



# *A Cost-based GD Optimizer*

Zoi Kaoudi

Jorge Quiané\*

Sara Thirumuruganathan

Sanjay Chawla

Divy Agrawal



**QCRI**

معهد قطر لبحوث الحوسبة  
Qatar Computing Research Institute

جامعة حمد بن خليفة  
HAMAD BIN KHALIFA UNIVERSITY



# *-based GD Optimizer*

Zoi Kaoudi

Jorge Quiané\*

Sara Thirumuruganathan

Sanjay Chawla

Divy Agrawal



معهد قطر لبحوث الحوسبة  
Qatar Computing Research Institute



# Data Scientist?

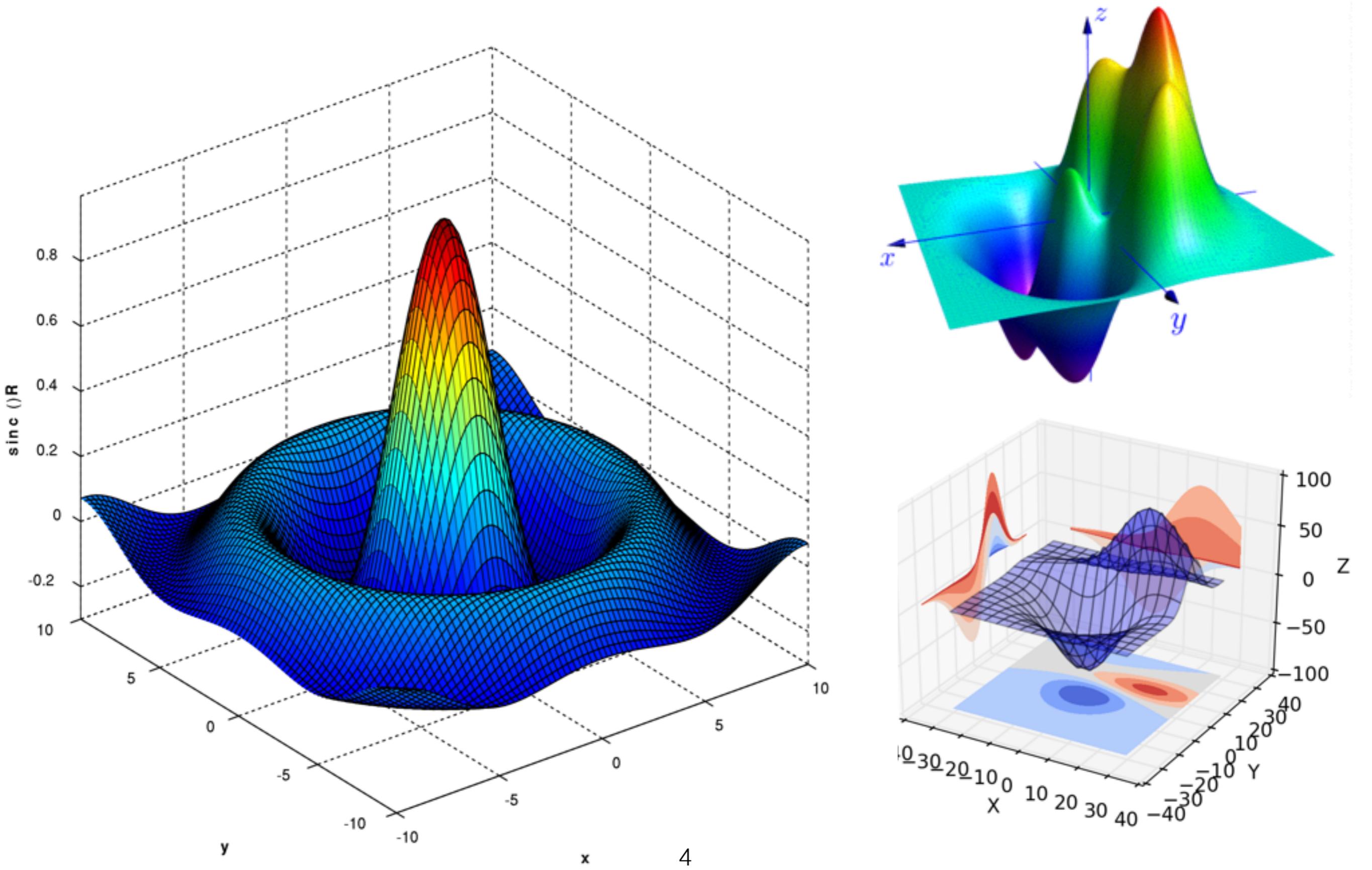
# Data Scientist?

The sexiest job  
of the 21st century!

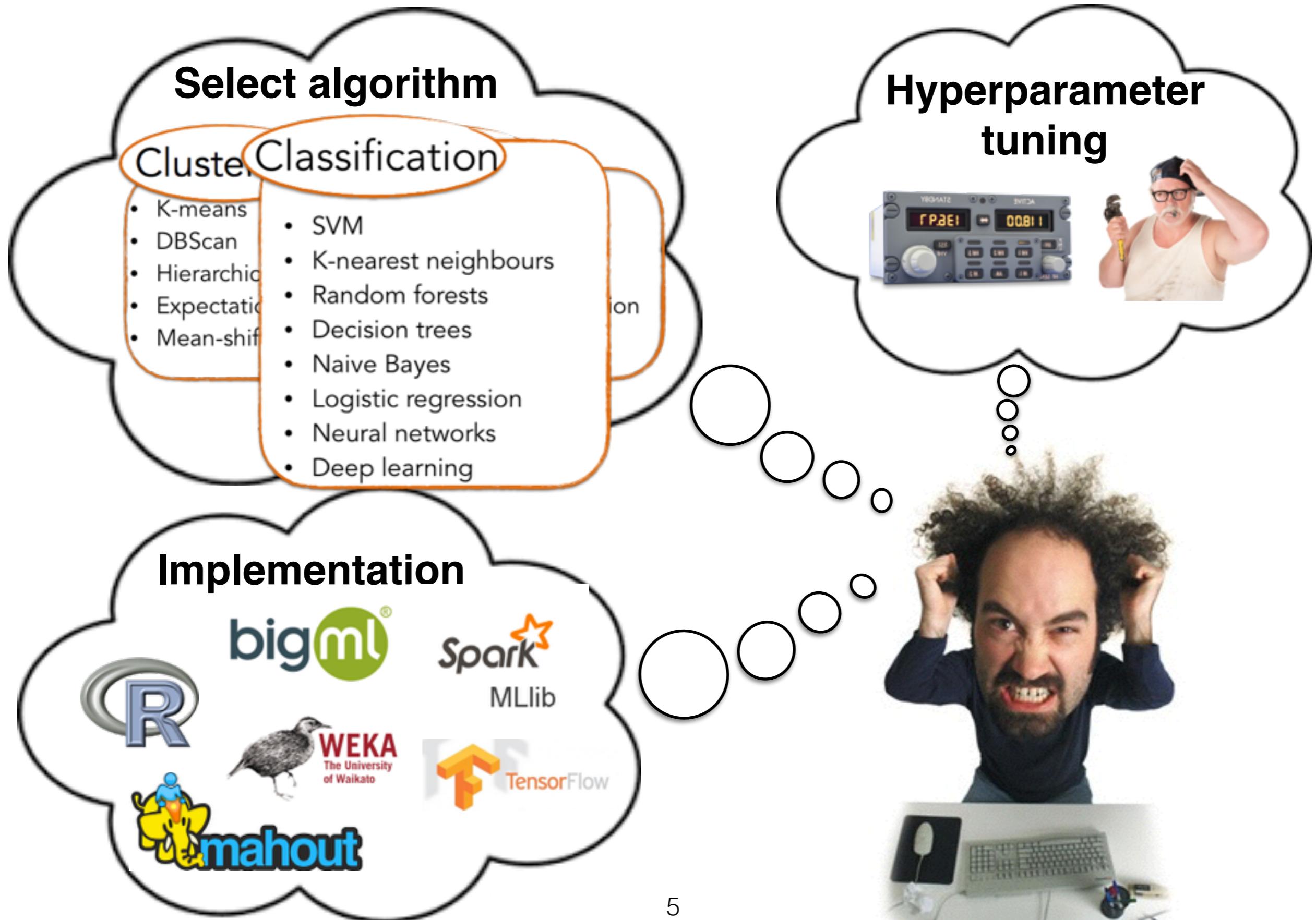
# What people think he does



# What he thinks he does



# This is what he really does!



Oh! You're dropping your numbers!

No worries hon! I've so many of them.  
I'm a data scientist

Data Scientist:  
The Sexiest Job of the 21st Century

Oh! You're dropping your numbers!

No worries hon! I've so many of them.  
I'm a data scientist



Data S~~cientist~~  
The Sexiest Job of the 21st Century

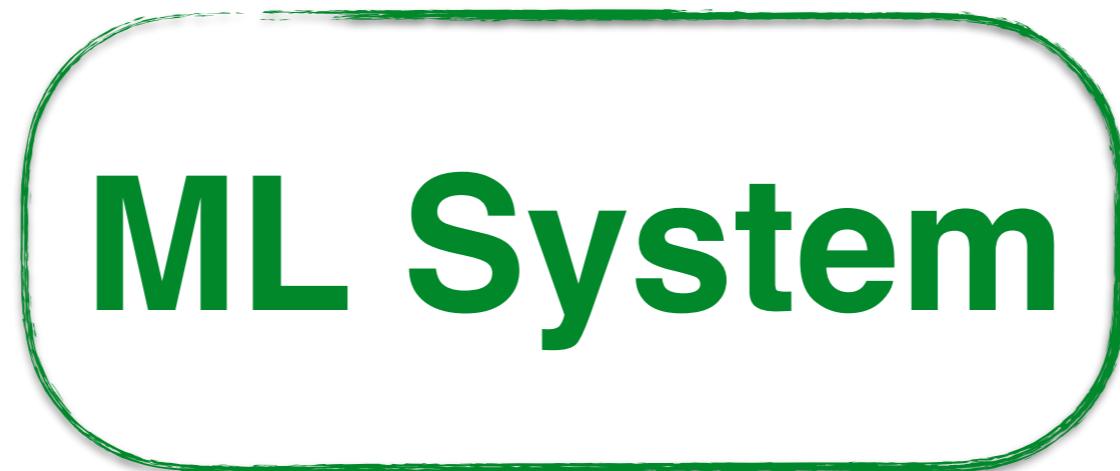
Can we have a  
**DB-like ML system?**

DB-like

# Data Scientist Dream



ML task



Model

DB-like

# Data Scientist Dream



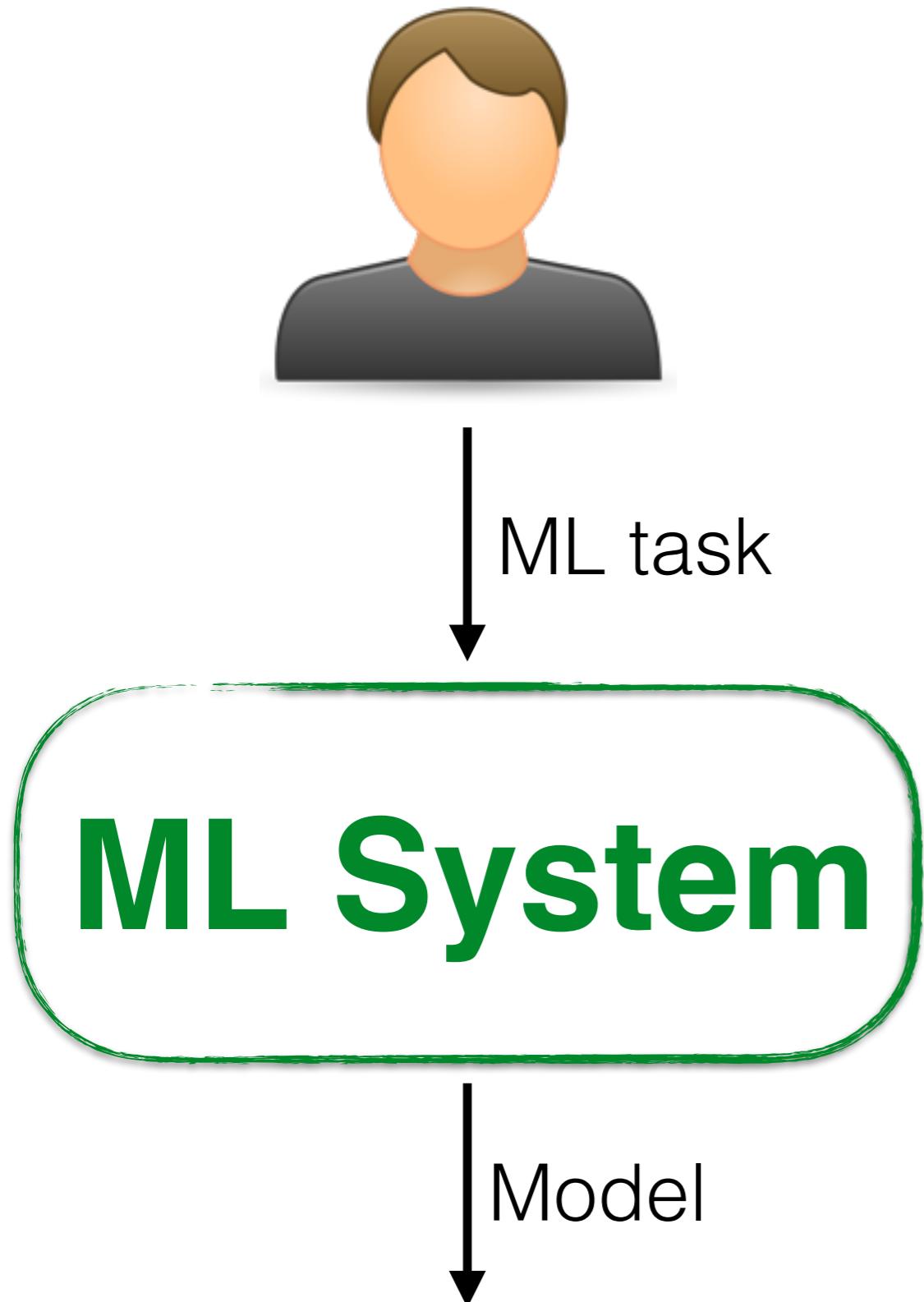
**Standard DB  
Statistics are  
NOT VERY useful**

ML task

**ML System**

Model

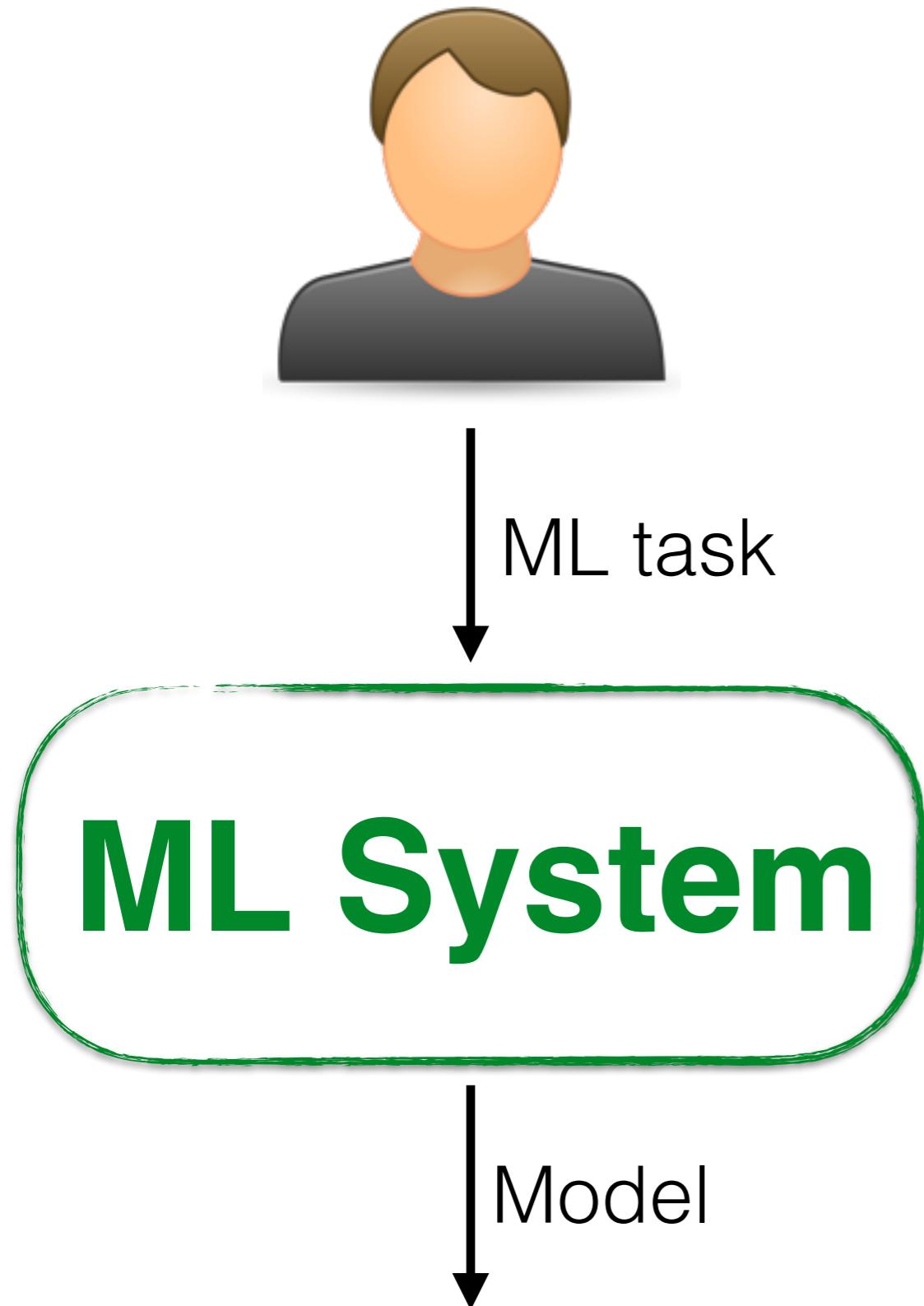
# ML as an Optimization Problem



**ML task:**

$$f(\omega) = \sum_{i \text{ in Data}} \ell_i(\omega) + \mathcal{R}(\omega)$$

# ML as an Optimization Problem



**ML task:**

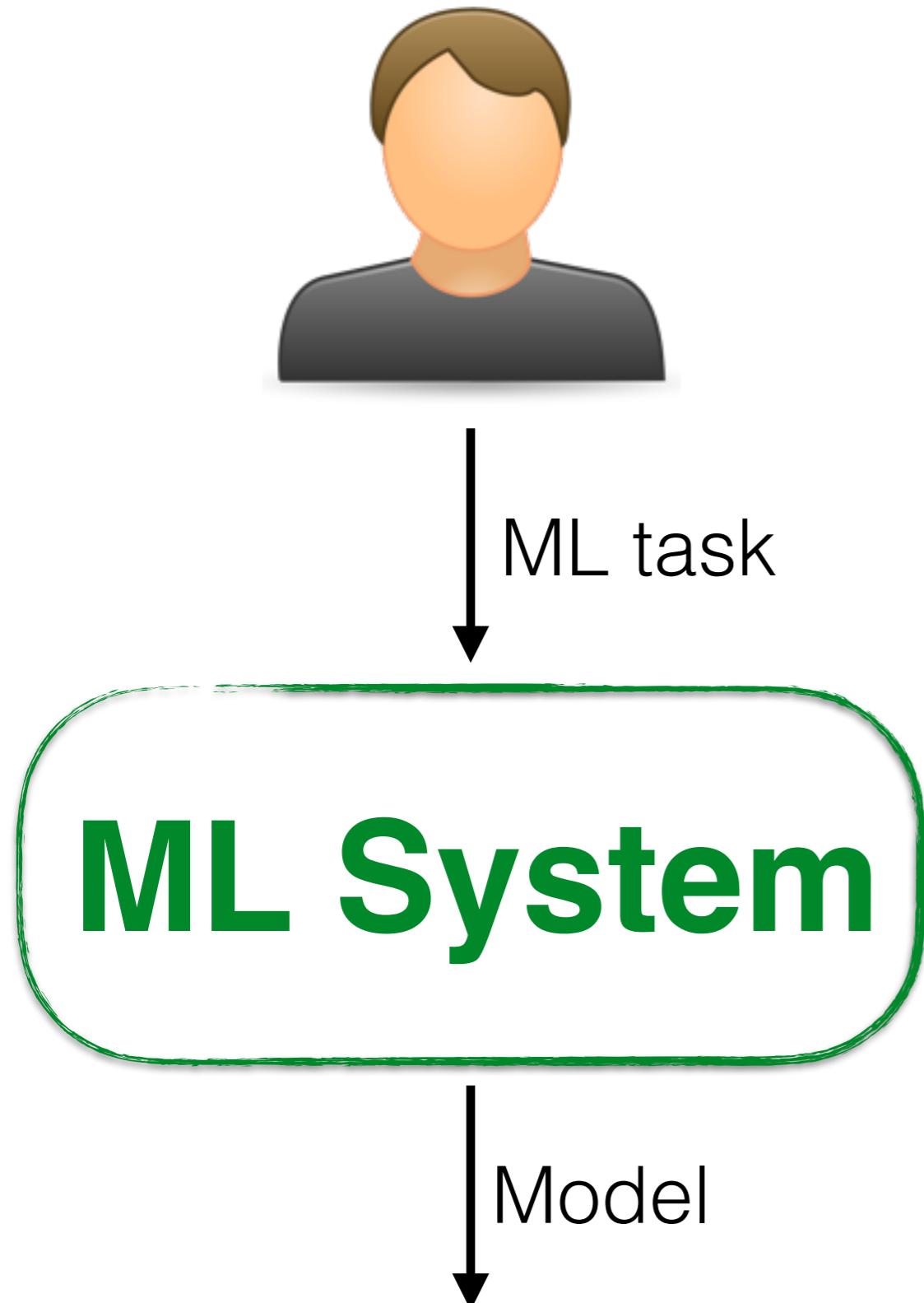
$$f(\omega) = \sum_{i \text{ in Data}} \ell_i(\omega) + \mathcal{R}(\omega)$$

$$\downarrow \min_{\omega} f(\omega)$$

**Solve with a GD algorithm**

$$\omega^{k+1} = \omega^k - a_k \nabla f(\omega^k)$$

# ML as an Optimization Problem



**ML task:**

$$f(\omega) = \sum_{i \text{ in Data}} \ell_i(\omega) + \mathcal{R}(\omega)$$

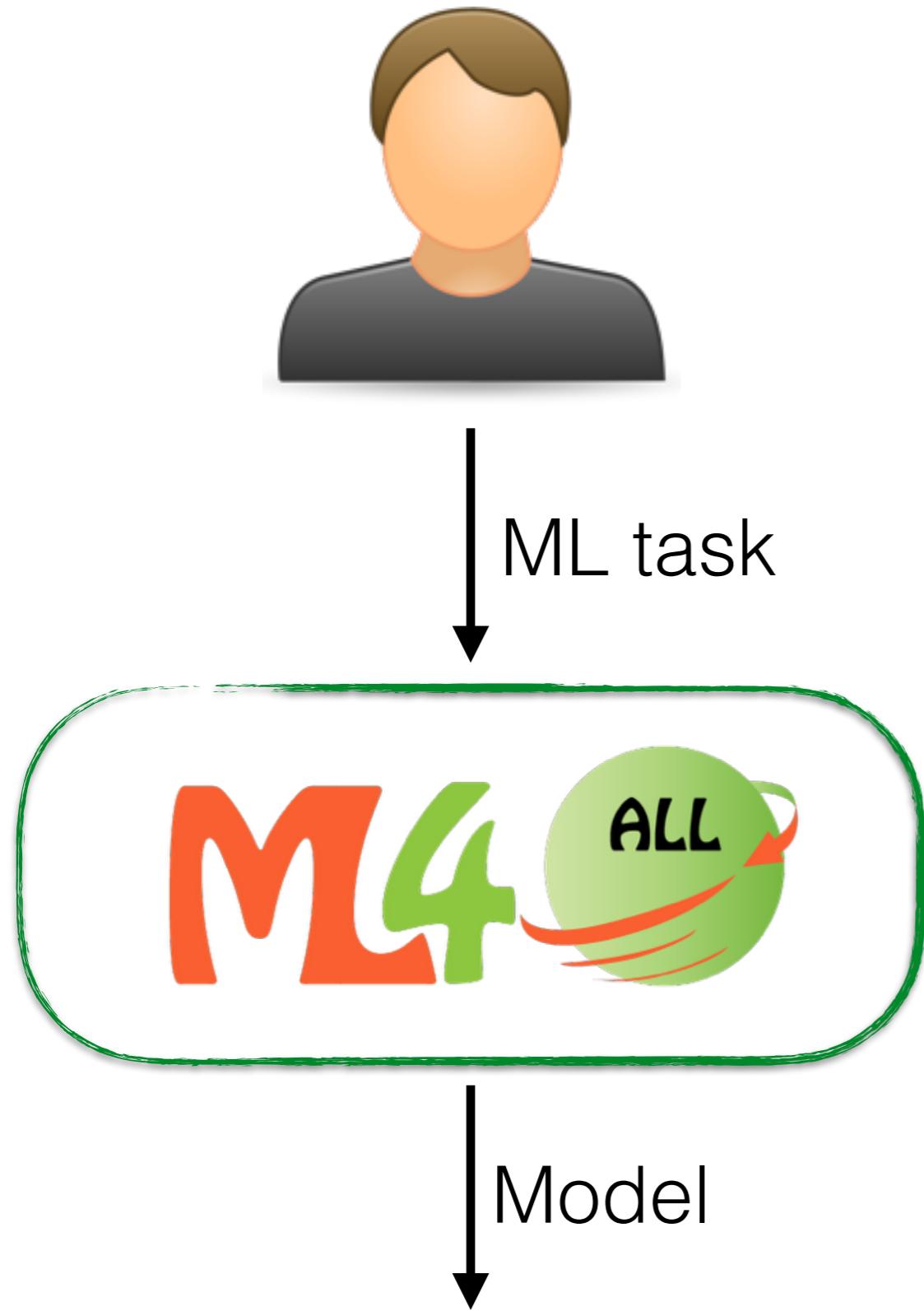
$$\downarrow \min_{\omega} f(\omega)$$

**Solve with a GD algorithm**

$$\omega^{k+1} = \omega^k - a_k \nabla f(\omega^k)$$

$$\|\omega^{k+1} - \omega^k\|$$

# ML as an Optimization Problem



**ML task:**

$$f(\omega) = \sum_{i \text{ in Data}} \ell_i(\omega) + \mathcal{R}(\omega)$$

$$\downarrow \min_{\omega} f(\omega)$$

**Solve with a GD algorithm**

$$\omega^{k+1} = \omega^k - a_k \nabla f(\omega^k)$$

$$\|\omega^{k+1} - \omega^k\|$$

# SGD inside a DBMS

[SIGMOD'12]

