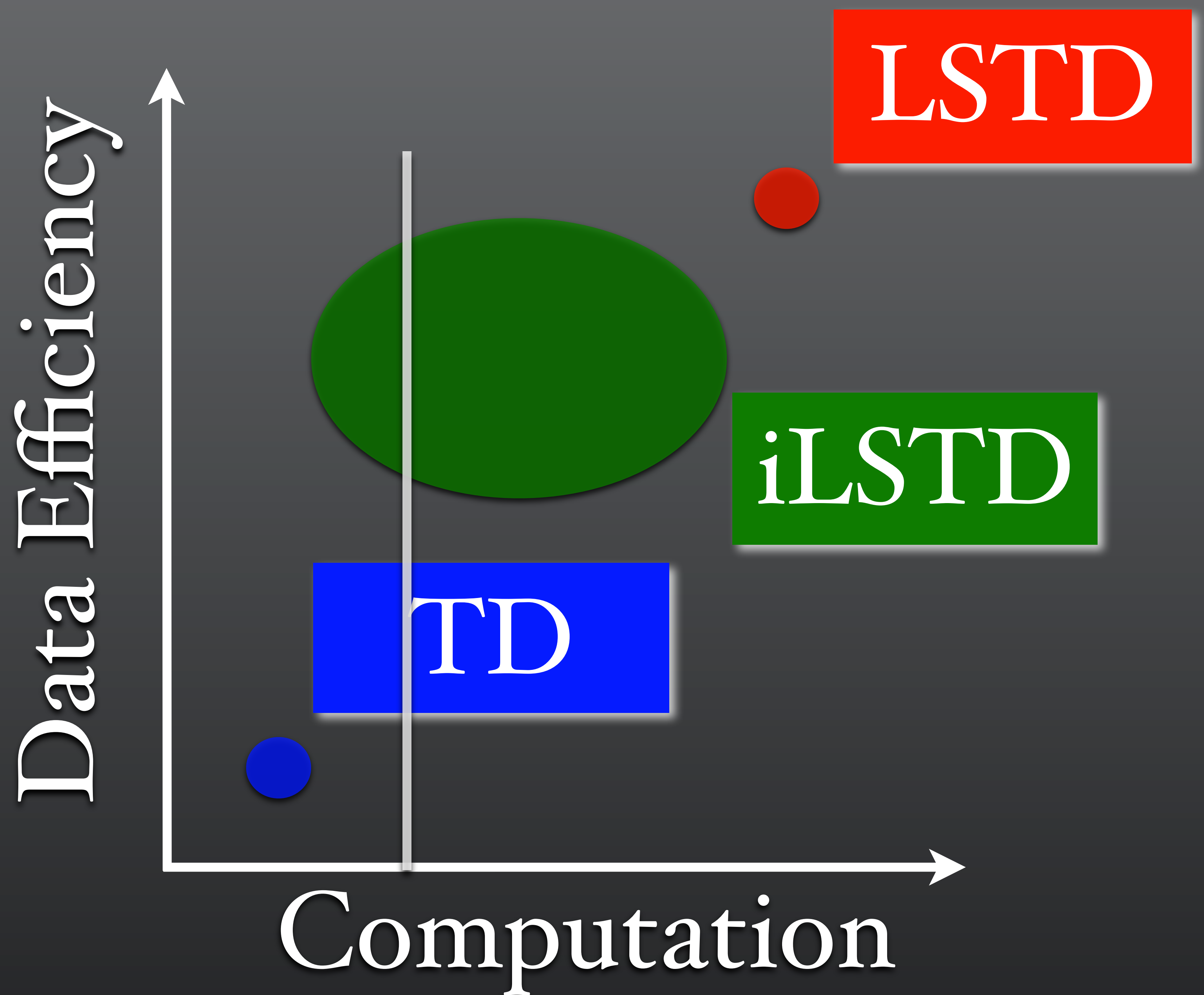
Conclusion



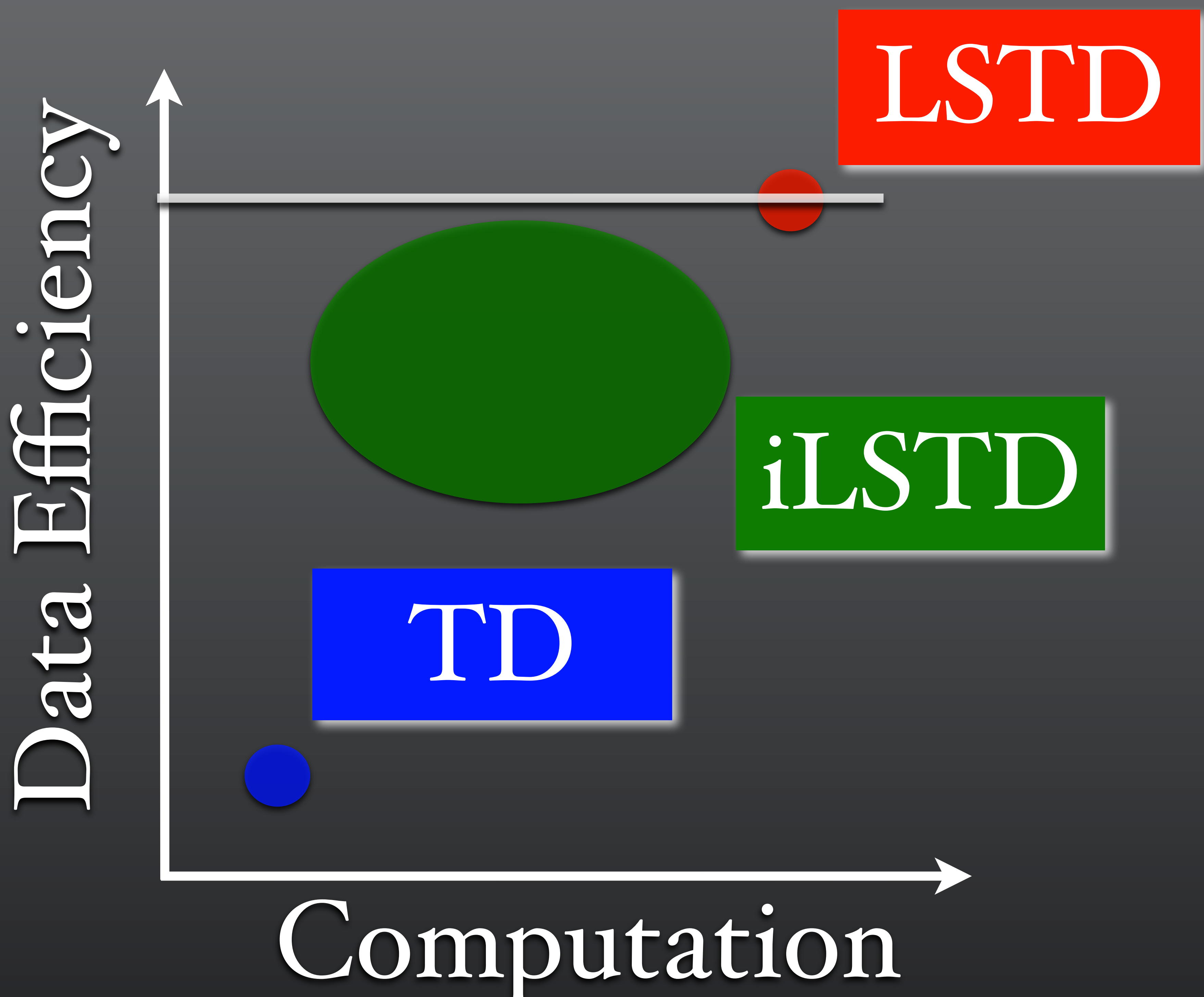
Conclusion



Conclusion

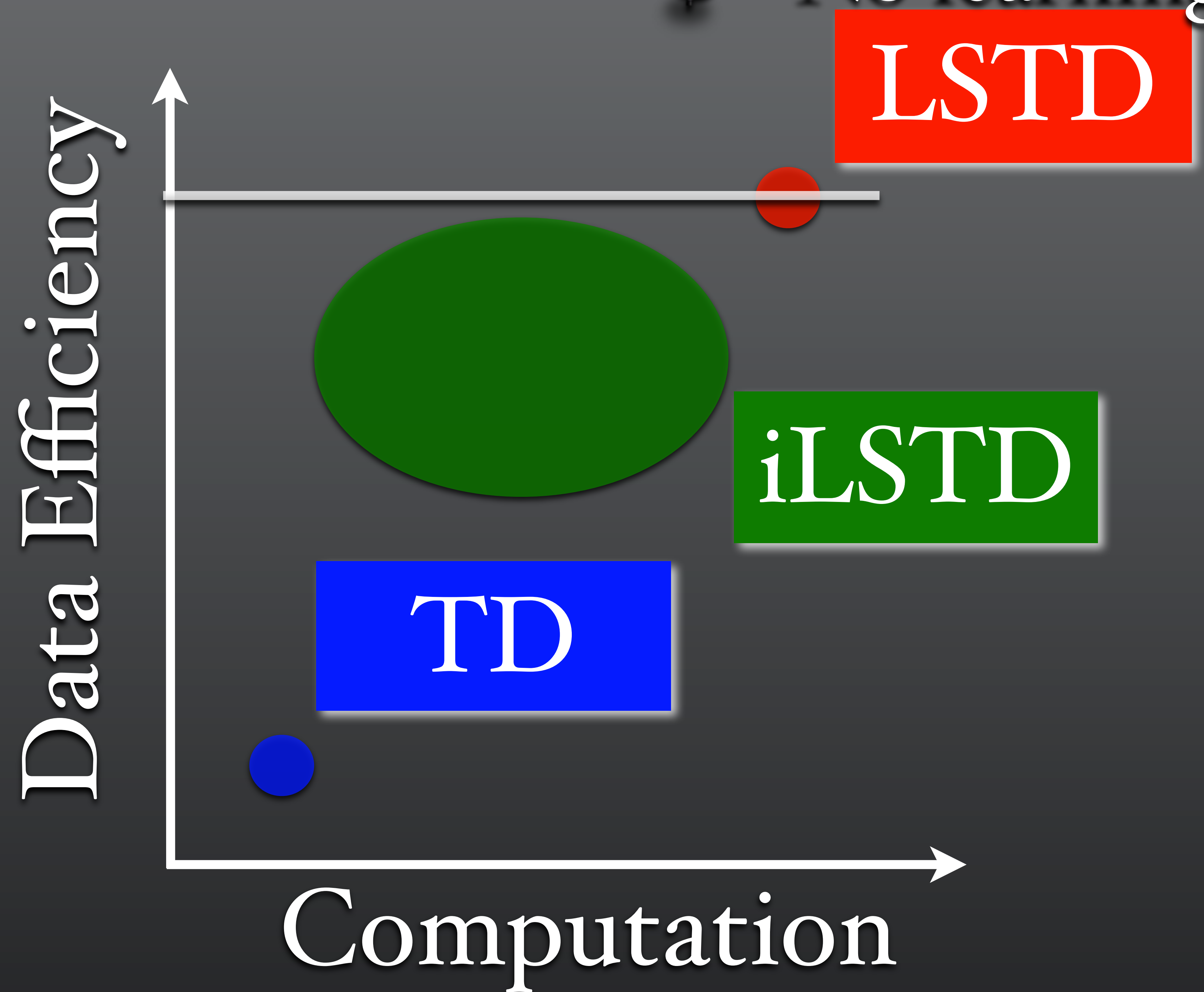


Conclusion



Conclusion

- No learning rate!



Contributions

Contributions

- iLSTD: A new policy evaluation algorithm
- Running time analysis
- Extension with eligibility traces
- Proof of convergence
- Dimension selection methods
- Empirical results

Contributions

- iLSTD: A new policy evaluation algorithm
- Running time analysis
- Extension with eligibility traces
- Proof of convergence
- Dimension selection methods
- Empirical results