## Towards a Unified Architecture for in-RDBMS Analytics

Xixuan Feng

Arun Kumar

Benjamin Recht

Christopher Ré

Department of Computer Sciences

University of Wisconsin-Madison

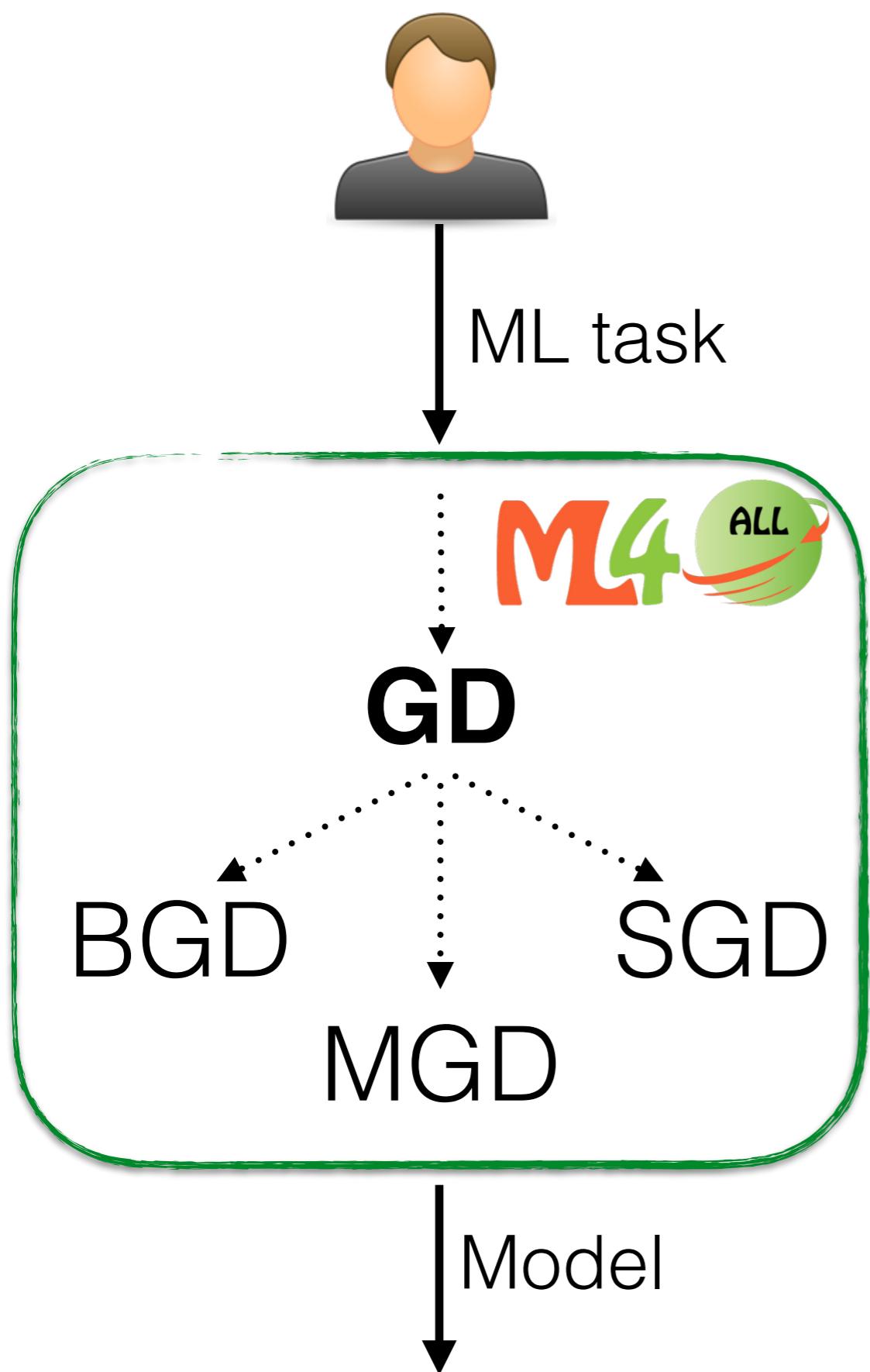
{xfeng, arun, brecht, chrisre}@cs.wisc.edu

### ABSTRACT

The increasing use of statistical data analysis in enterprise applications has created an arms race among database ven-

late 1990s and early 2000s, this brought a wave of data mining toolkits into the RDBMS. Several major vendors are again making an effort toward sophisticated in-database analytics with both open source efforts (e.g., the MADlib plat-

# Which GD Algorithm?



**ML task:**

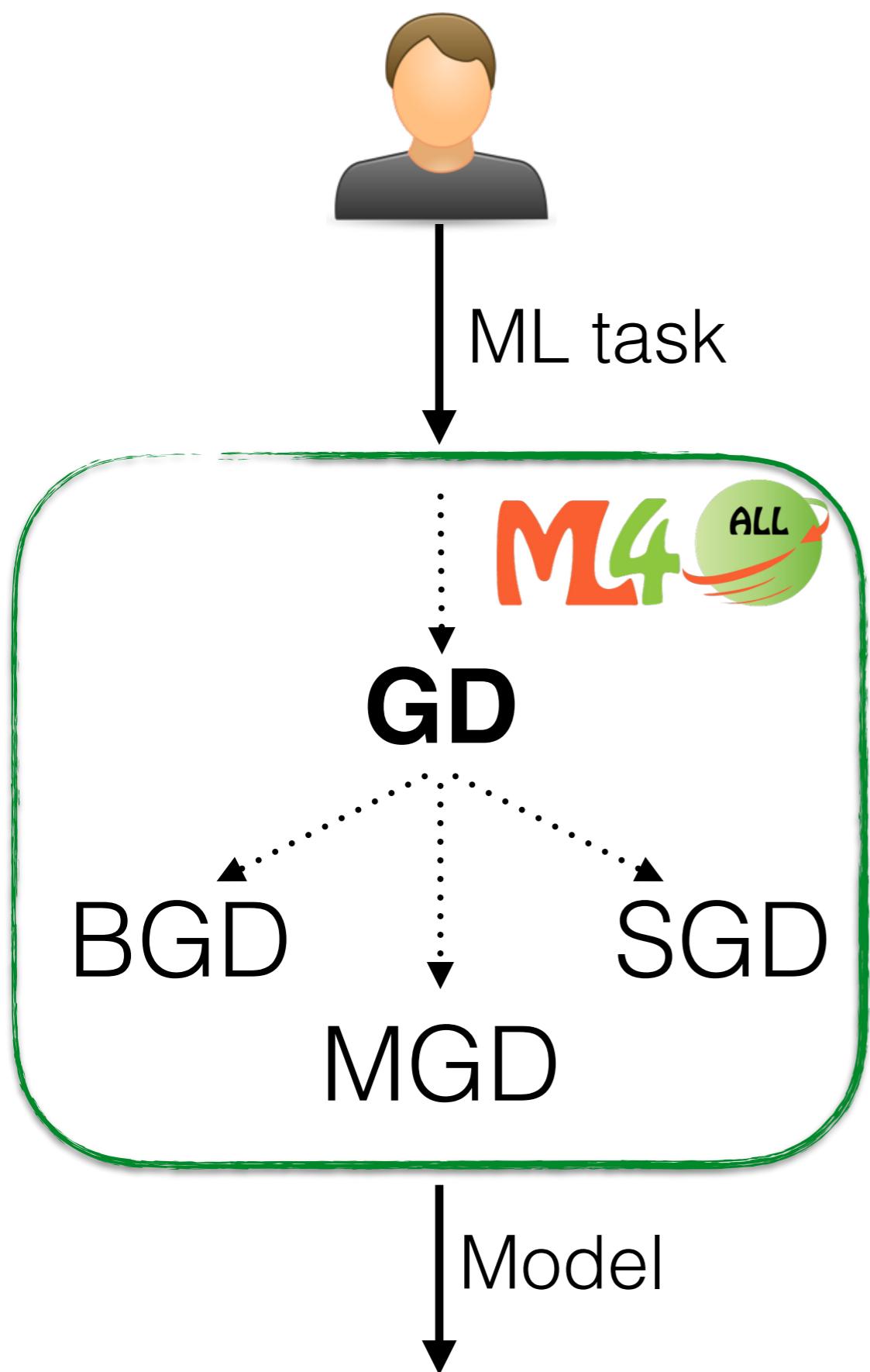
$$f(\omega) = \sum_{i \text{ in Data}} \ell_i(\omega) + \mathcal{R}(\omega)$$

$$\downarrow \min_{\omega} f(\omega)$$

**Solve with a GD algorithm**

$$\omega^{k+1} = \omega^k - a_k \nabla f(\omega^k)$$

# Which GD Algorithm?



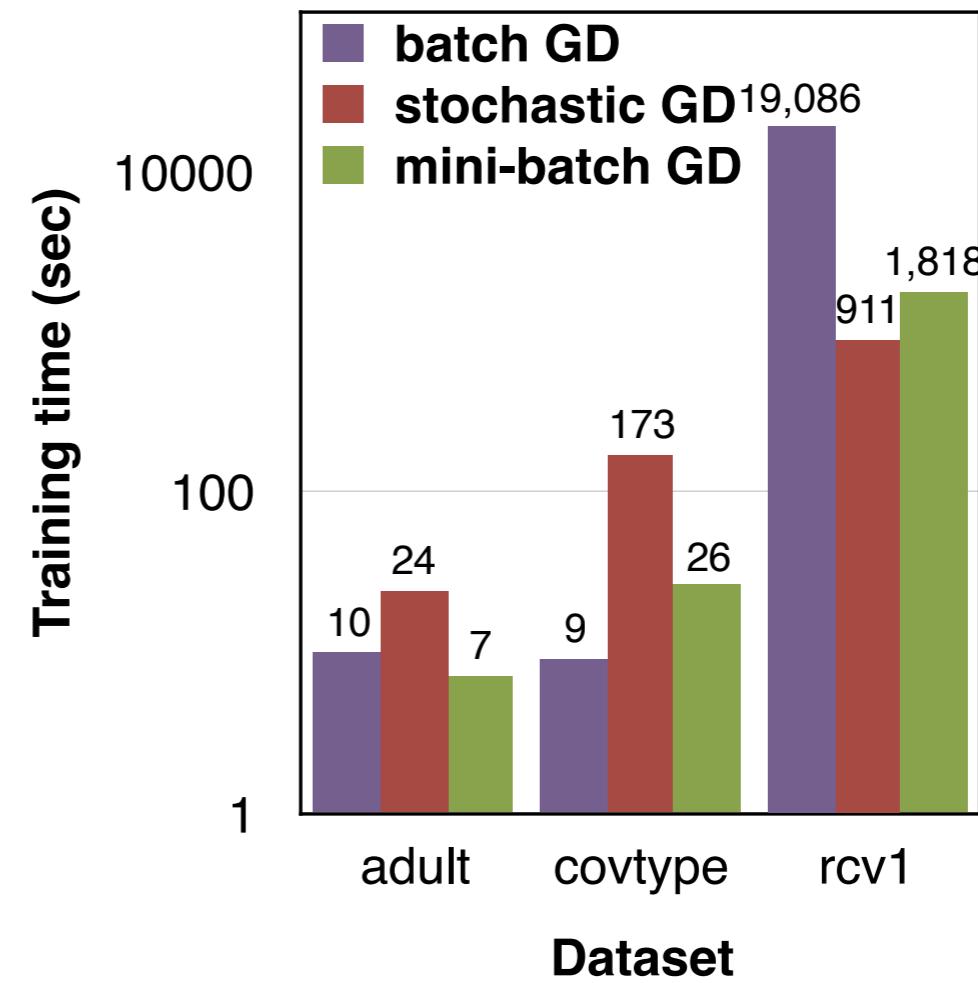
**ML task:**

$$f(\omega) = \sum_{i \text{ in Data}} \ell_i(\omega) + \mathcal{R}(\omega)$$

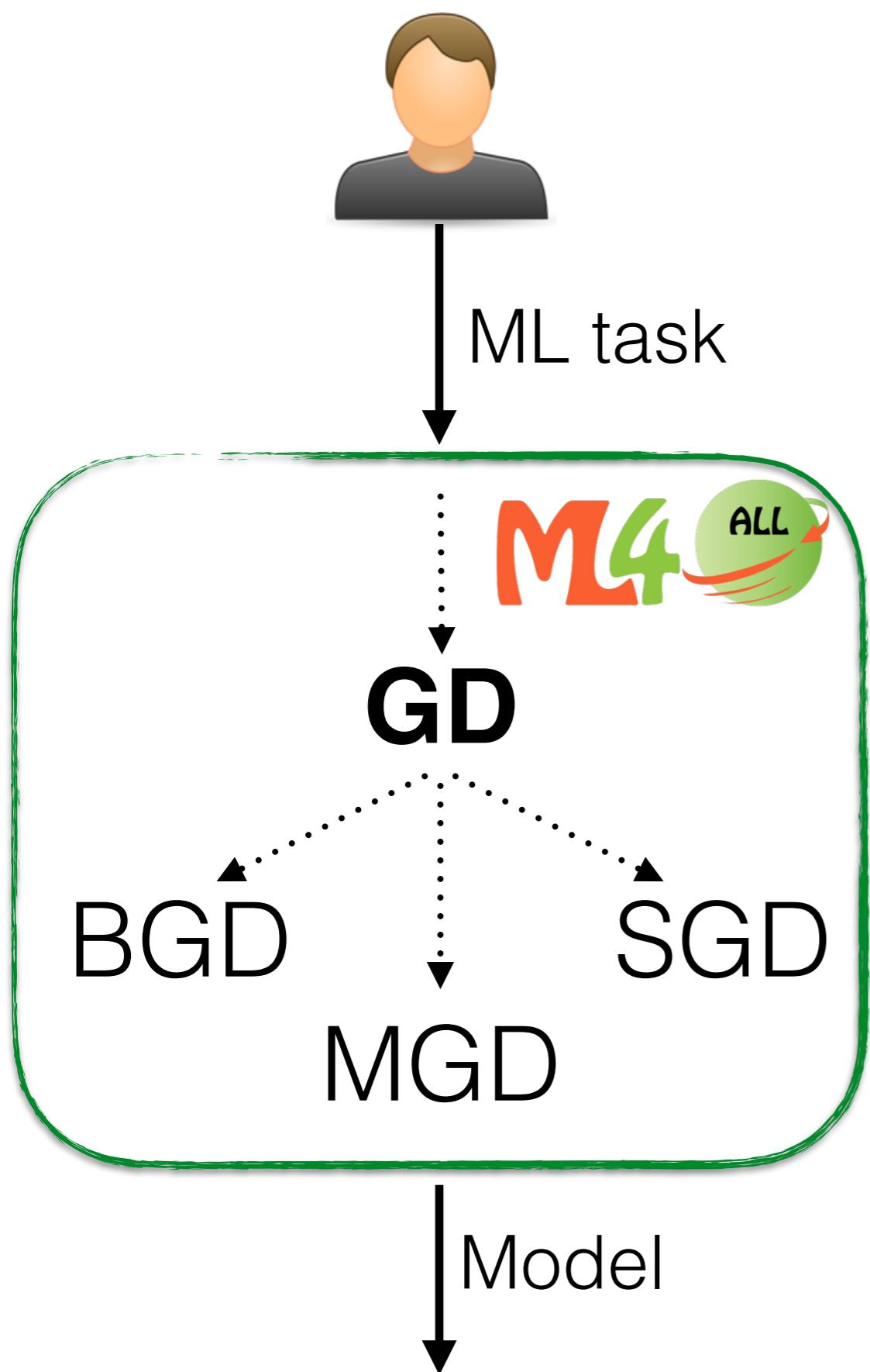
$$\downarrow \min_{\omega} f(\omega)$$

**Solve with a GD algorithm**

$$\omega^{k+1} = \omega^k - a_k \nabla f(\omega^k)$$



# Which GD Algorithm?



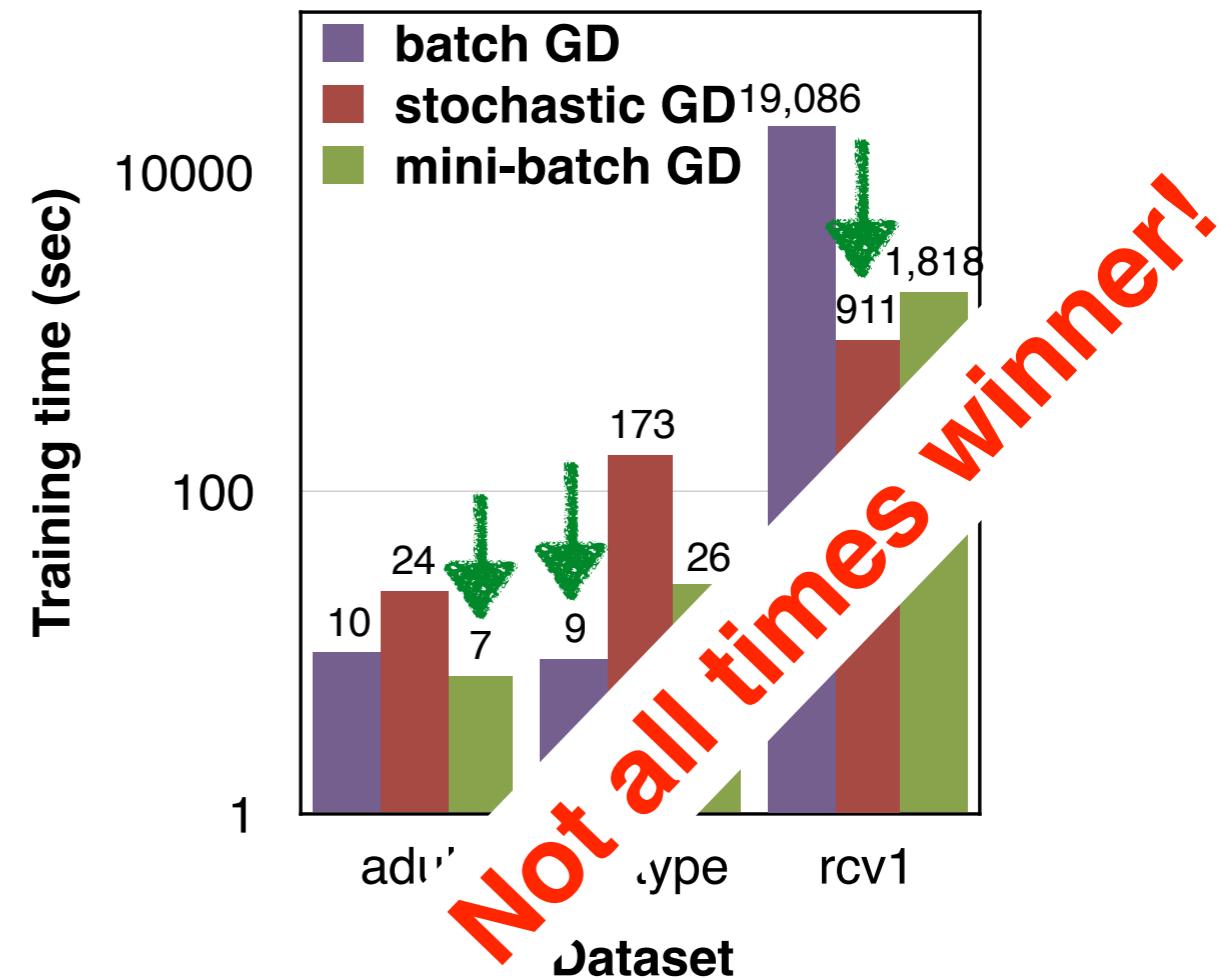
**ML task:**

$$f(\omega) = \sum_{i \text{ in Data}} \ell_i(\omega) + \mathcal{R}(\omega)$$

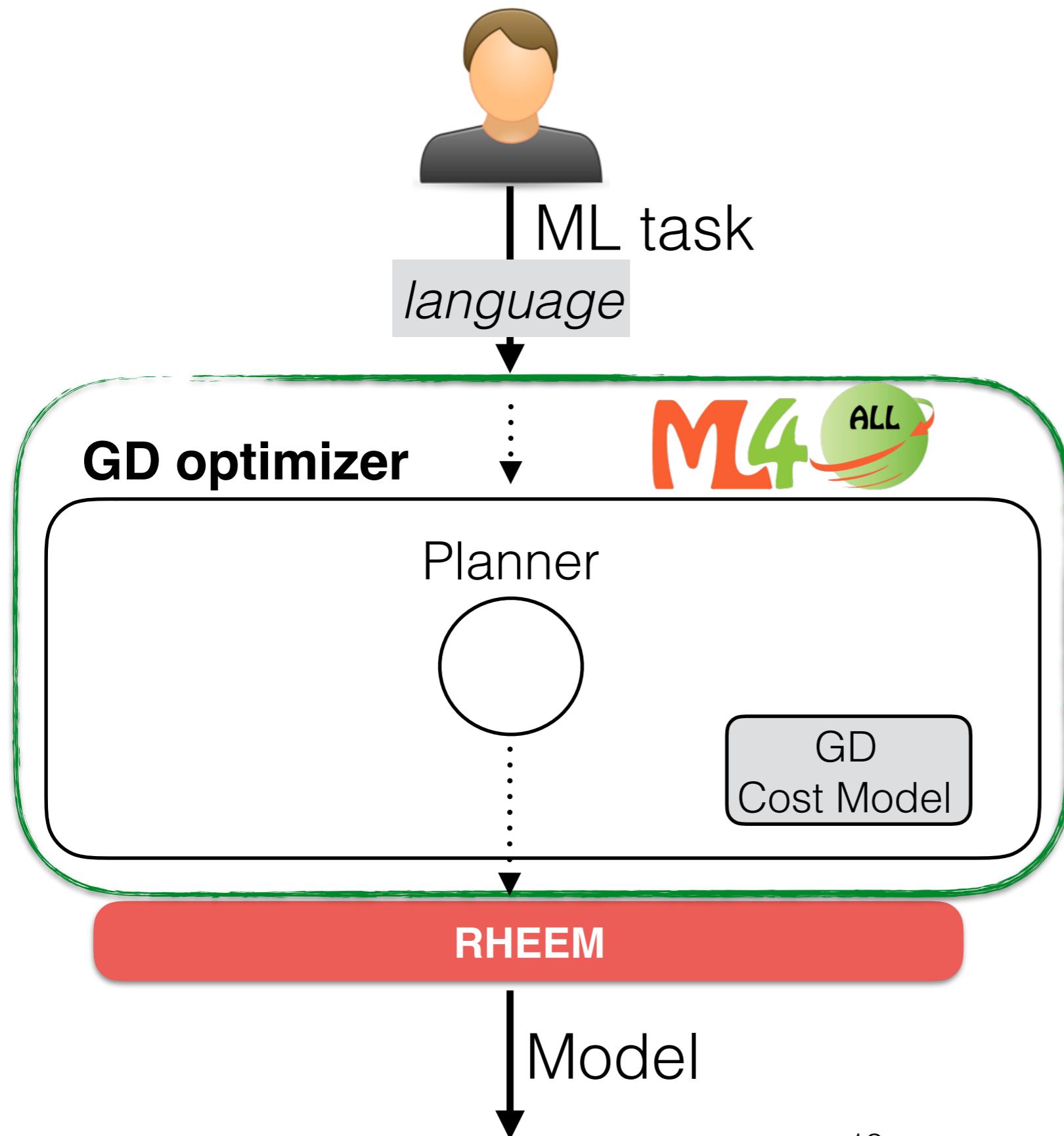
$$\downarrow \min_{\omega} f(\omega)$$

**Solve with a GD algorithm**

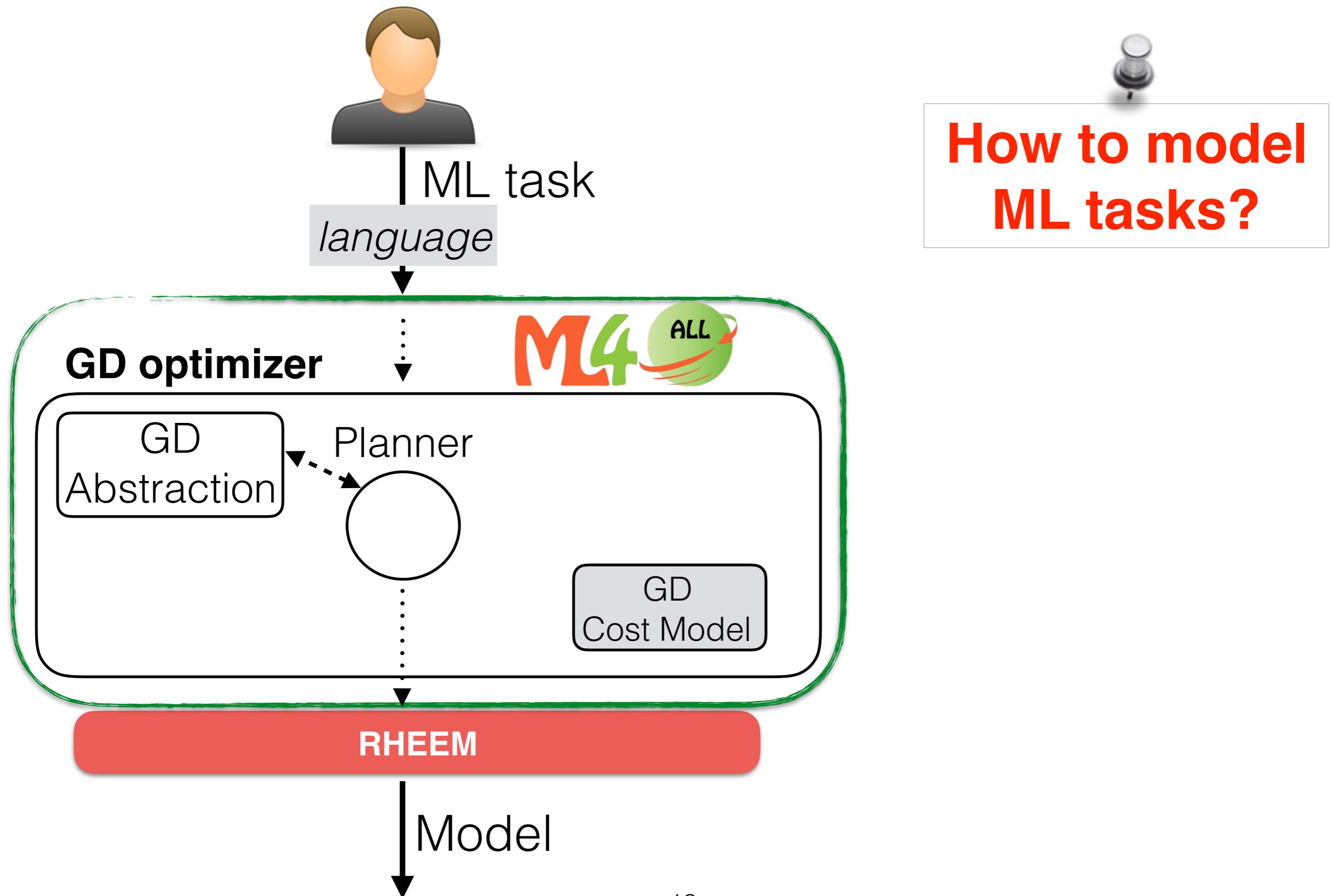
$$\omega^{K+1} = \omega^K - a_K \nabla f(\omega^K)$$



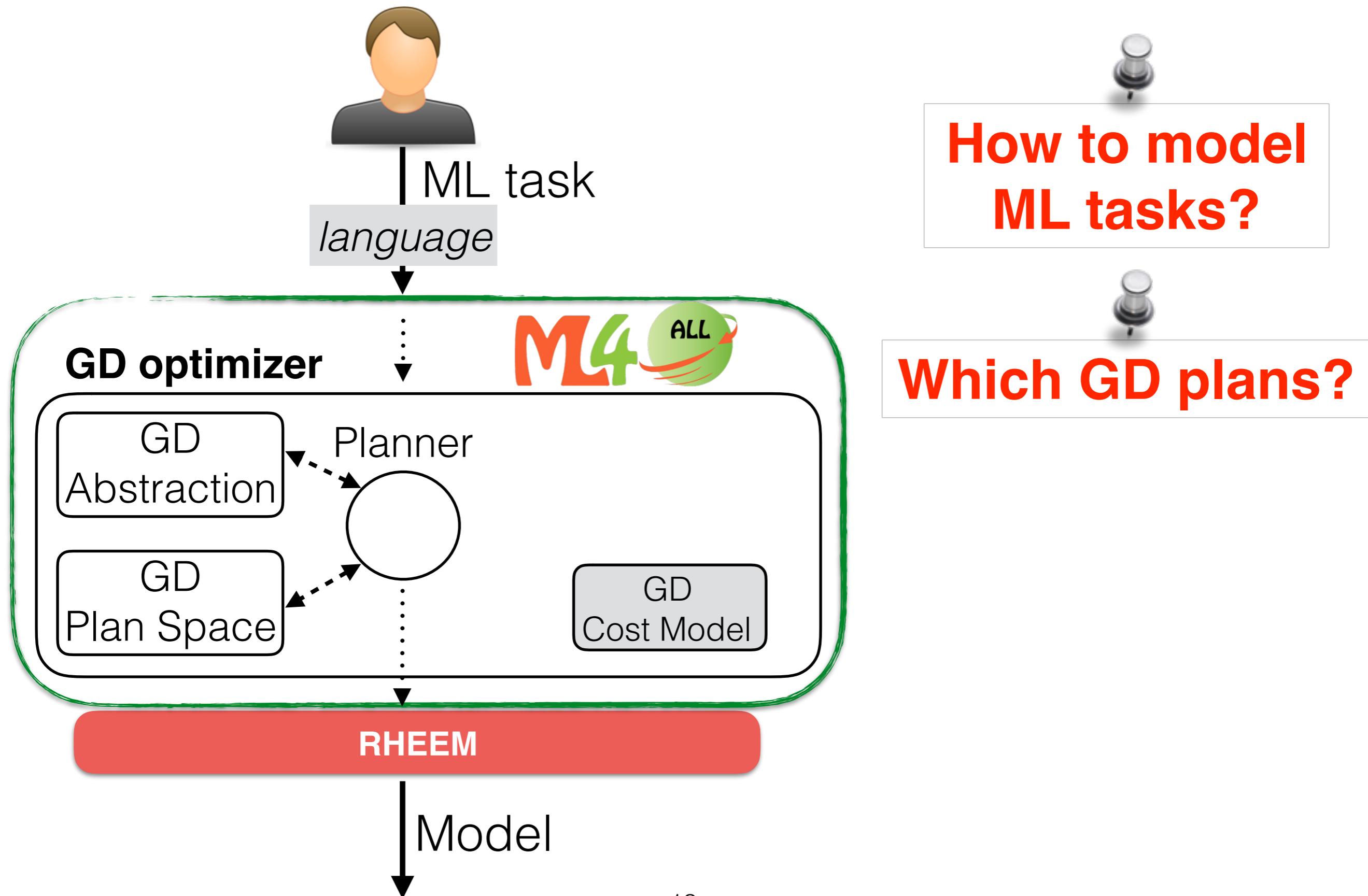
# Towards a DB-like ML System



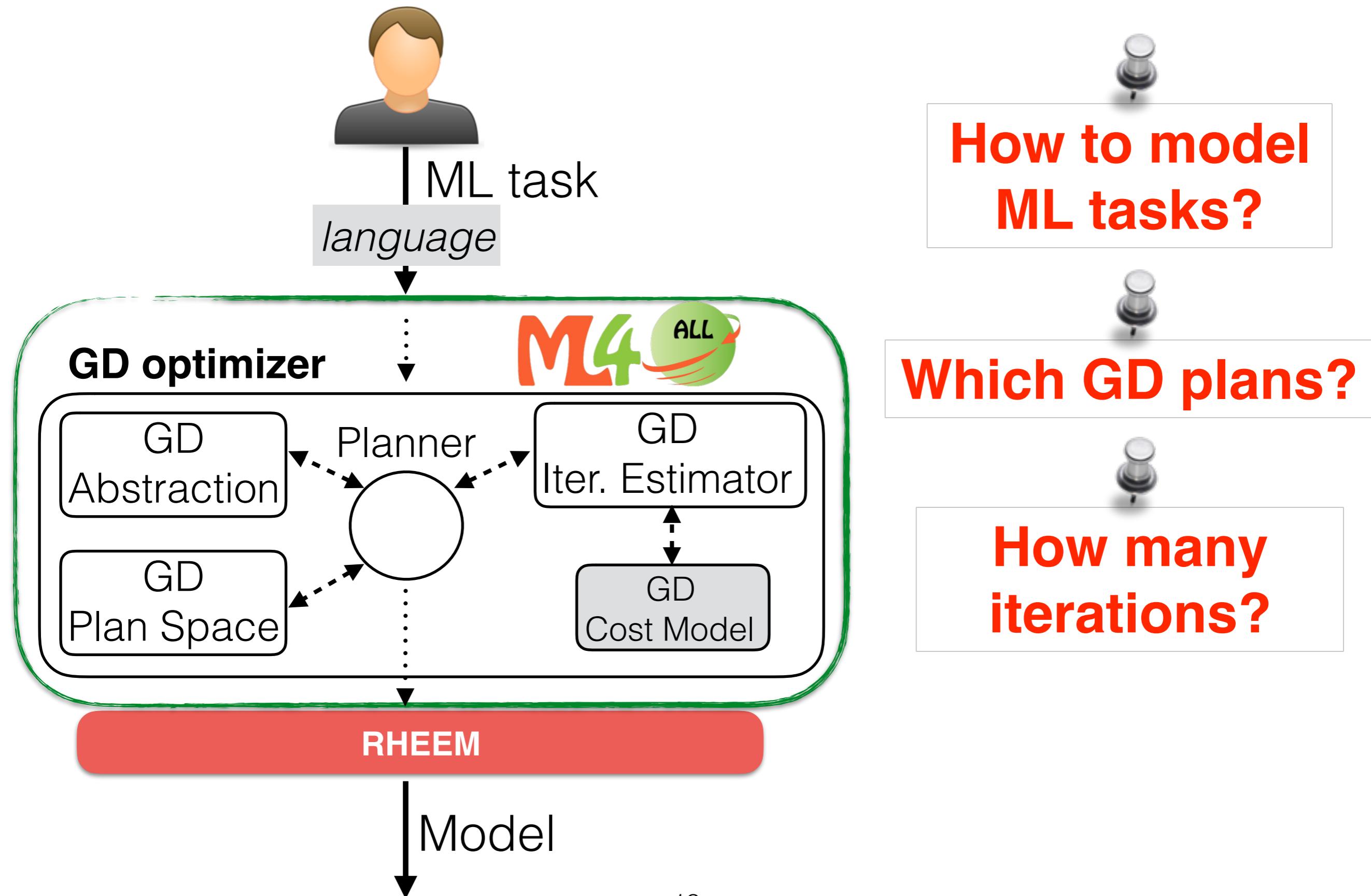
# Towards a DB-like ML System



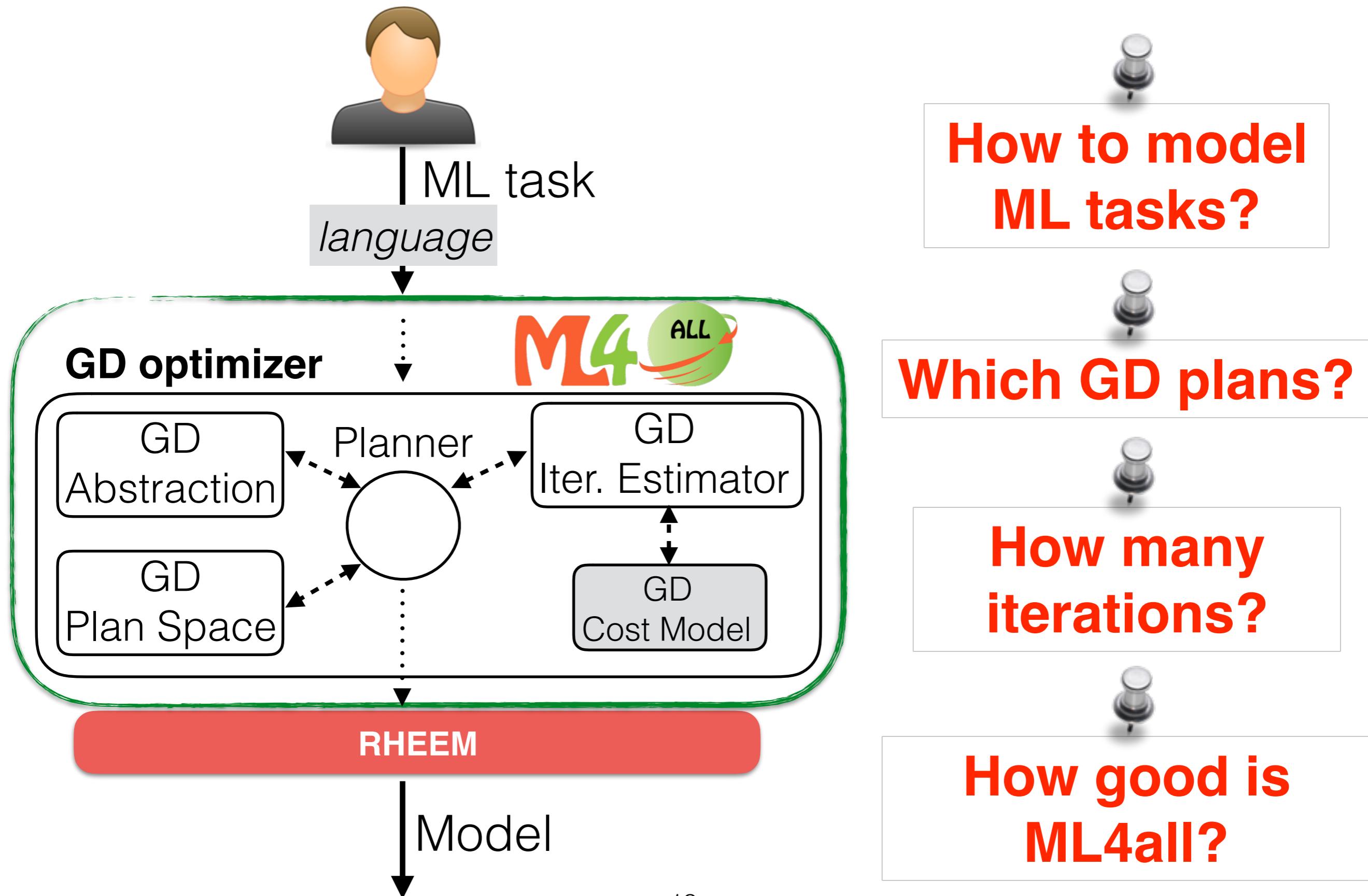
# Towards a DB-like ML System



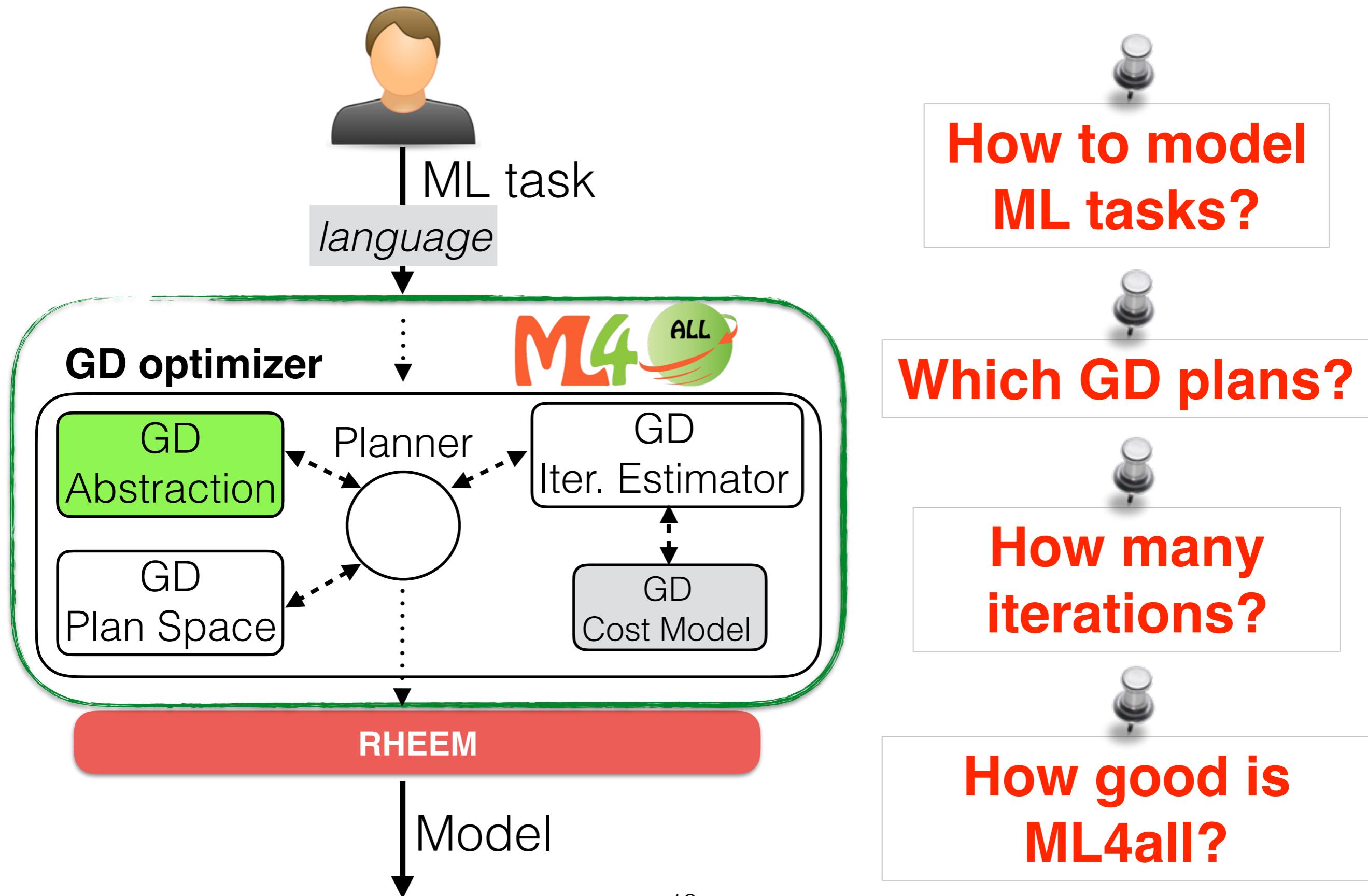
# Towards a DB-like ML System



# Towards a DB-like ML System



# Towards a DB-like ML System



# Preparation Phase

*SGD Plan*

# Preparation Phase

*SGD Plan*

## Data units

```
+1 2:0.1 4:0.4 10:0.3  
-1 3:0.3 4:0.5 9:0.5  
+1 1:0.1 2:0.7 6:0.2
```



**Transform**

	label	indices	values
1	[2, 4, 10]	[0.1, 0.4, 0.3]	
-1	[3, 4, 9]	[0.3, 0.5, 0.6]	
1	[1, 2, 6]	[0.1, 0.7, 0.2]	

# Preparation Phase

*SGD Plan*

## Data units

```
+1 2:0.1 4:0.4 10:0.3  
-1 3:0.3 4:0.5 9:0.5  
+1 1:0.1 2:0.7 6:0.2
```

Transform

Stage

	label	indices	values
1	[2, 4, 10]	[0.1, 0.4, 0.3]	
-1	[3, 4, 9]	[0.3, 0.5, 0.6]	
1	[1, 2, 6]	[0.1, 0.7, 0.2]	

i:=0

step:=1.0

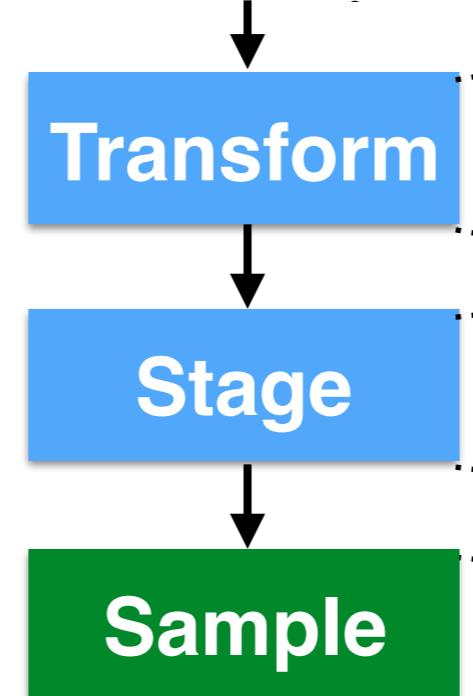
w:=[0.0, 0.0, ..., 0.0]

# Processing Phase

*SGD Plan*

## Data units

```
+1 2:0.1 4:0.4 10:0.3  
-1 3:0.3 4:0.5 9:0.5  
+1 1:0.1 2:0.7 6:0.2
```



	label	indices	values
1	[2, 4, 10]	[0.1, 0.4, 0.3]	
-1	[3, 4, 9]	[0.3, 0.5, 0.6]	
1	[1, 2, 6]	[0.1, 0.7, 0.2]	

i:=0  
step:=1.0  
w:=[0.0, 0.0, ..., 0.0]

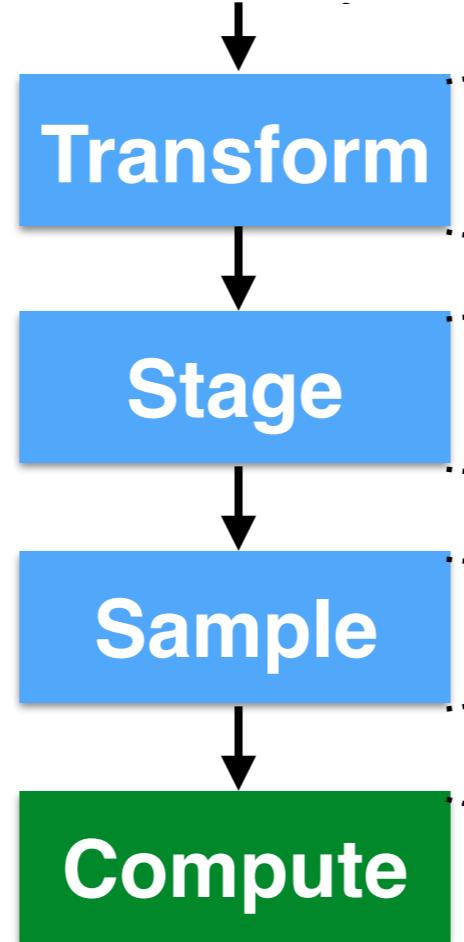
1	[2, 4, 10]	[0.1, 0.4, 0.3]
---	------------	-----------------

# Processing Phase

*SGD Plan*

## Data units

```
+1 2:0.1 4:0.4 10:0.3  
-1 3:0.3 4:0.5 9:0.5  
+1 1:0.1 2:0.7 6:0.2
```



label	indices	values
1	[2, 4, 10]	[0.1, 0.4, 0.3]
-1	[3, 4, 9]	[0.3, 0.5, 0.6]
1	[1, 2, 6]	[0.1, 0.7, 0.2]

$i := 0$   
 $step := 1.0$   
 $w := [0.0, 0.0, \dots, 0.0]$

1	[2, 4, 10]	[0.1, 0.4, 0.3]
---	------------	-----------------

$$\nabla(1 | [2, 4, 10] | [0.1, 0.4, 0.3], w)$$

# Processing Phase

**SGD Plan**

## Data units

```
+1 2:0.1 4:0.4 10:0.3  
-1 3:0.3 4:0.5 9:0.5  
+1 1:0.1 2:0.7 6:0.2
```

Transform

Stage

Sample

Compute

Update

	label	indices	values
1	[2, 4, 10]	[0.1, 0.4, 0.3]	
-1	[3, 4, 9]	[0.3, 0.5, 0.6]	
1	[1, 2, 6]	[0.1, 0.7, 0.2]	

i:=0

step:=1.0

w:=[0.0, 0.0, ..., 0.0]

1	[2, 4, 10]	[0.1, 0.4, 0.3]
---	------------	-----------------

$$\nabla \left( 1 | [2, 4, 10] | [0.1, 0.4, 0.3], w \right)$$

w:=w - step x  $\nabla_{\text{sum}}$   
i := i+1

# Convergence Phase

**SGD Plan**

## Data units

```
+1 2:0.1 4:0.4 10:0.3  
-1 3:0.3 4:0.5 9:0.5  
+1 1:0.1 2:0.7 6:0.2
```

label	indices	values
1	[2, 4, 10]	[0.1, 0.4, 0.3]
-1	[3, 4, 9]	[0.3, 0.5, 0.6]
1	[1, 2, 6]	[0.1, 0.7, 0.2]

Transform

Stage

Sample

Compute

Update

i:=0  
step:=1.0  
w:=[0.0, 0.0, ..., 0.0]

1 [2, 4, 10] [0.1, 0.4, 0.3]

$\nabla(1 | [2, 4, 10] | [0.1, 0.4, 0.3], w)$

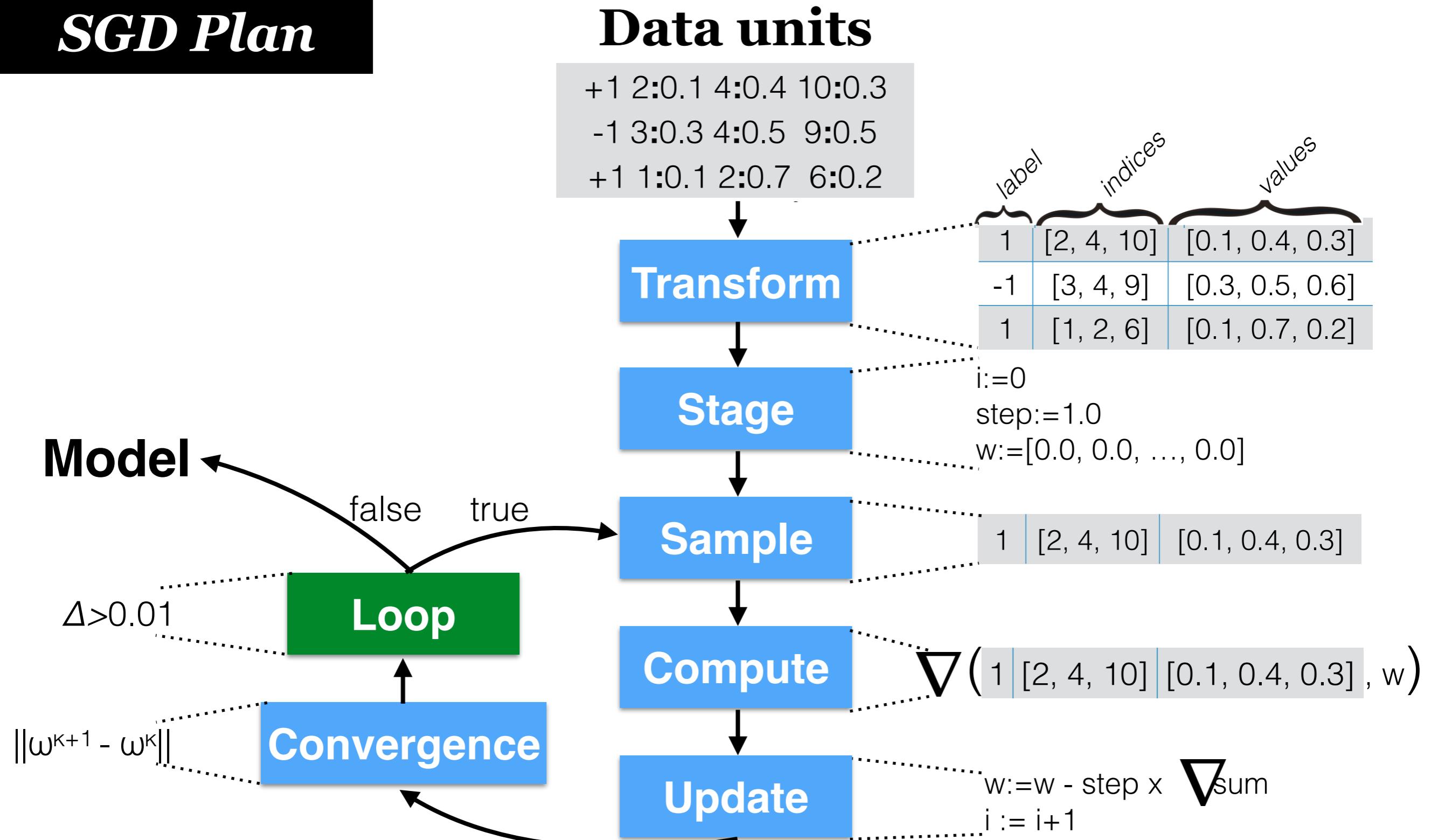
w:=w - step x  $\nabla_{\text{sum}}$   
i := i+1

Convergence

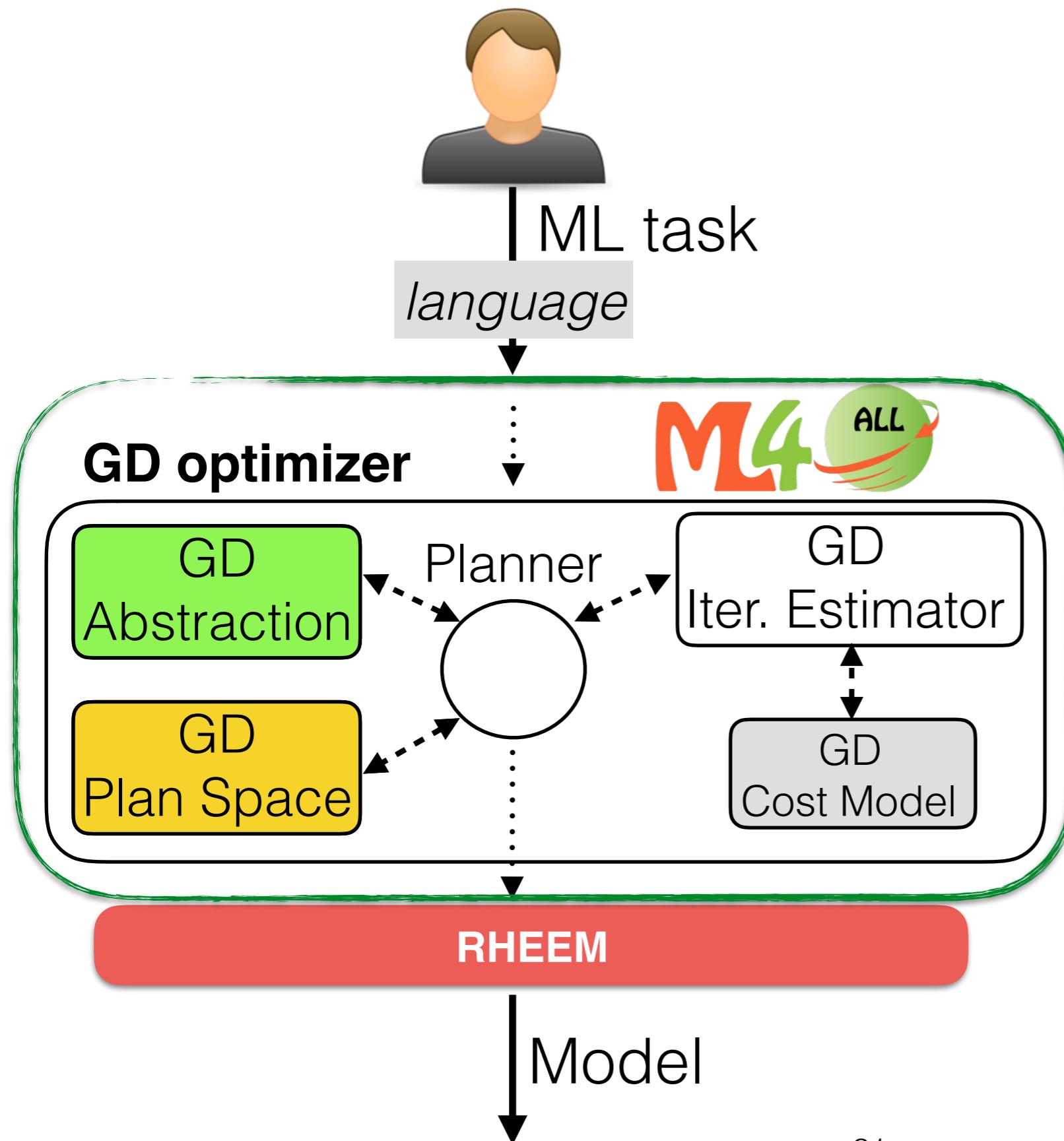
$\omega^{k+1} - \omega^k$

# Convergence Phase

**SGD Plan**



# Towards a DB-like System



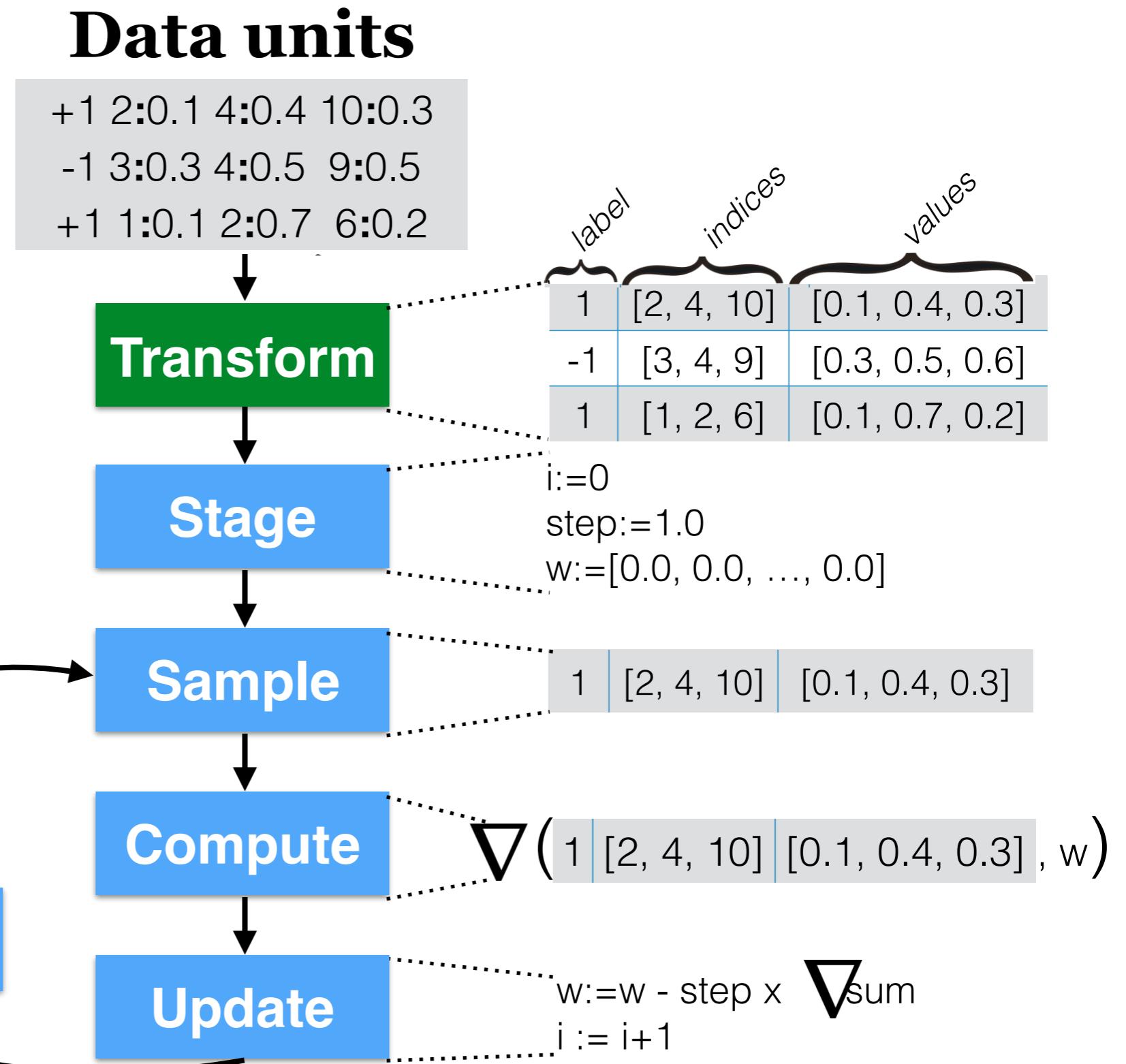
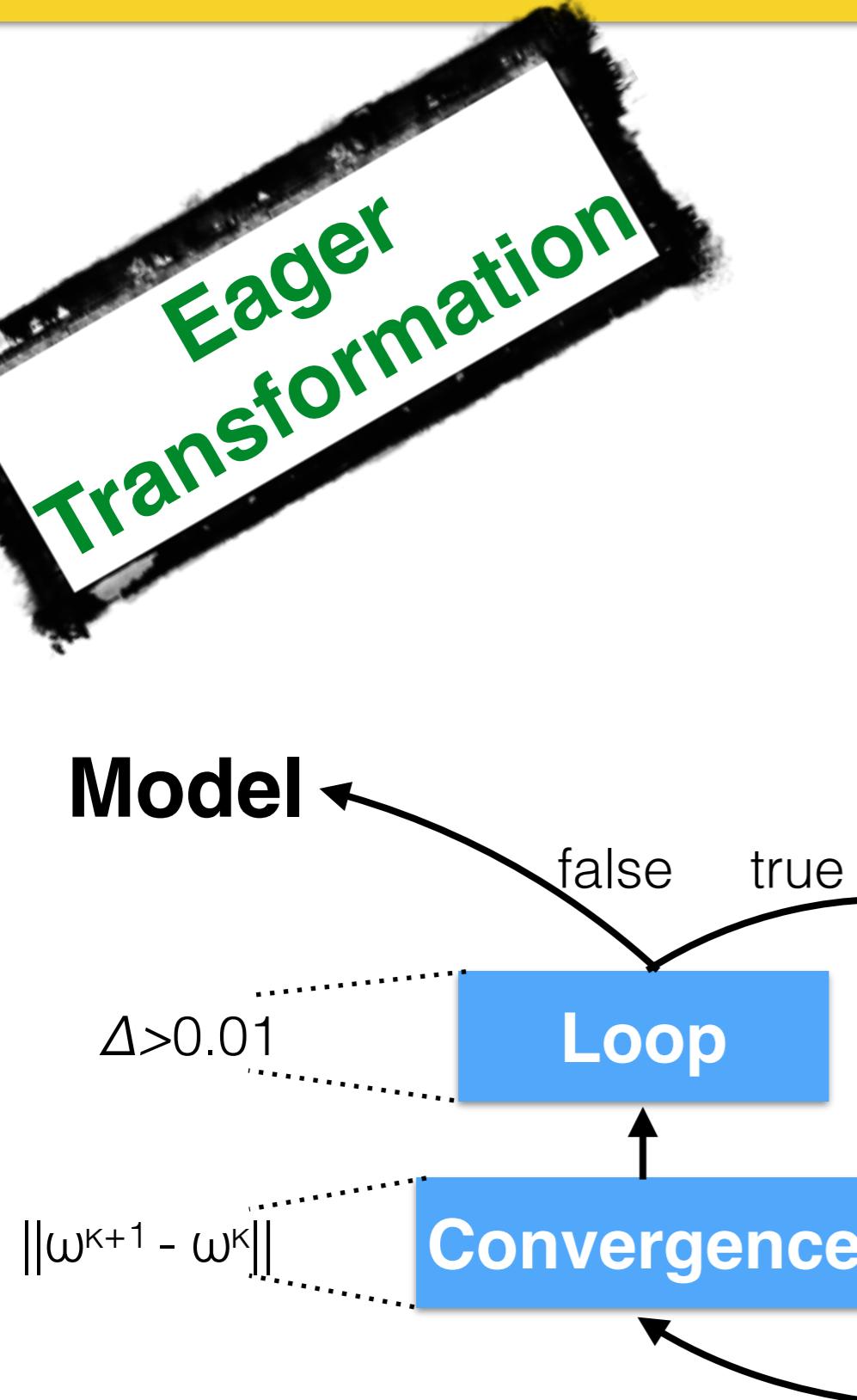
How to model  
ML tasks?

Which GD plans?

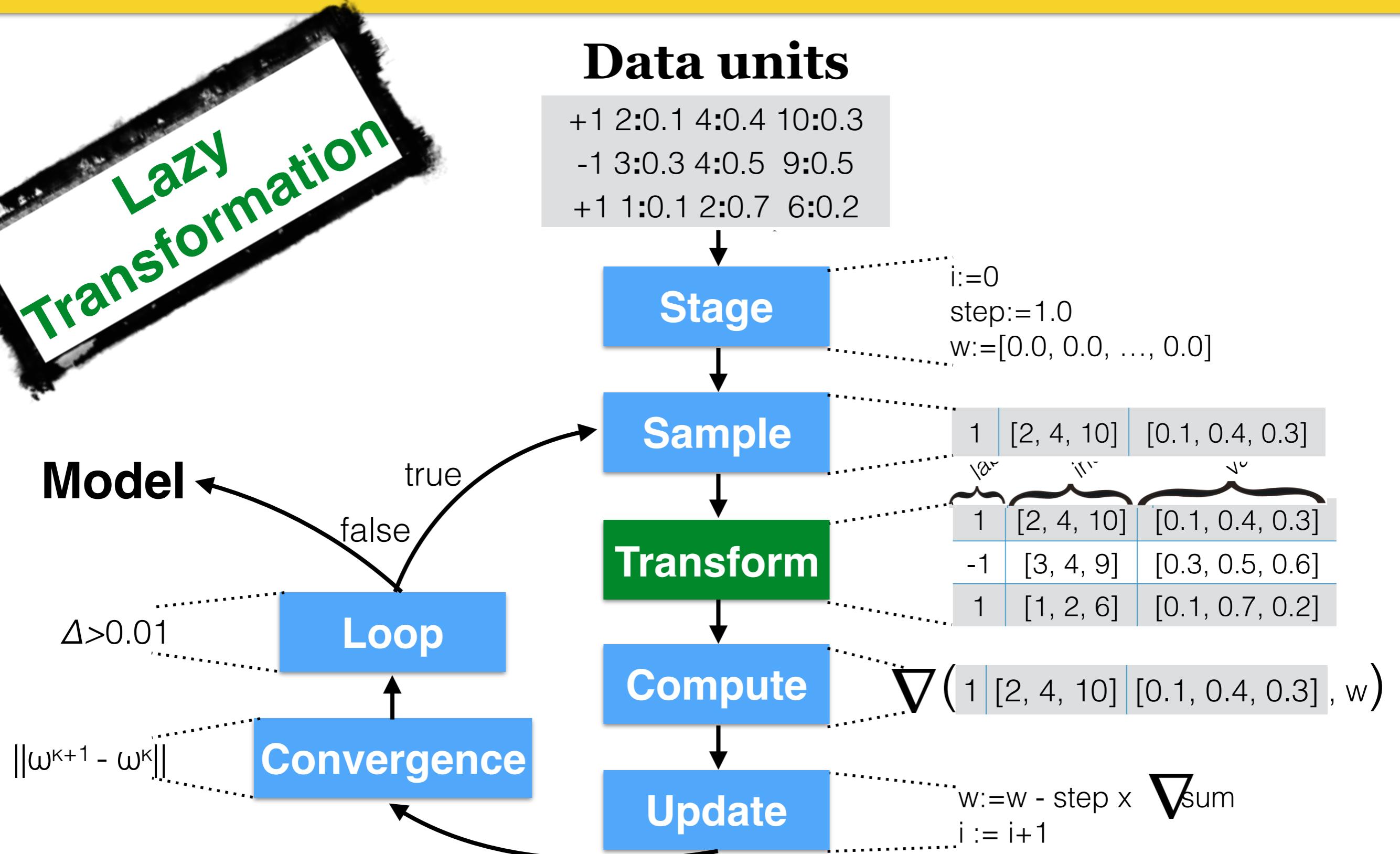
How many  
iterations?

How good is  
ML4all?

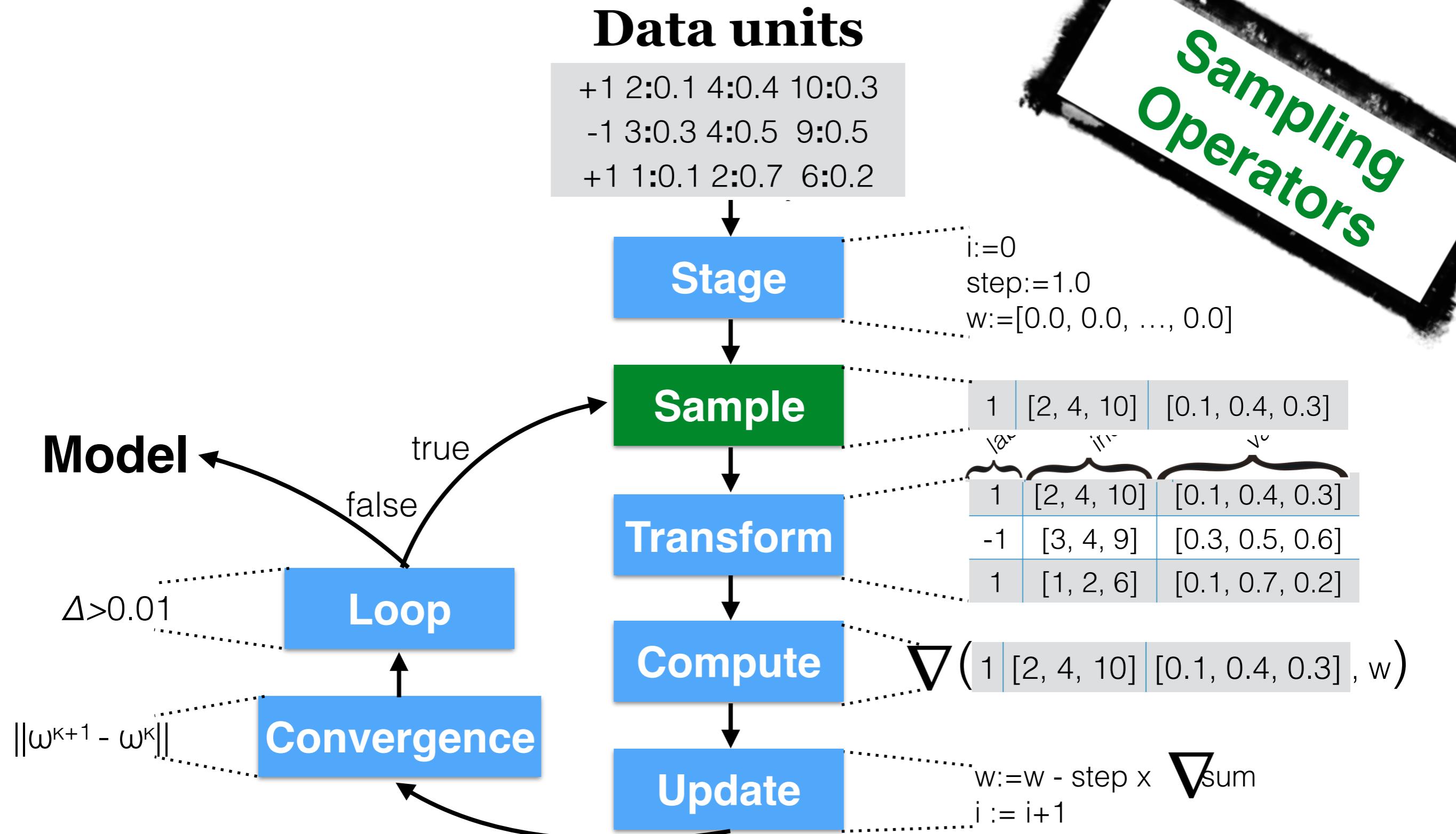
# Optimized GD Plans



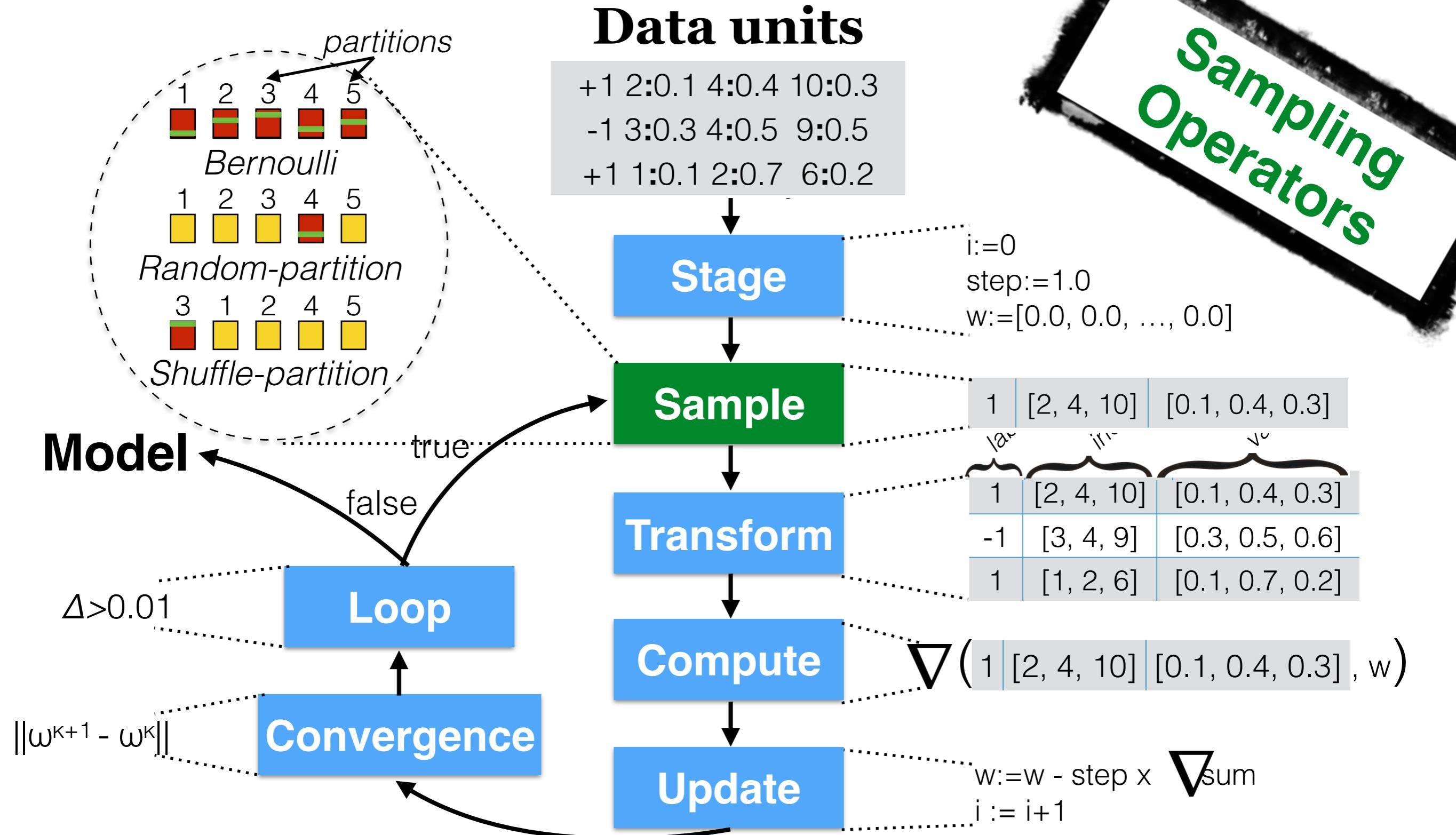
# Optimized GD Plans



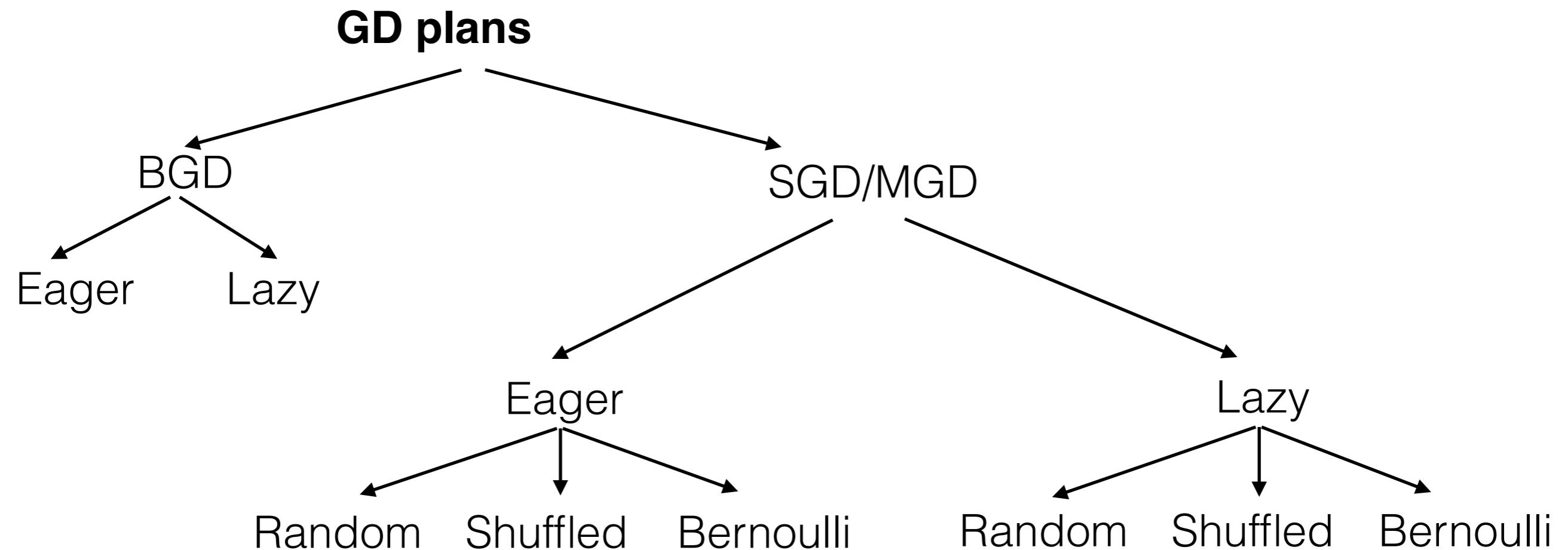
# Optimized GD Plans



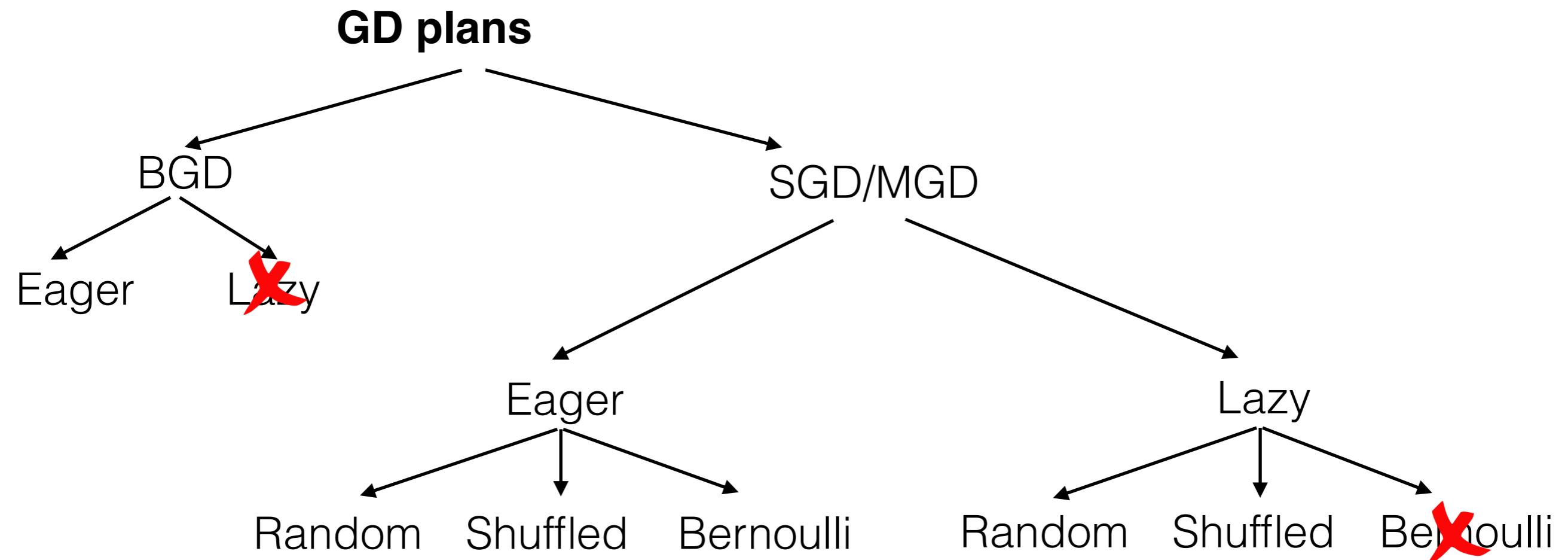
# Optimized GD Plans



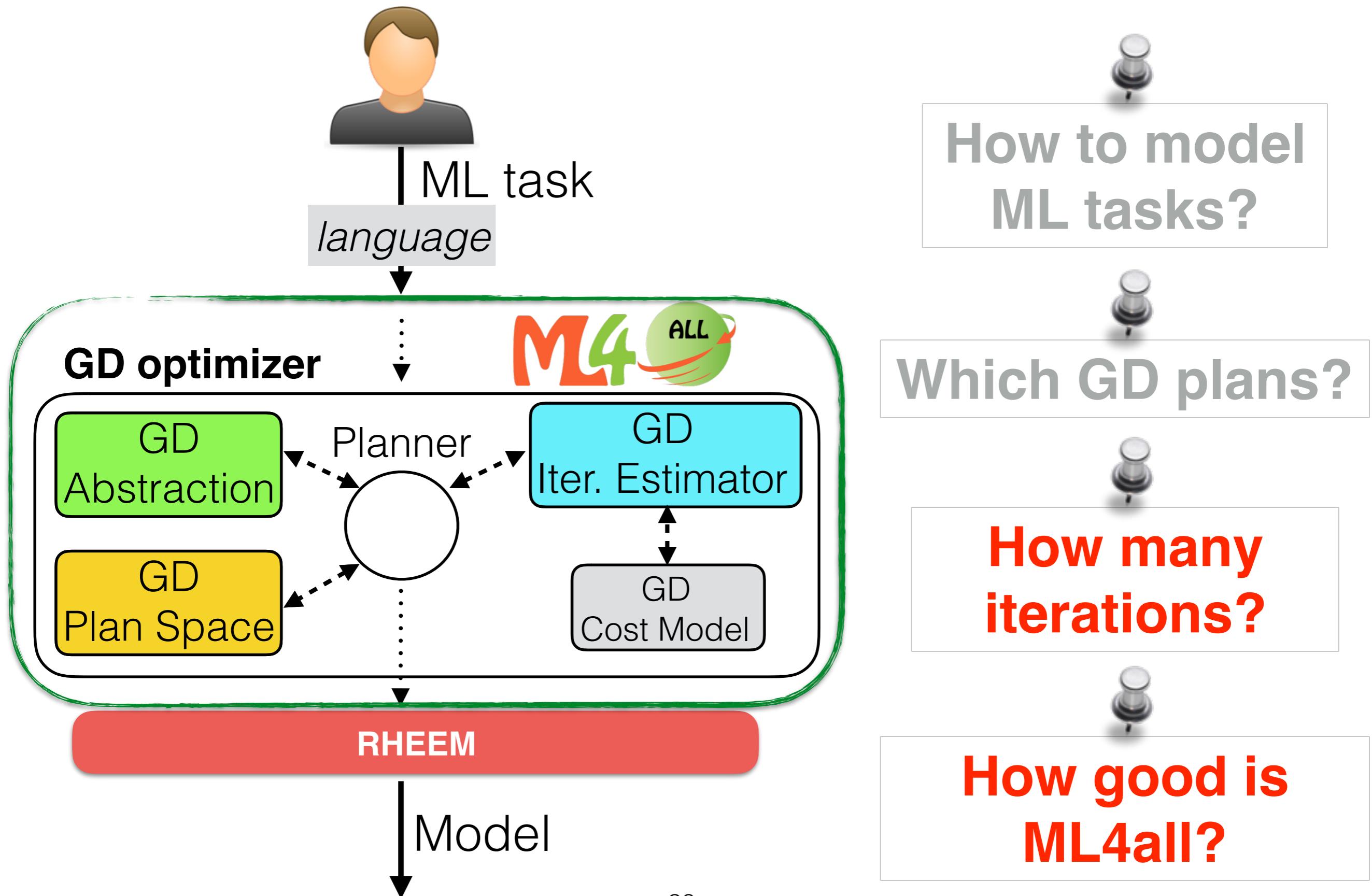
# GD Search Space



# GD Search Space

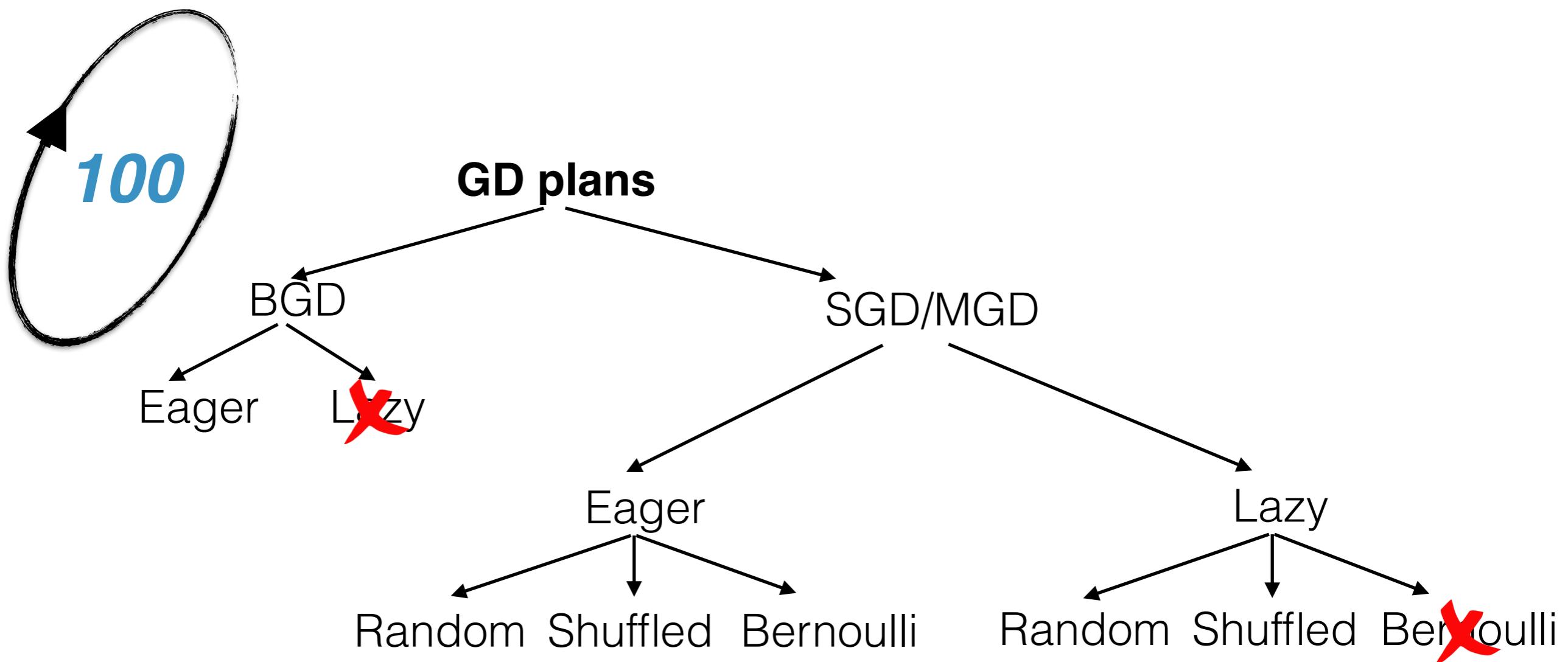


# Towards a DB-like System



# Cost Model

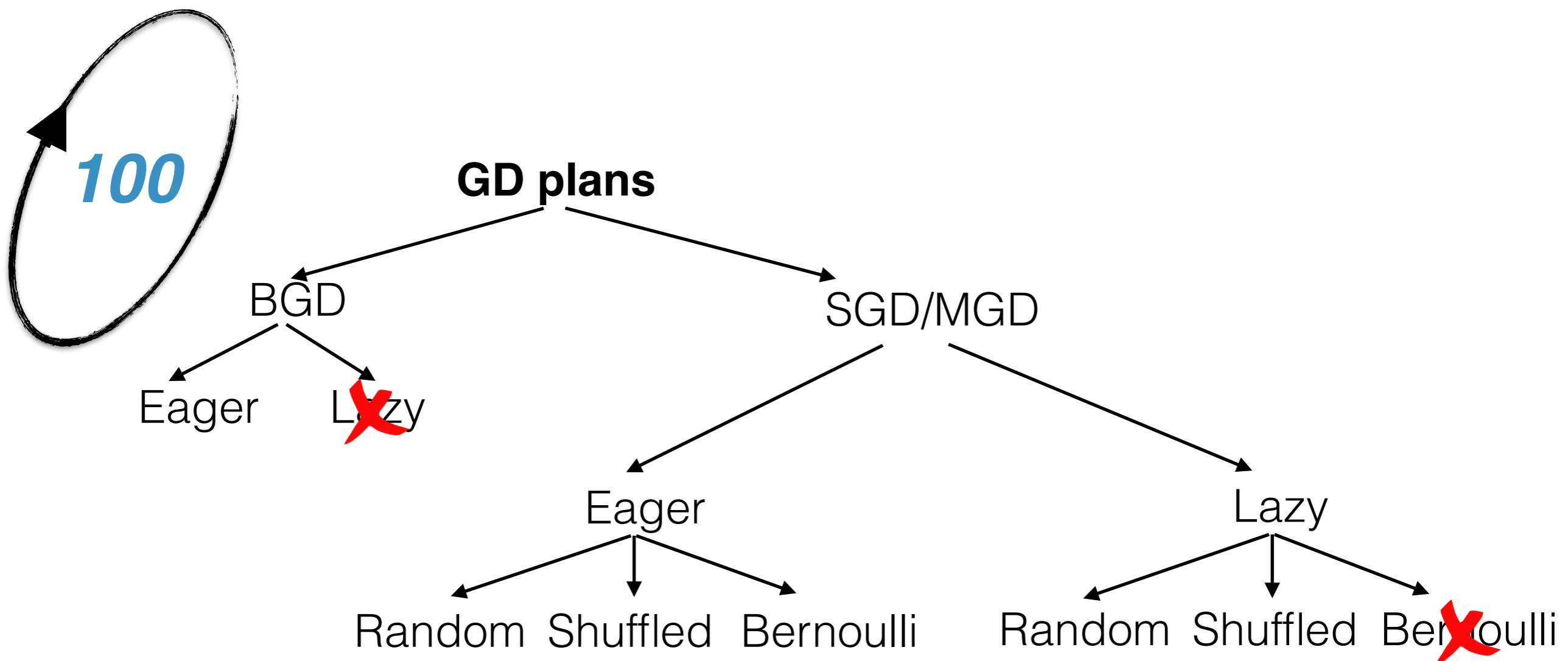
Given #iterations



# Cost Model

## Cost of BGD plan

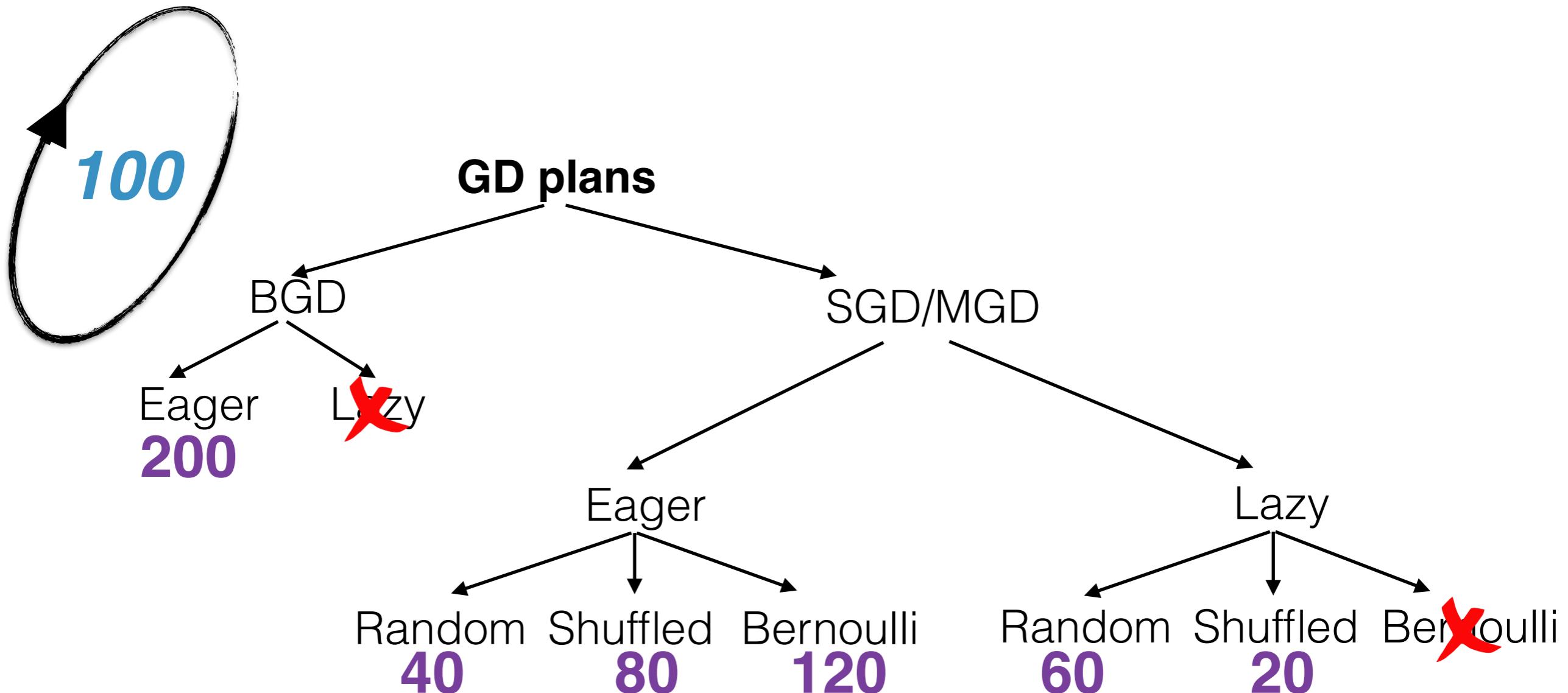
$$C_{BGD}(D) = c_S(D) + c_T(D) + T \times (c_C(D) + c_U + c_D(D) + c_L)$$



# Cost Model

## Cost of BGD plan

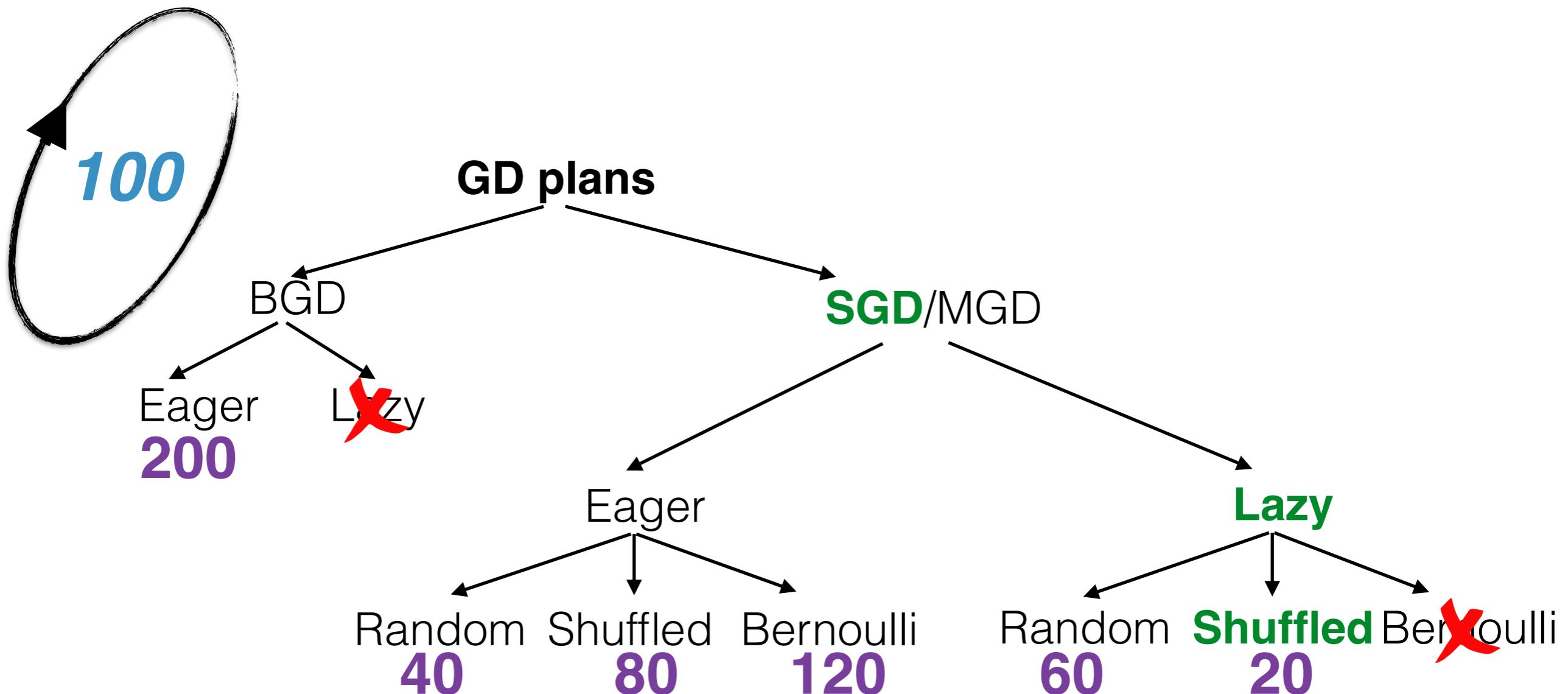
$$C_{BGD}(D) = c_S(D) + c_T(D) + T \times (c_C(D) + c_U + c_D(D) + c_L)$$



# Cost Model

## Cost of BGD plan

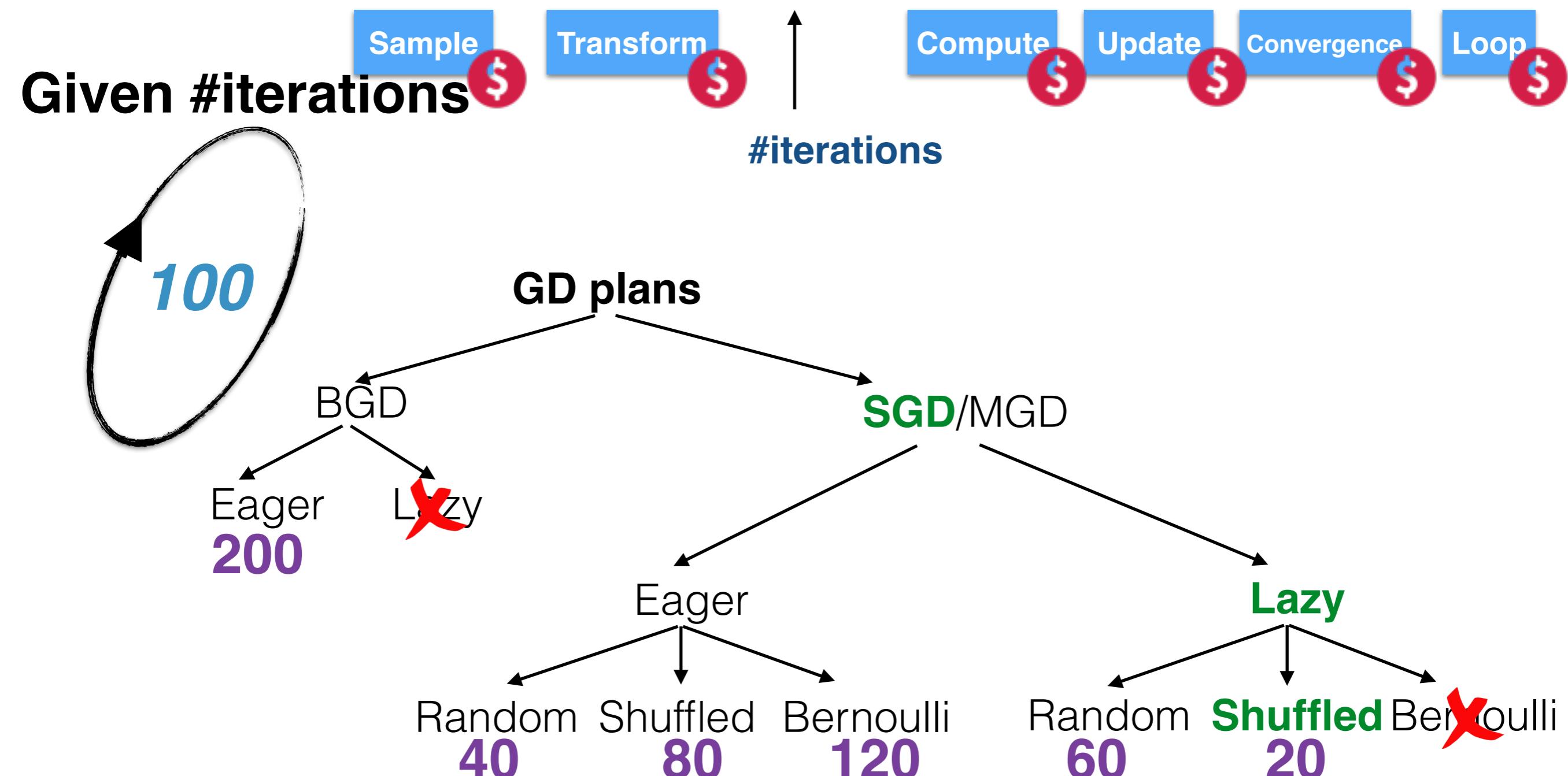
$$C_{BGD}(D) = c_S(D) + c_T(D) + T \times (c_C(D) + c_U + c_D(D) + c_L)$$



# Cost Model

## Cost of BGD plan

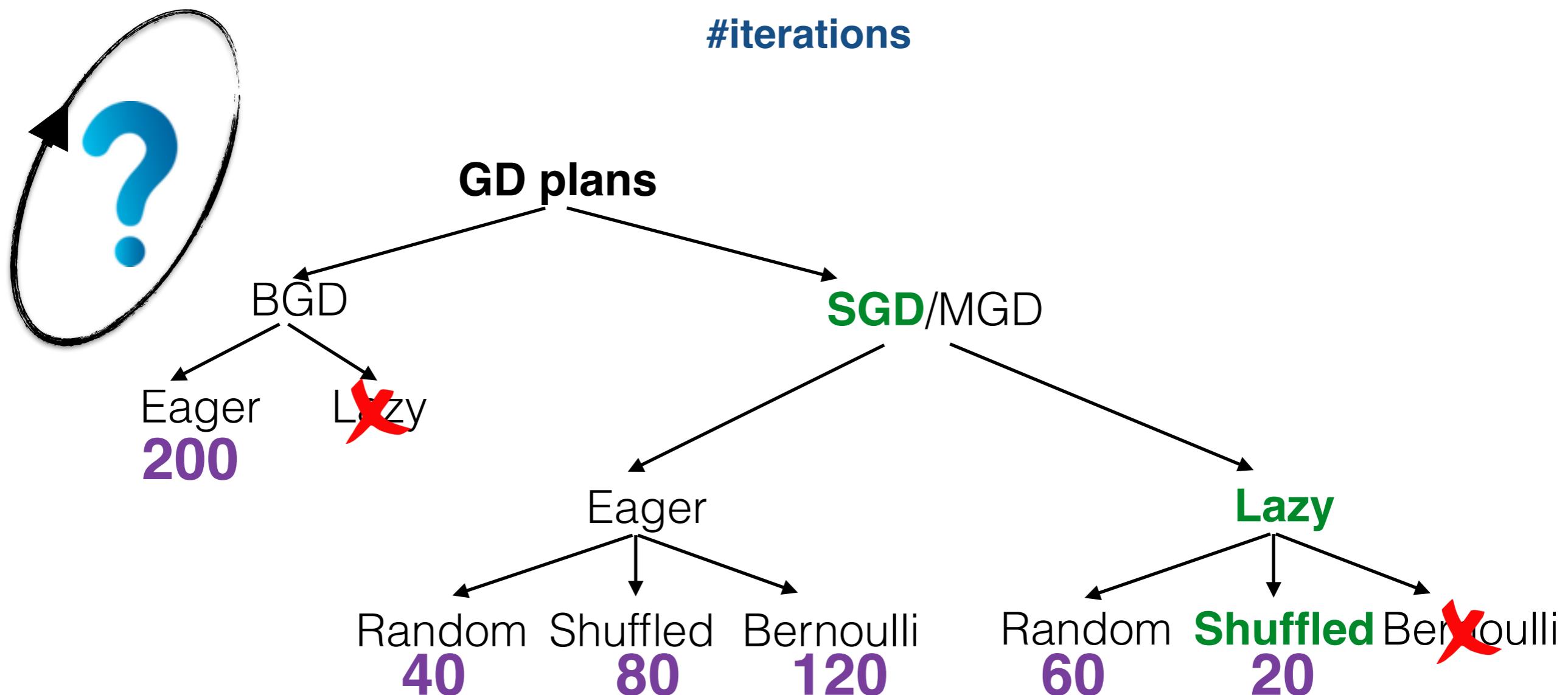
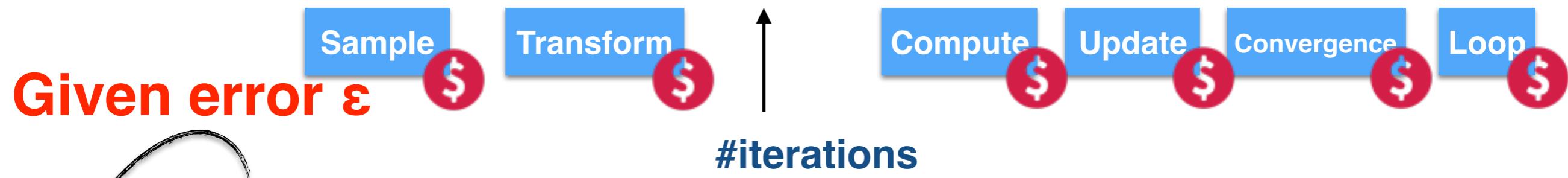
$$C_{BGD}(D) = c_S(D) + c_T(D) + T \times (c_C(D) + c_U + c_D(D) + c_L)$$



# Cost Model

## Cost of BGD plan

$$C_{BGD}(D) = c_S(D) + c_T(D) + T \times (c_C(D) + c_U + c_D(D) + c_L)$$



# Speculative Approach

- 1 *Error sequence follows known distribution*

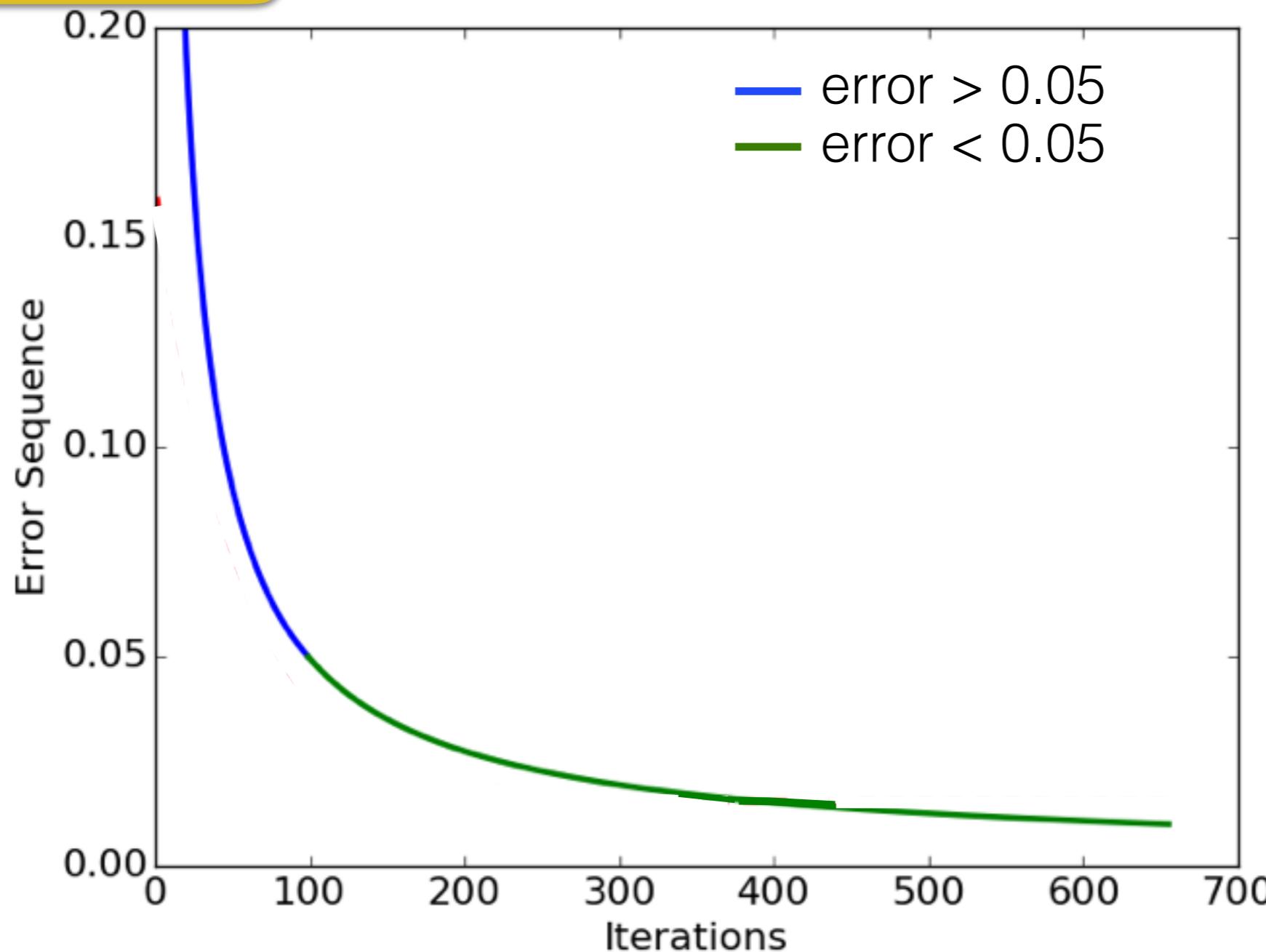


# Speculative Approach

1

*Error sequence follows known distribution*

## Extrapolation

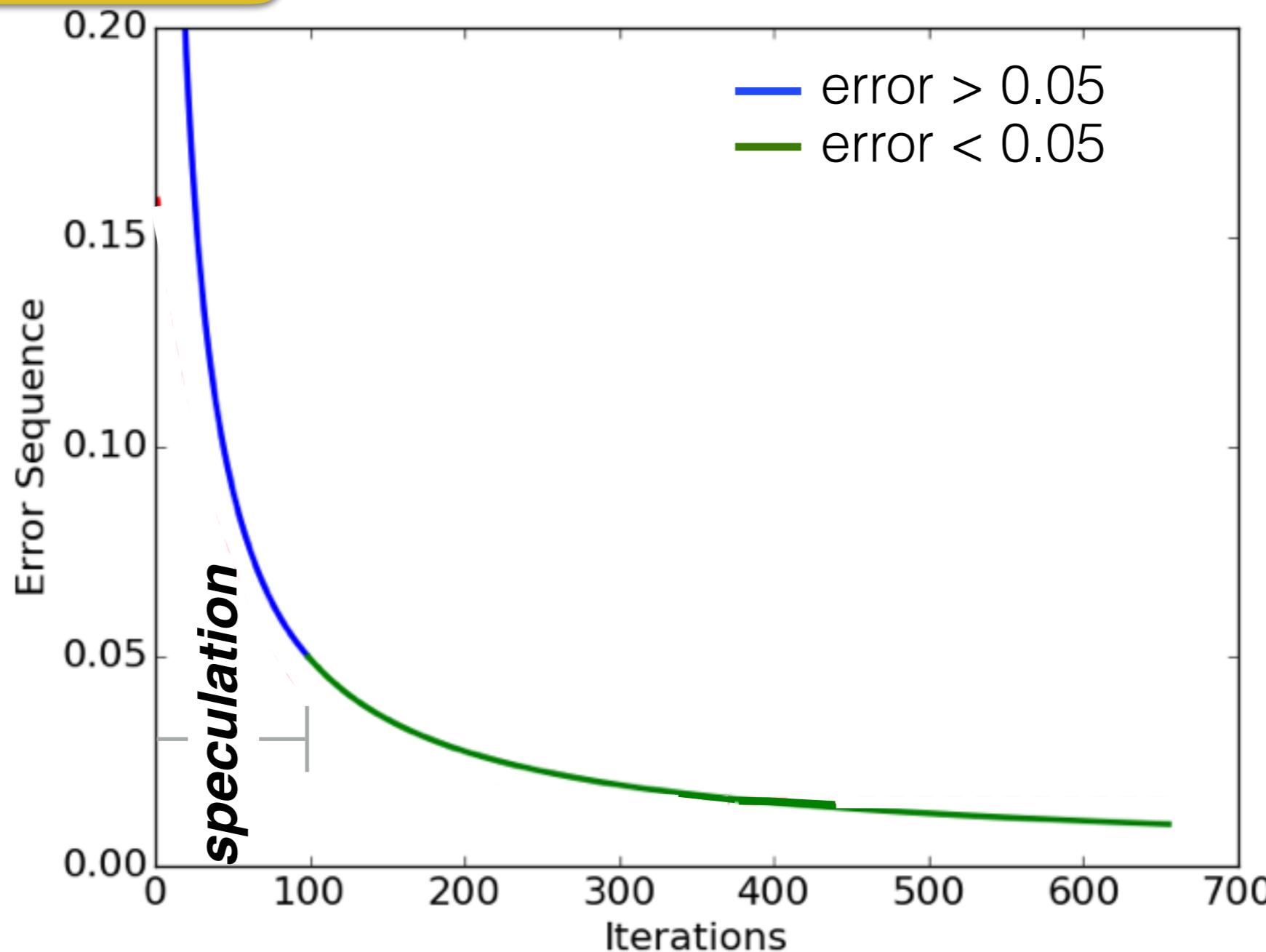


# Speculative Approach

1

*Error sequence follows known distribution*

## Extrapolation

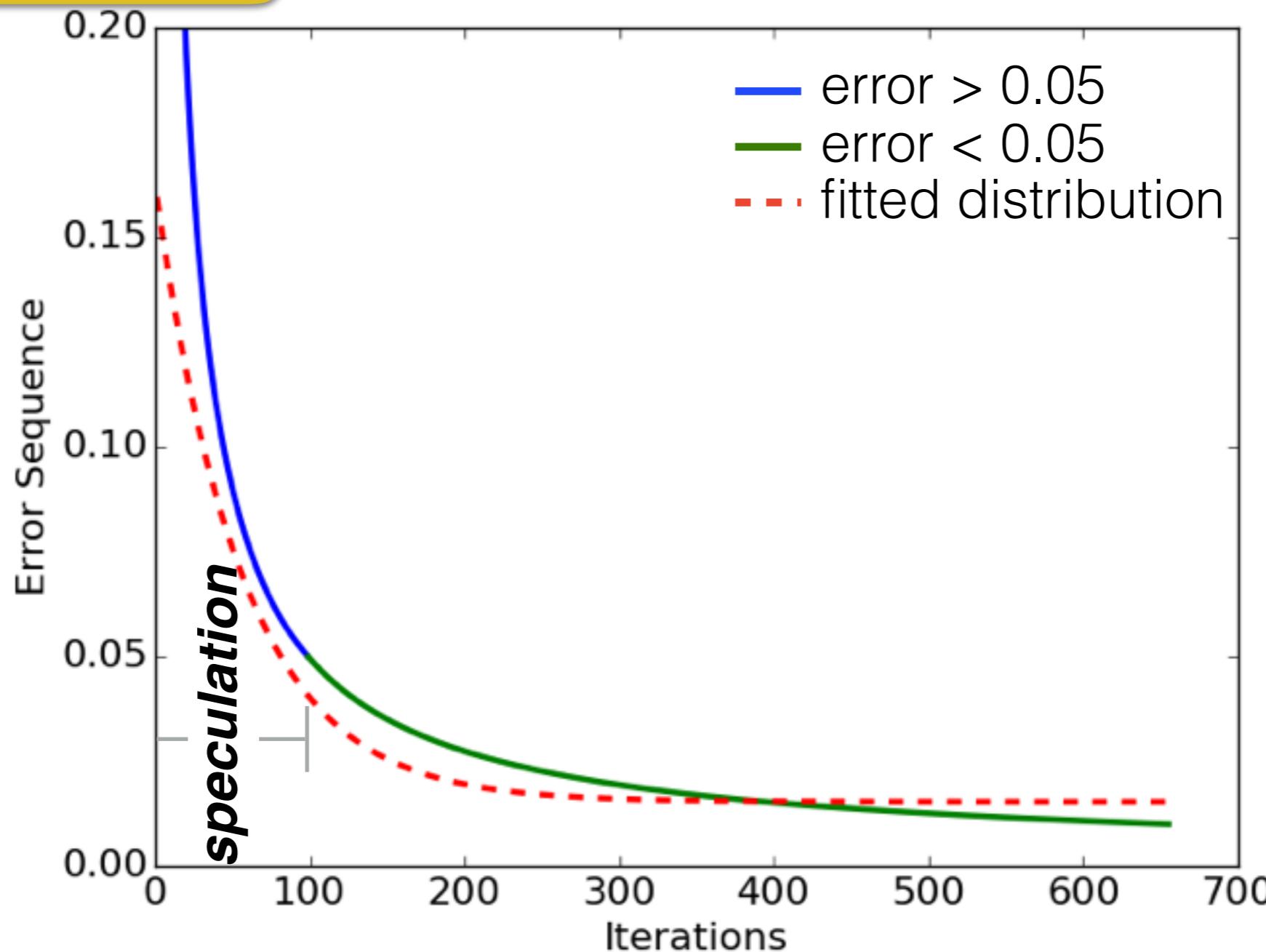


# Speculative Approach

1

*Error sequence follows known distribution*

## Extrapolation

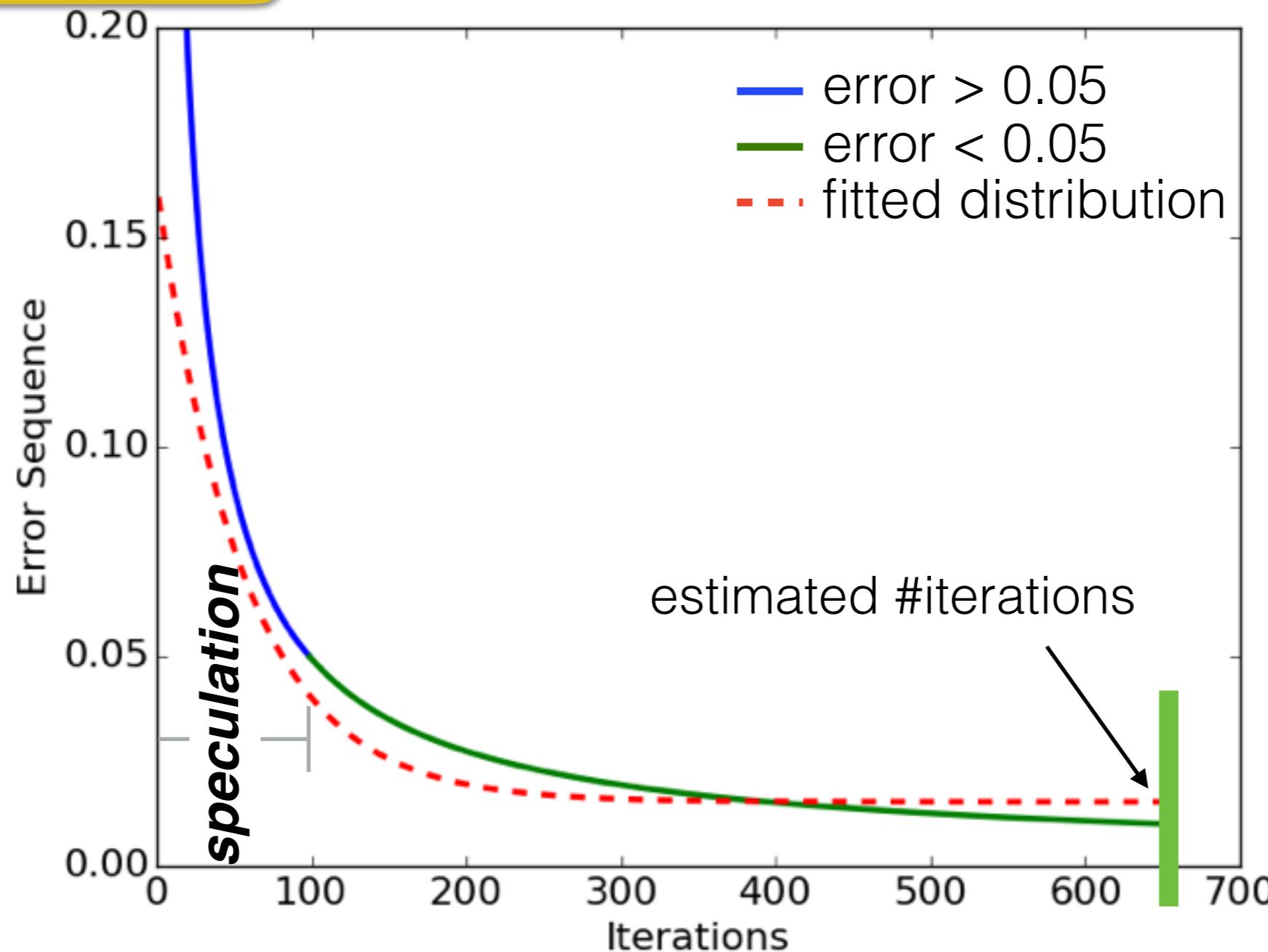


# Speculative Approach

1

*Error sequence follows known distribution*

## Extrapolation



# Speculative Approach

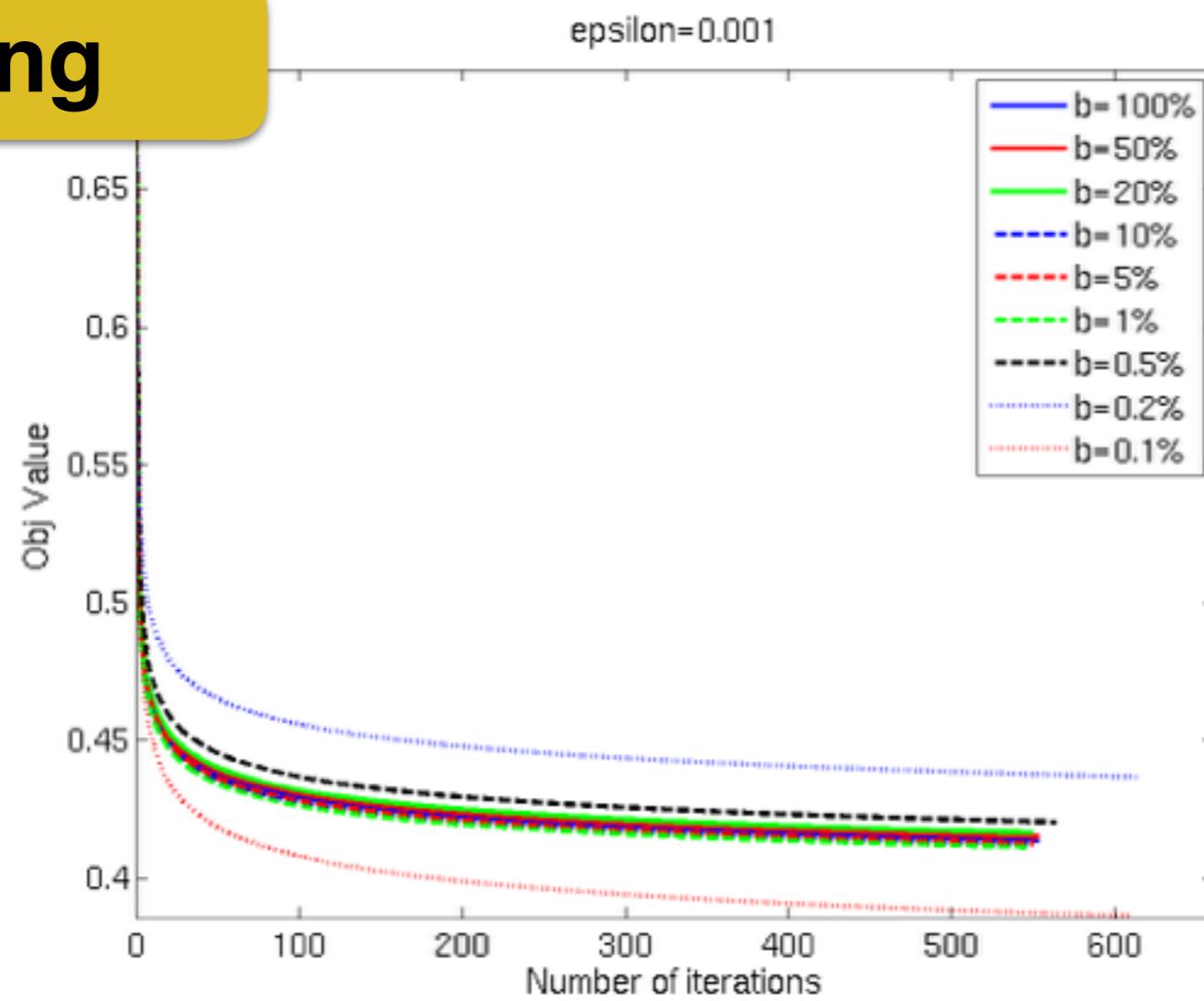
- 1 *Error sequence follows known distribution*
- 2 *Shape of the error sequence over a sample  $D' \ll D$*   
*Shape of the error sequence over  $D$*   $\underset{\sim}{\sim}$



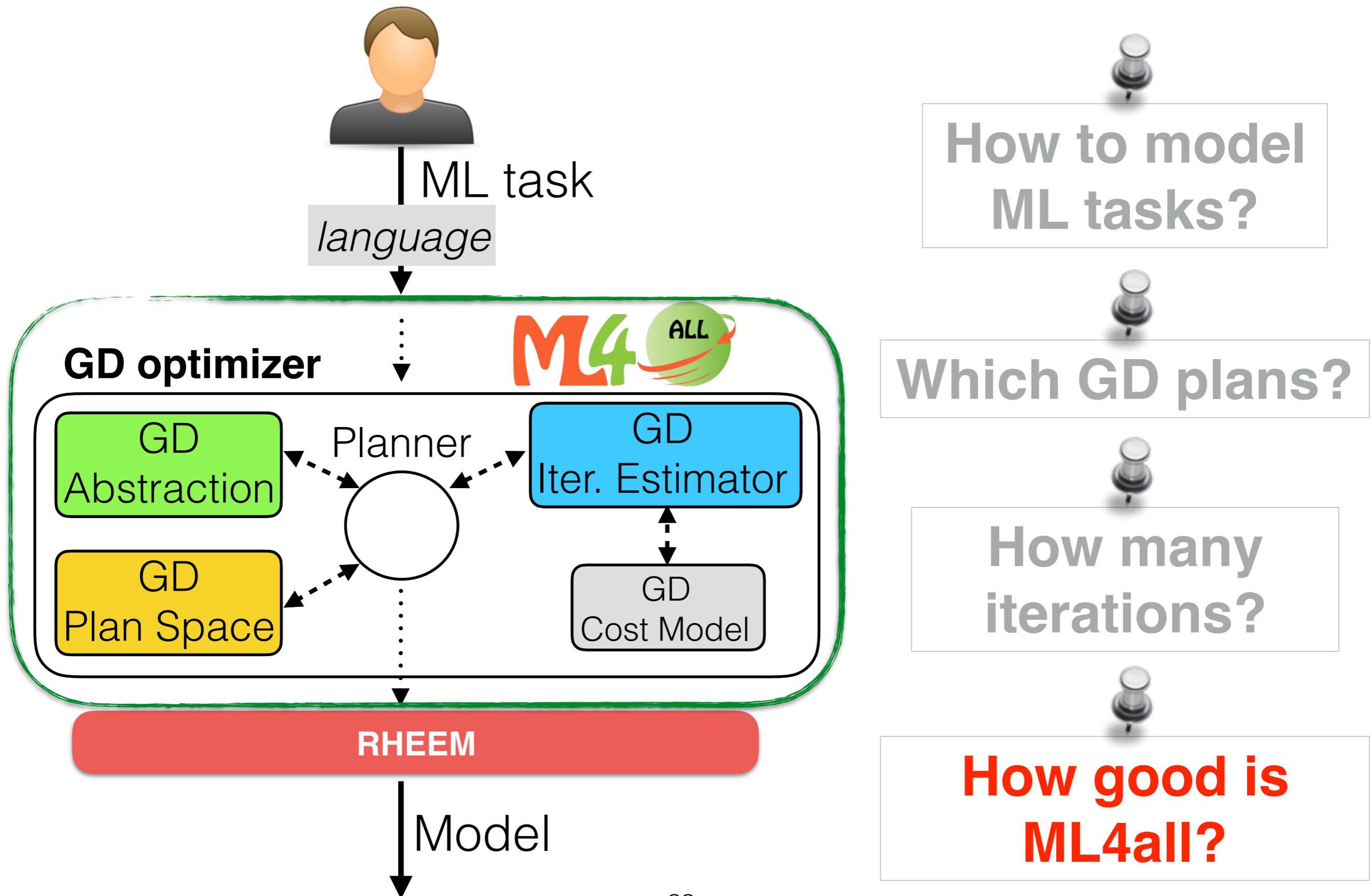
# Speculative Approach

- 1 *Error sequence follows known distribution*
- 2 *Shape of the error sequence over a sample  $D' \ll D$*   
 $\sim$   
*Shape of the error sequence over  $D$*

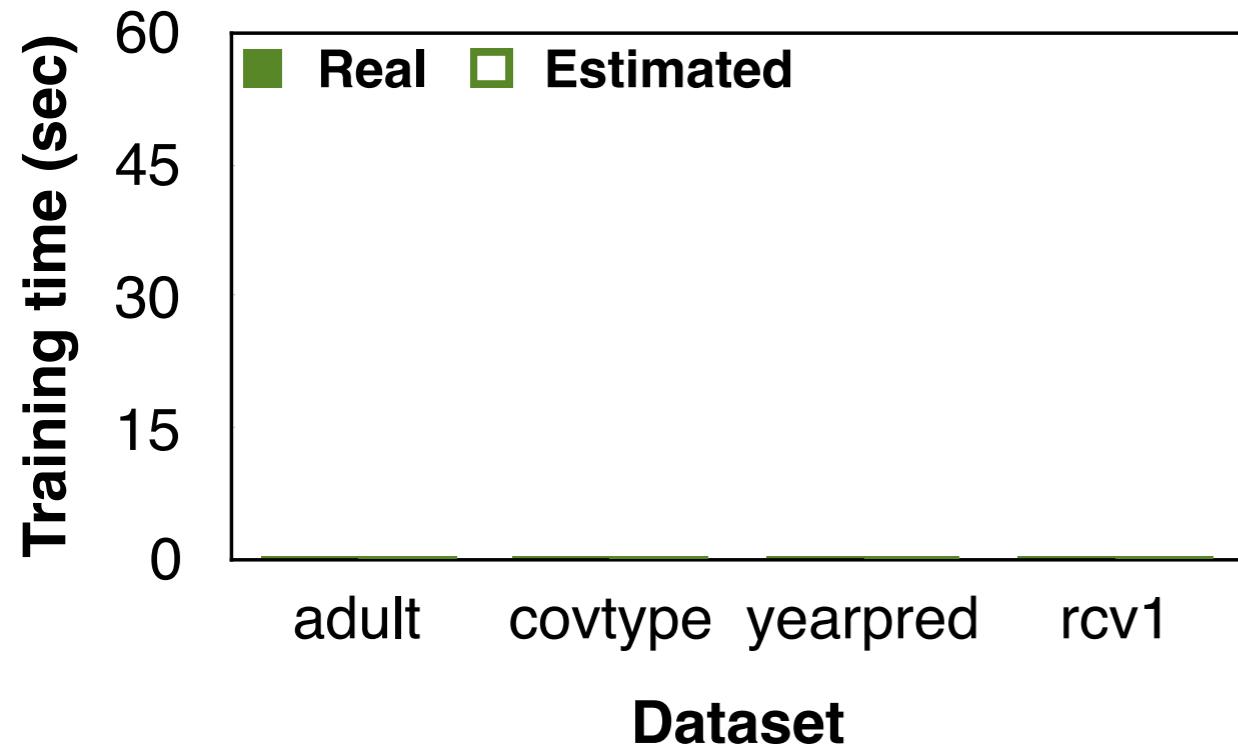
## Sampling



# Towards a DB-like System

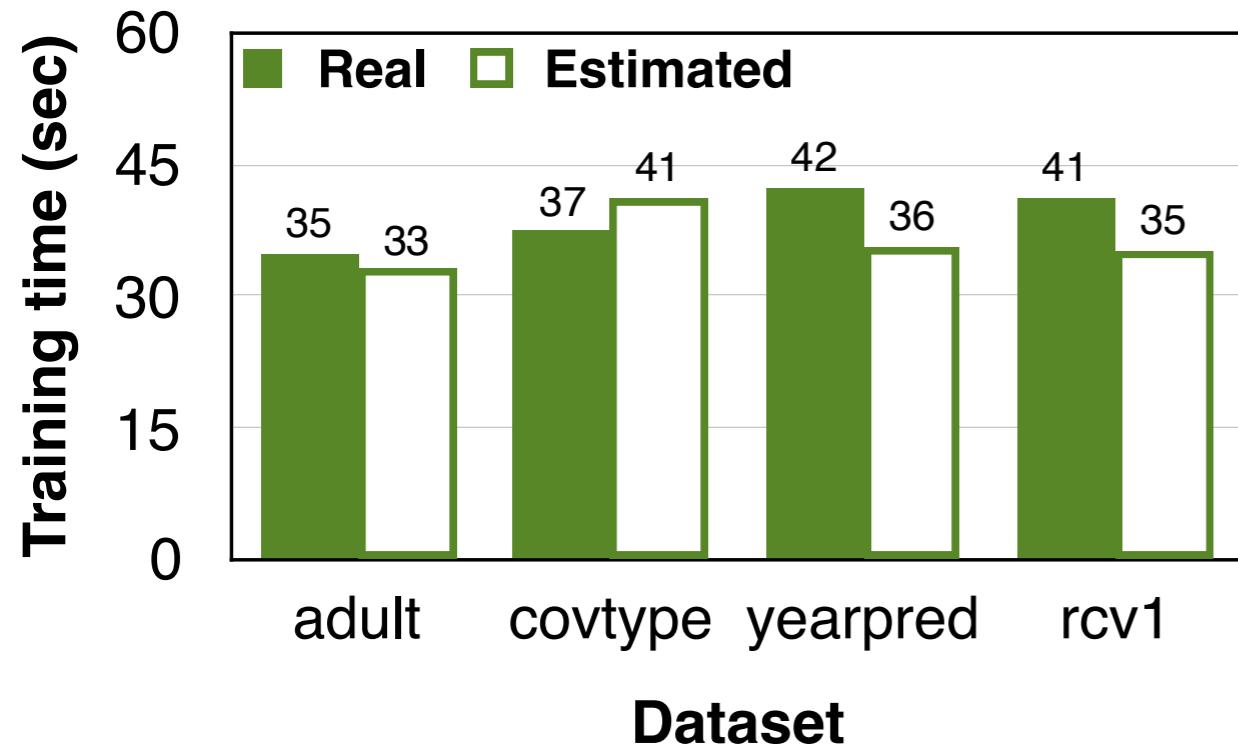


# Time Estimates



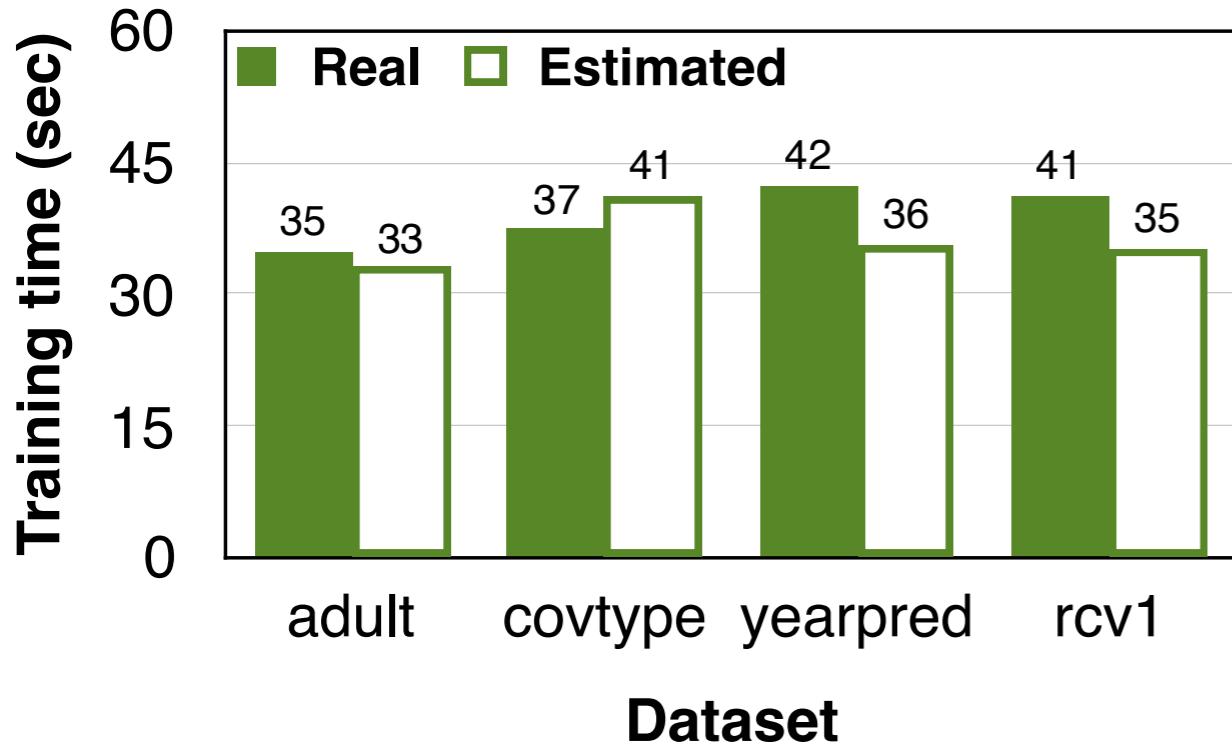
(a) Run of 1,000 iterations

# Time Estimates



(a) Run of 1,000 iterations

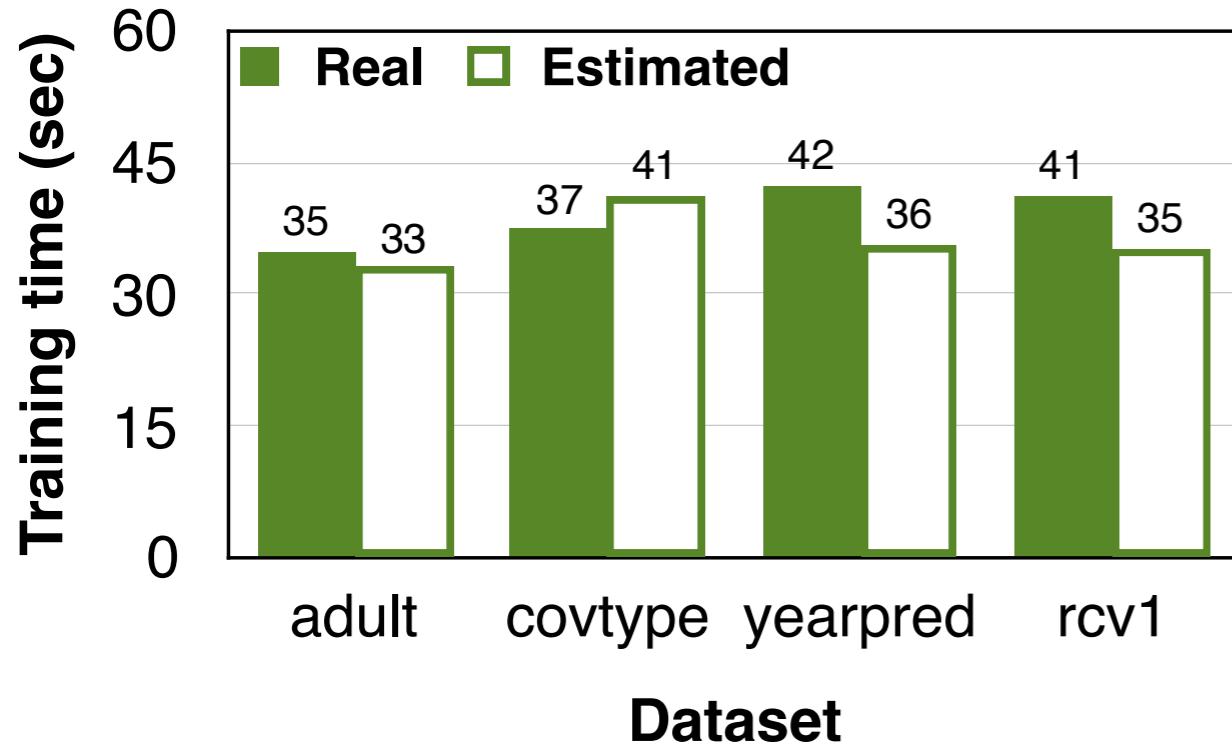
# Time Estimates



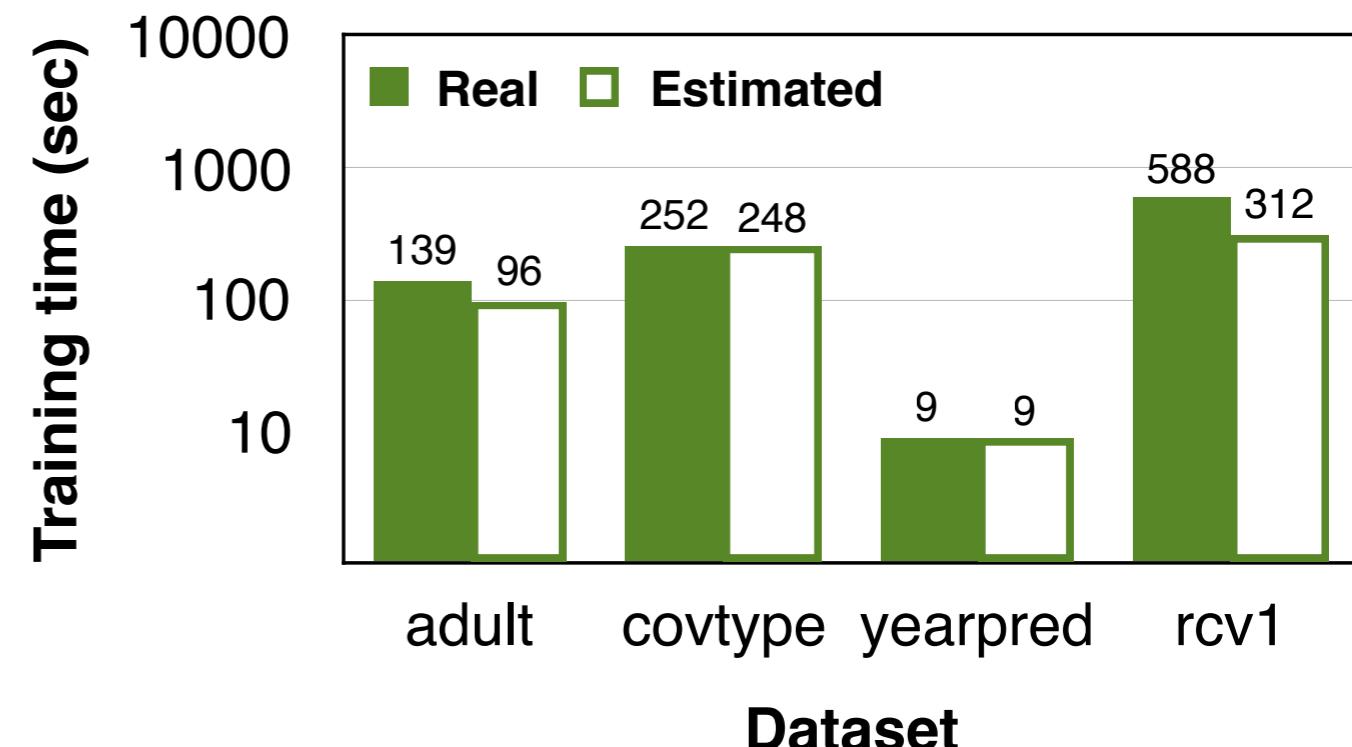
(a) Run of 1,000 iterations

Accurate cost model

# Time Estimates



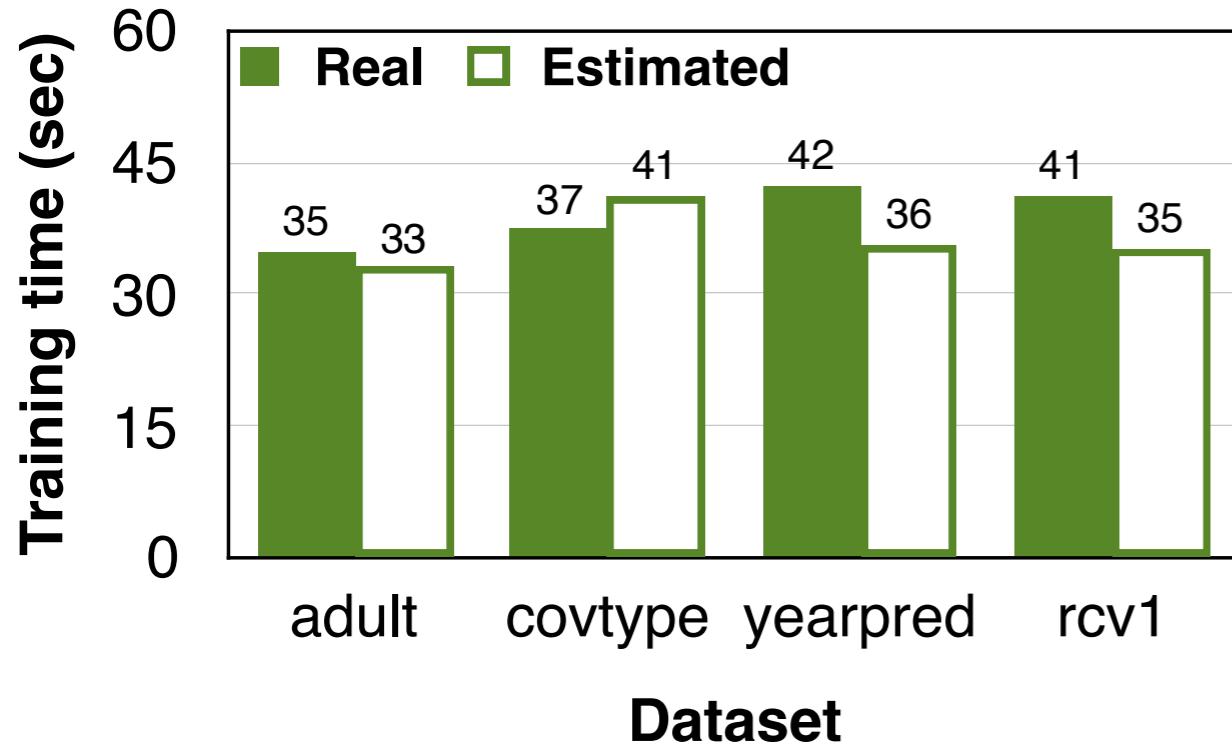
(a) Run of 1,000 iterations



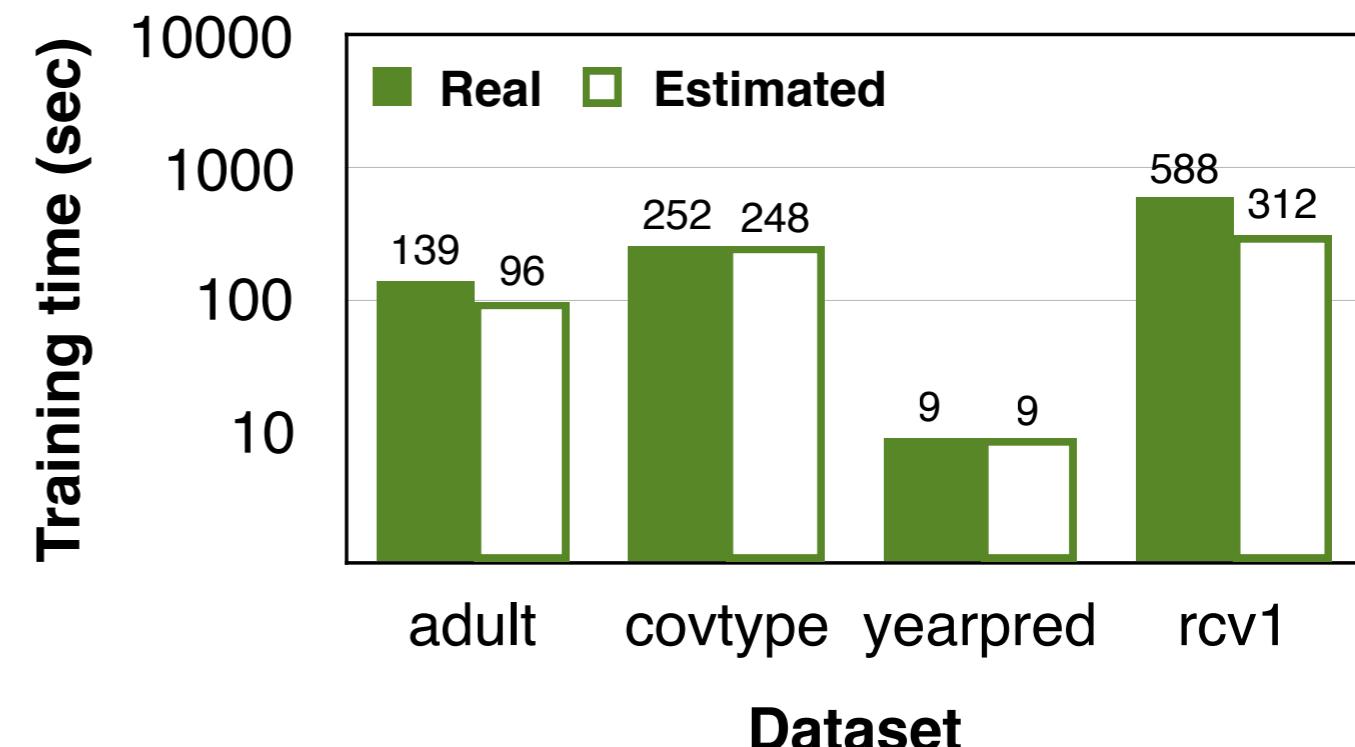
(b) Run to convergence

Accurate cost model

# Time Estimates



(a) Run of 1,000 iterations

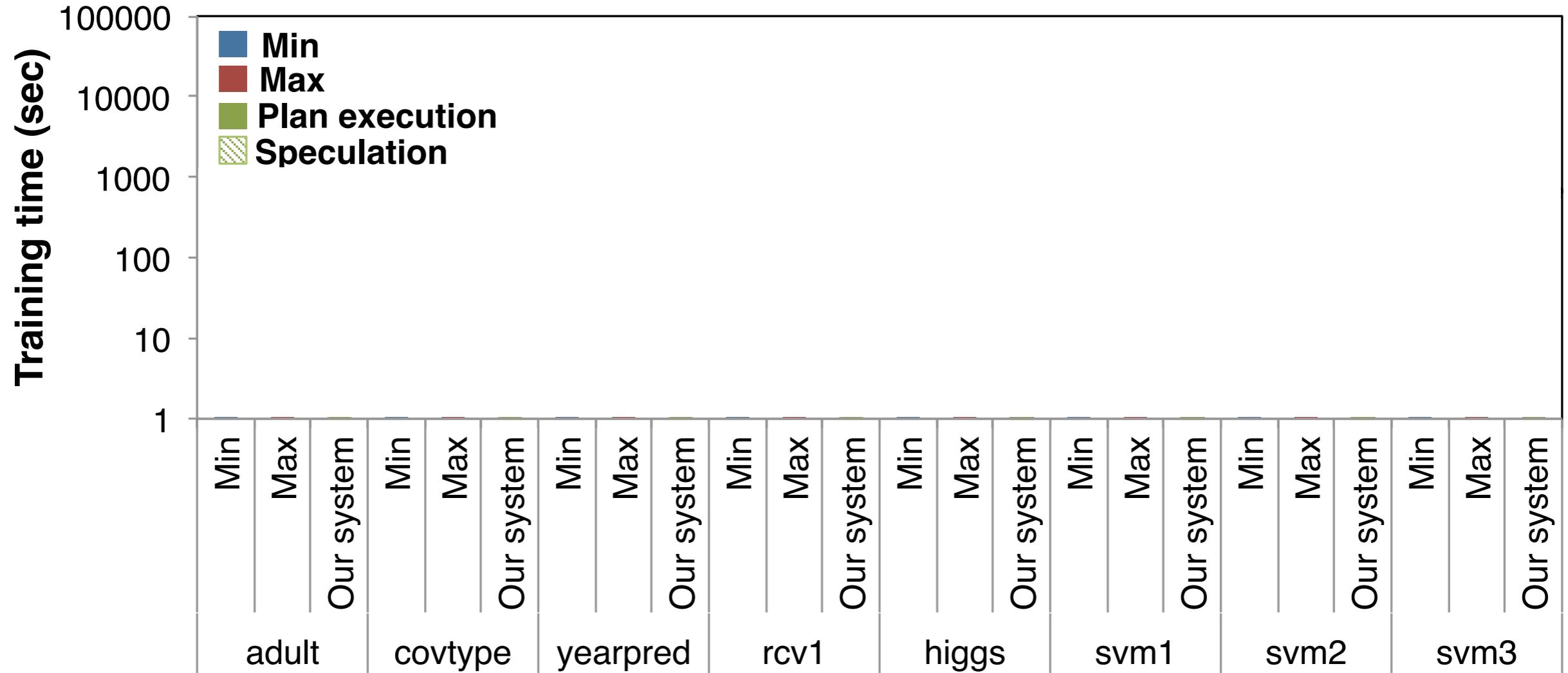


(b) Run to convergence

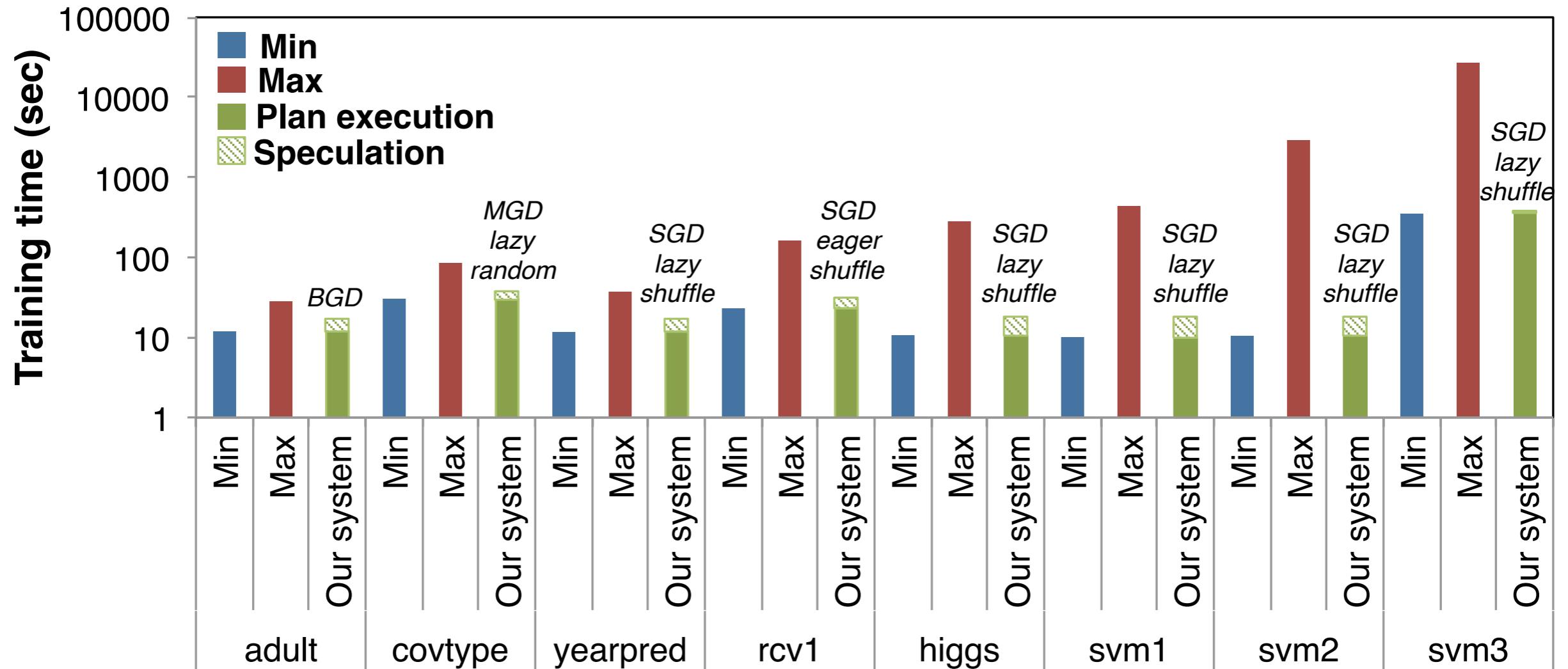
Accurate cost model

Accurate Iter. Estimator

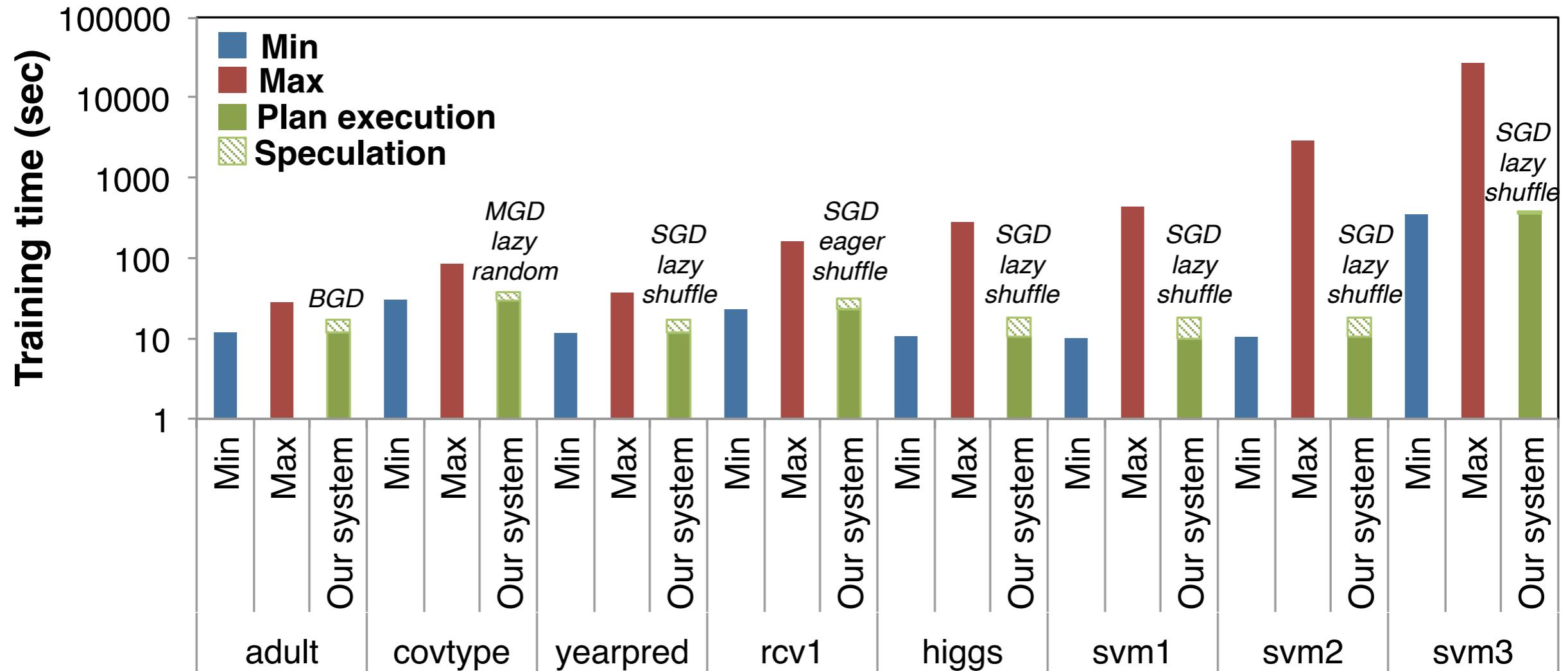
# GD Plan Choice



# GD Plan Choice



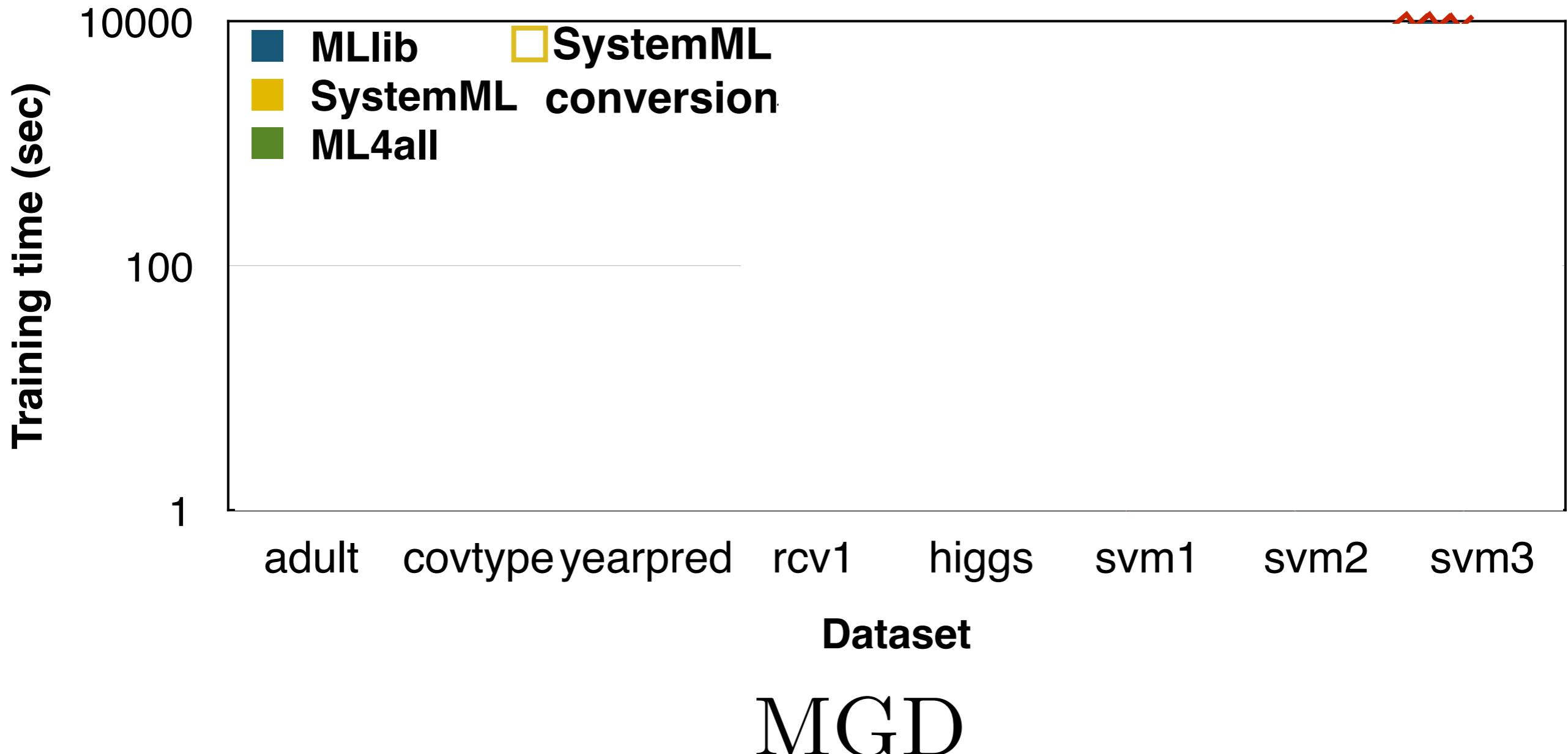
# GD Plan Choice



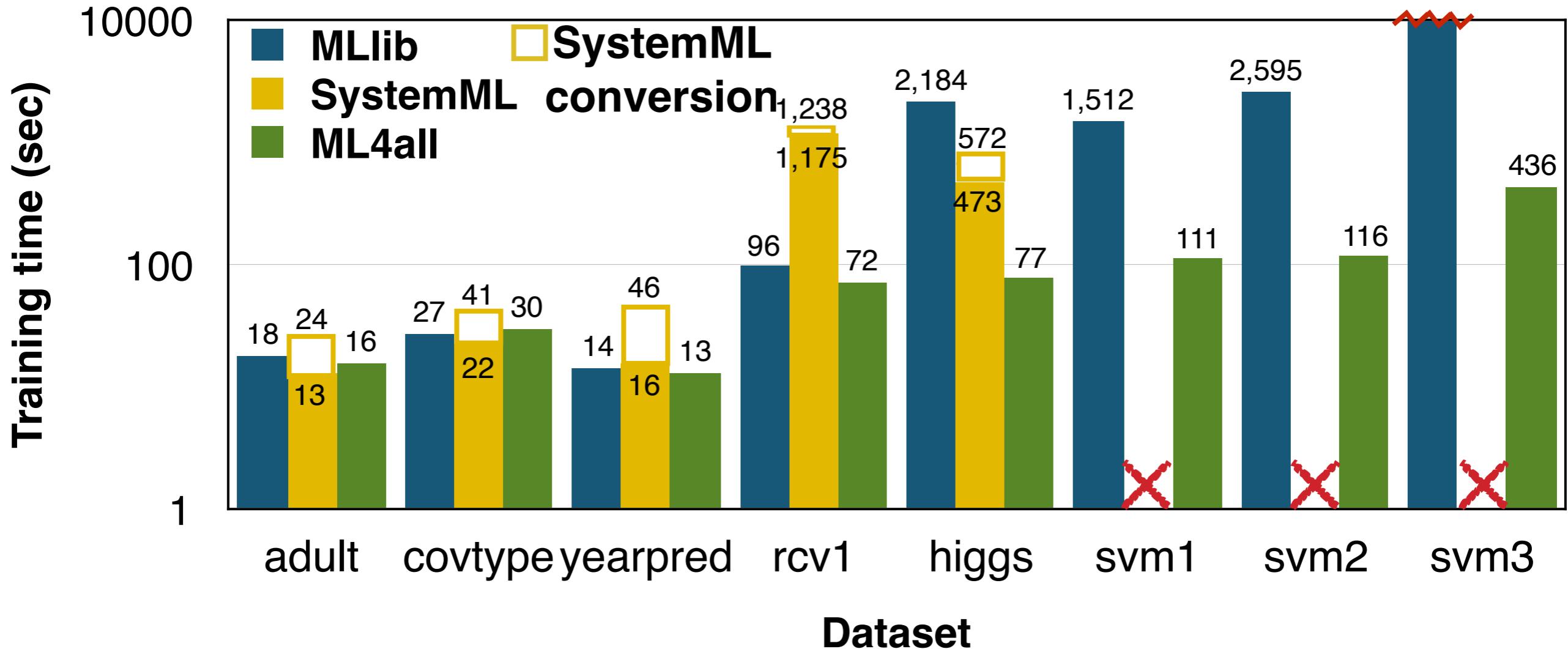
Always the best plan

Negligible overhead

# Efficiency

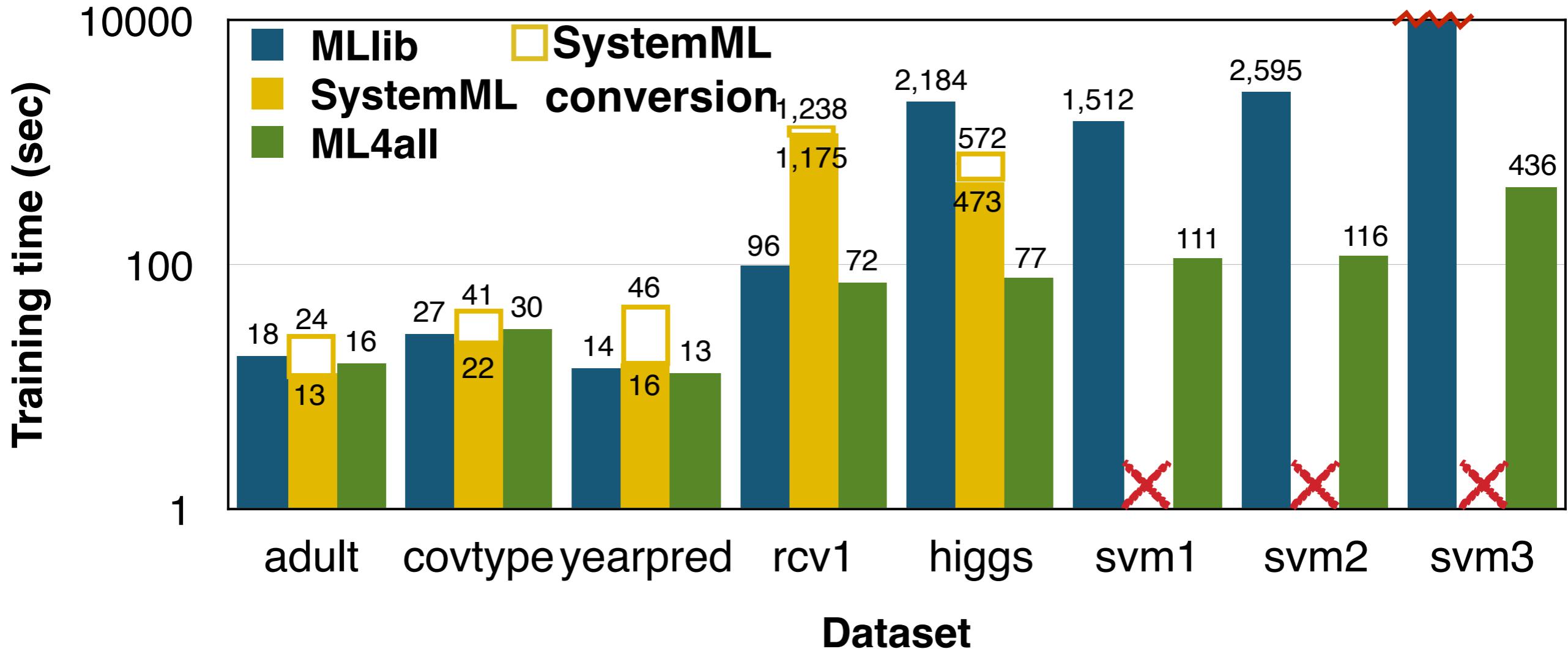


# Efficiency



MGD

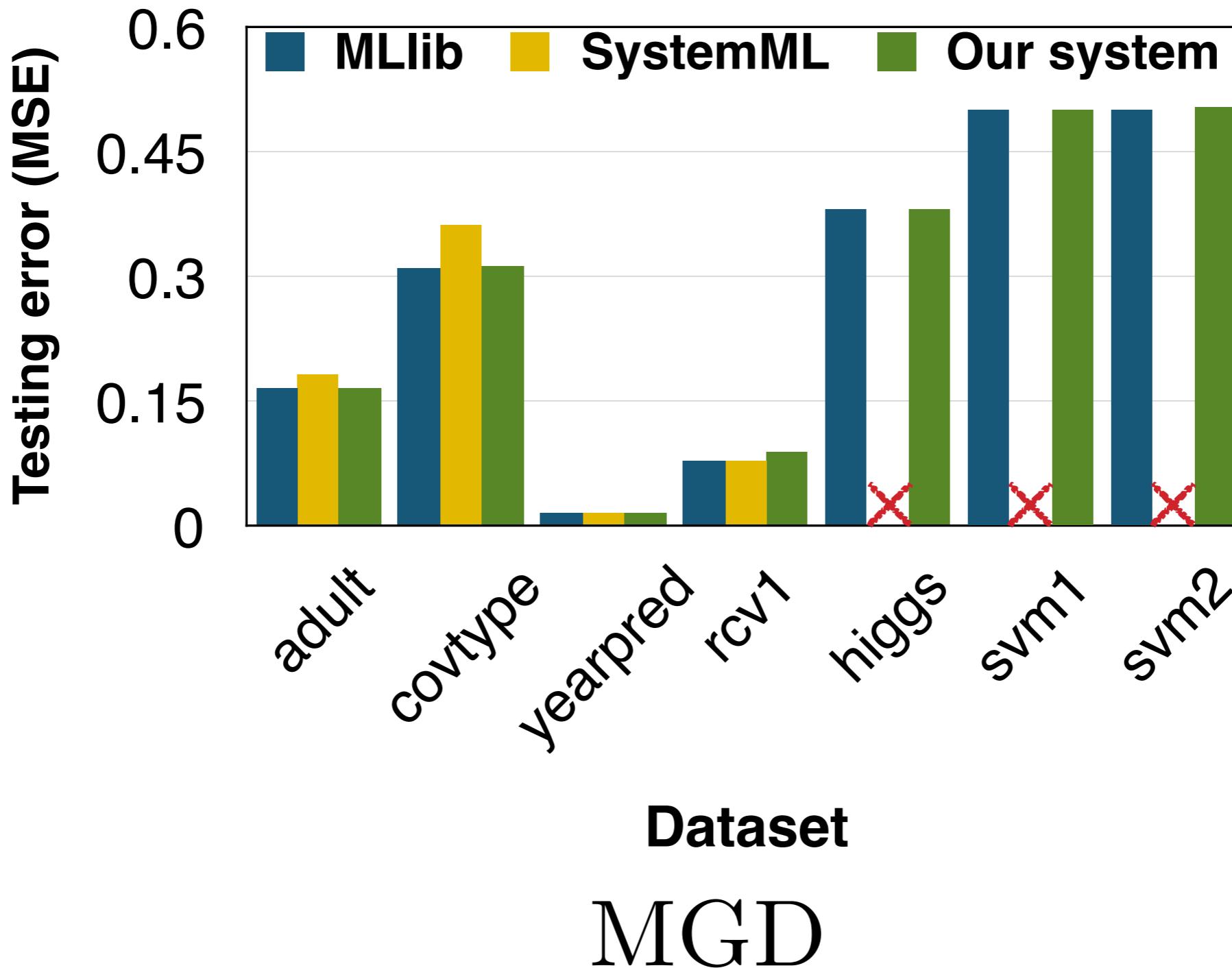
# Efficiency



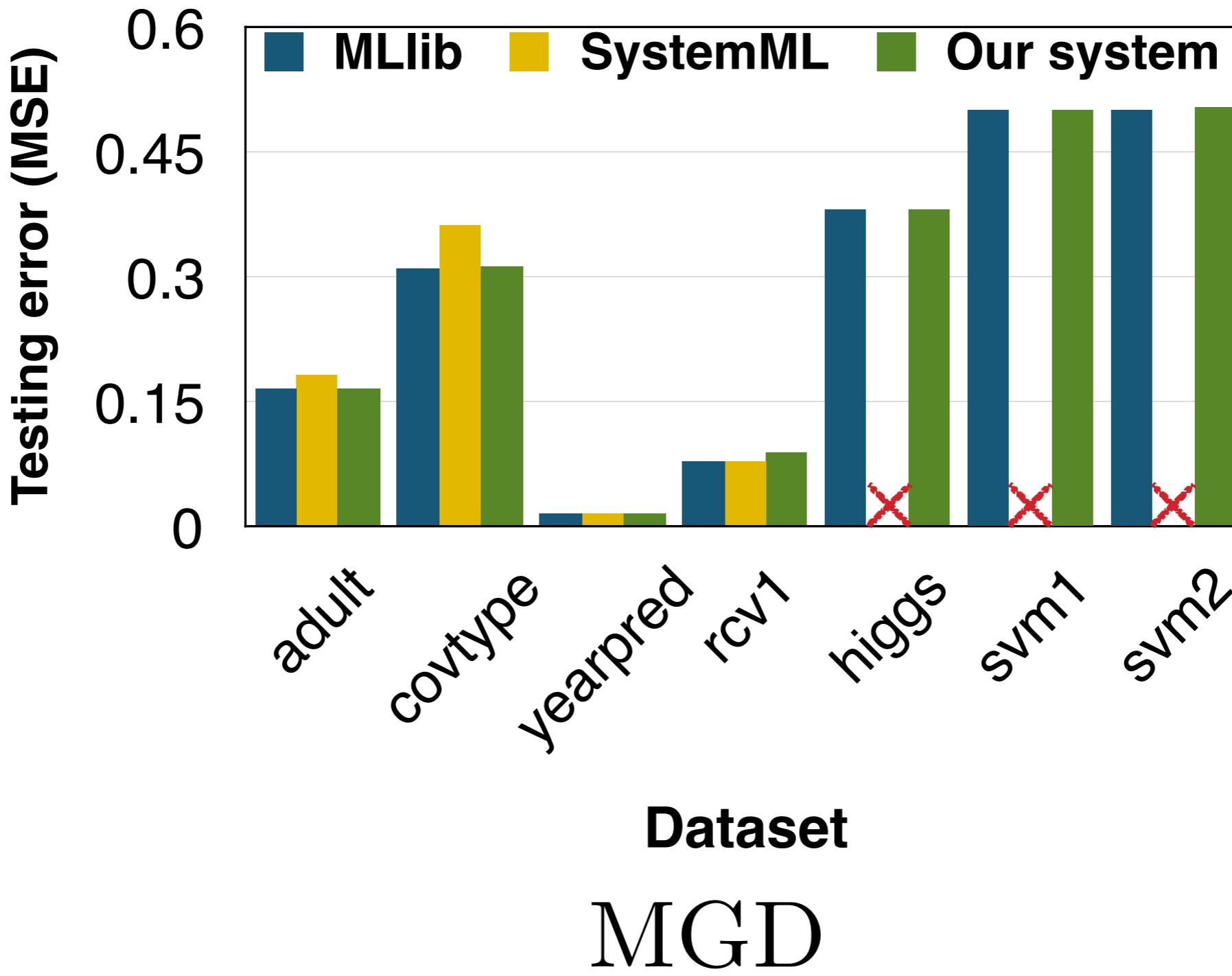
MGD

More than 1 order of magnitude faster!

# Effectiveness

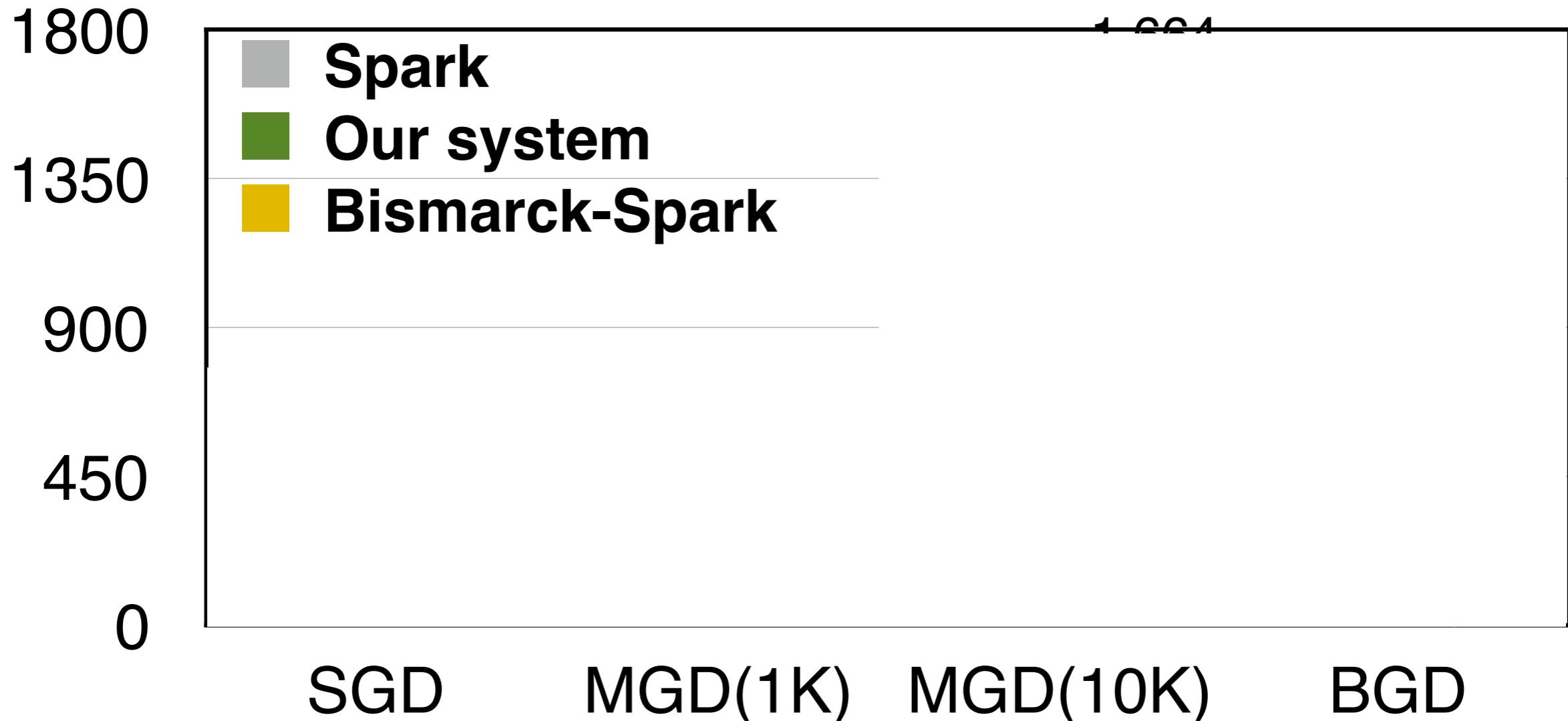


# Effectiveness



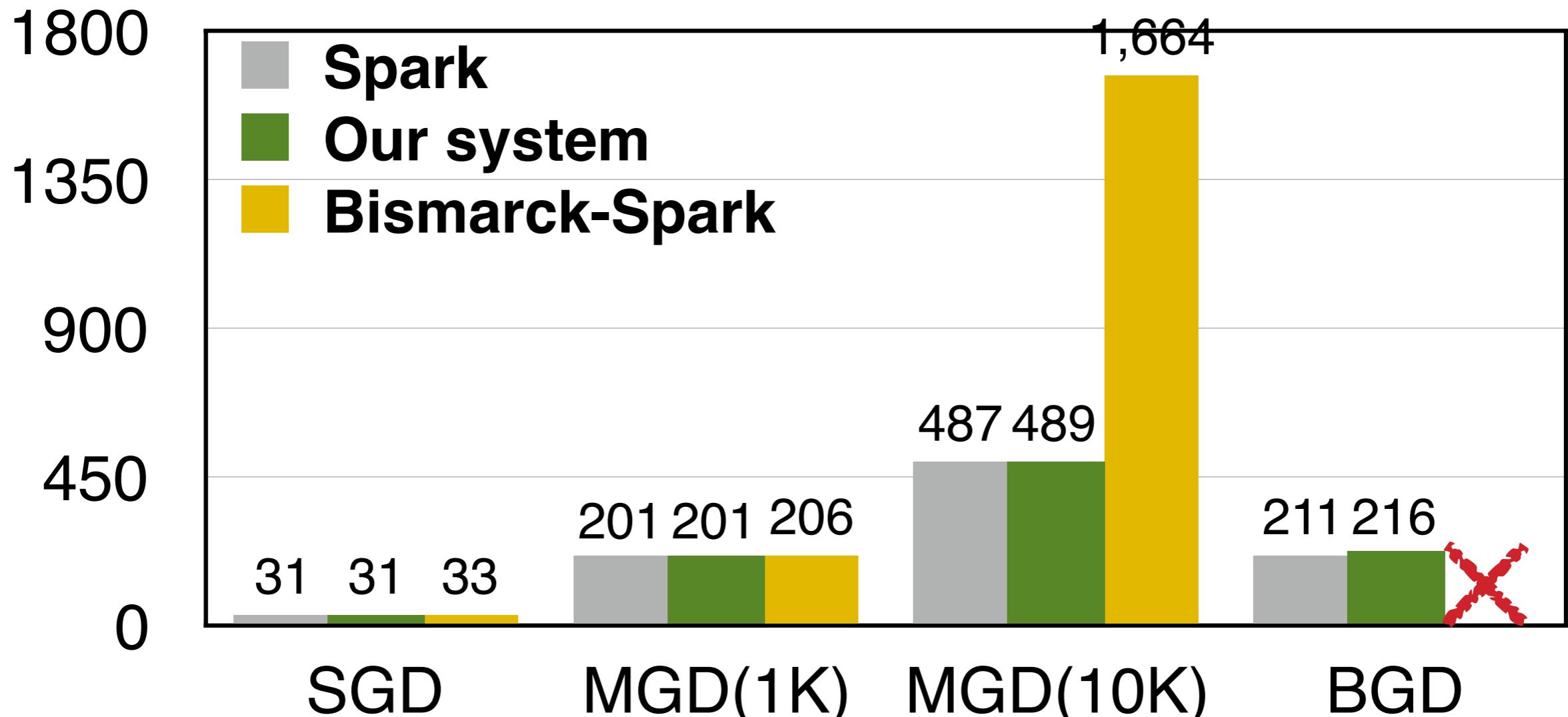
No Sacrifice on the MSE

# Abstraction Benefit



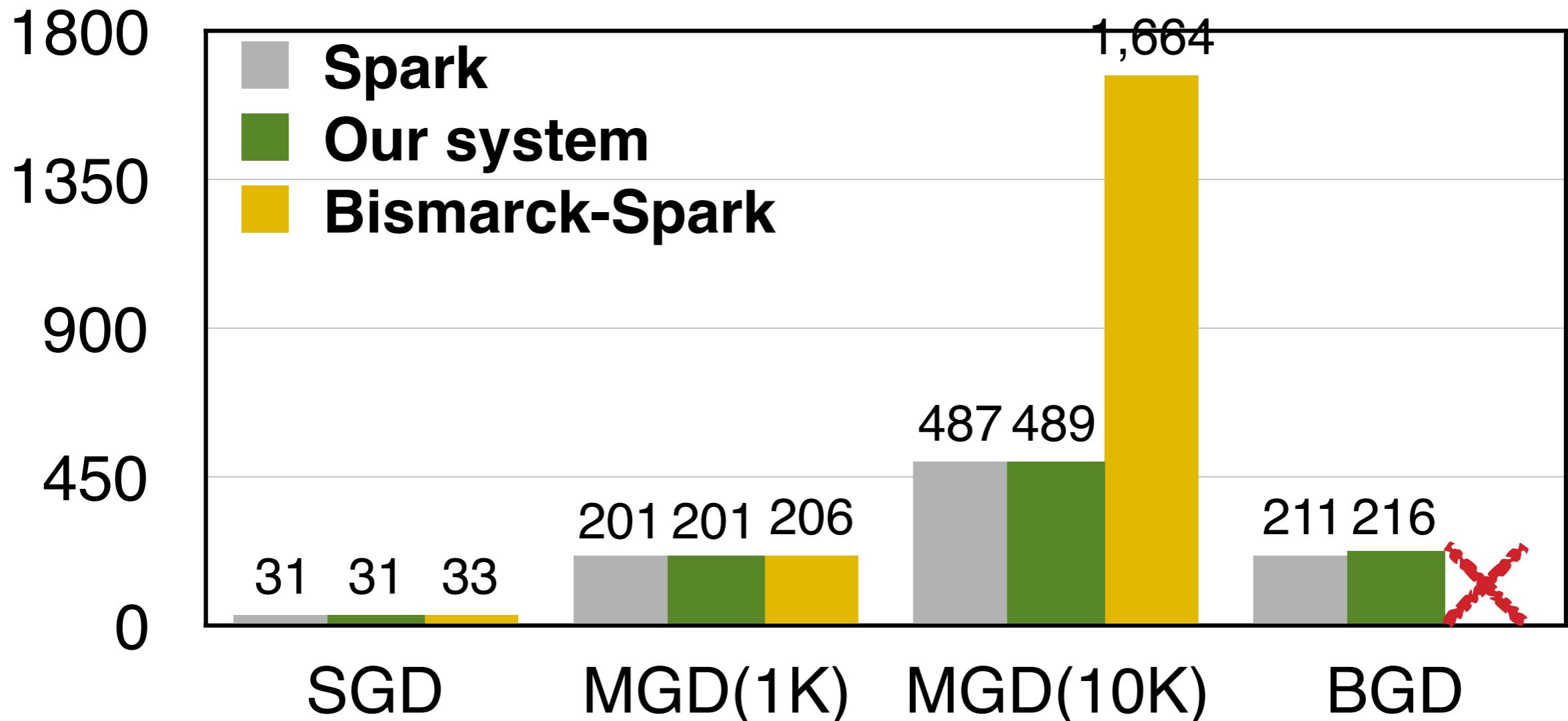
svm1 dataset

# Abstraction Benefit



svm1 dataset

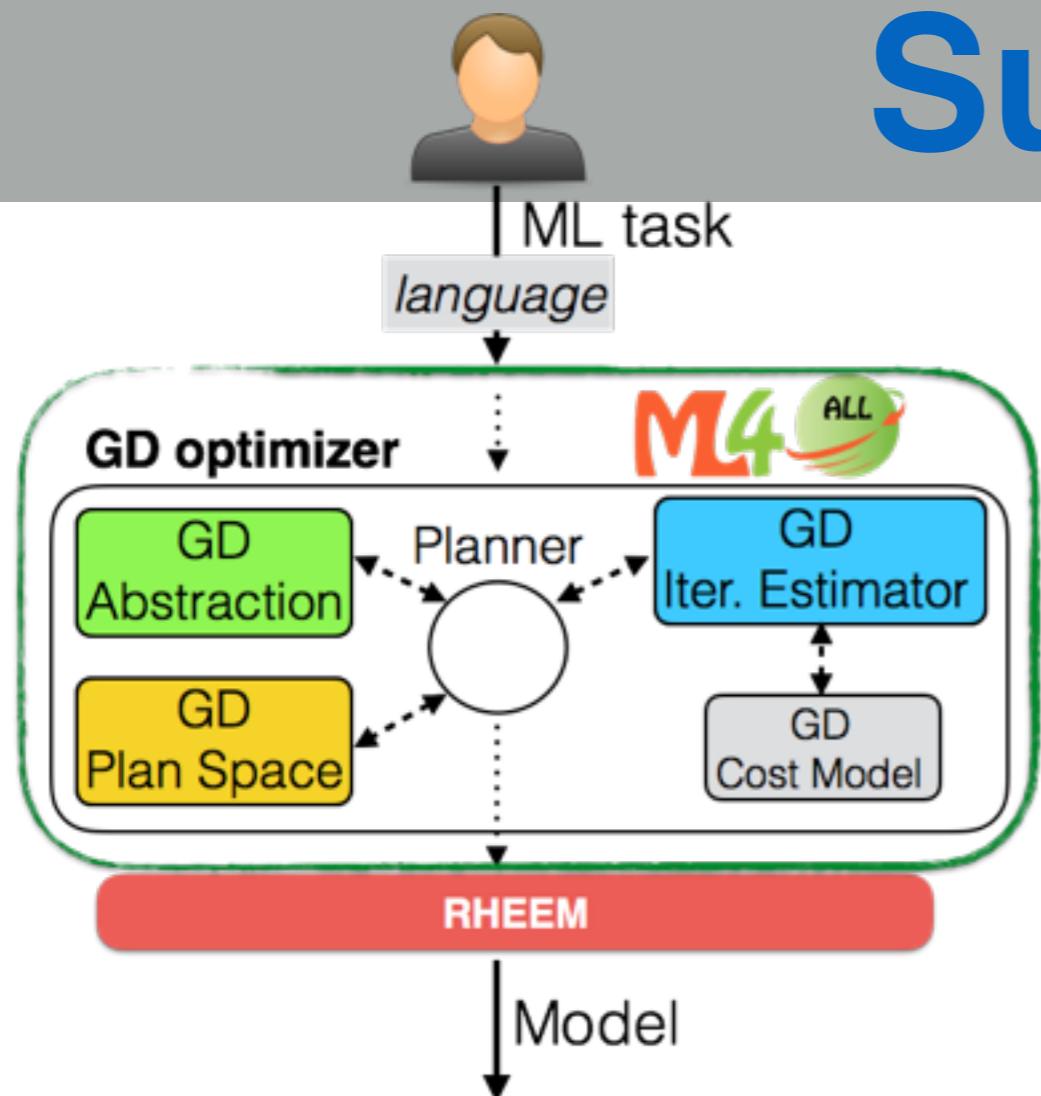
# Abstraction Benefit



svm1 dataset

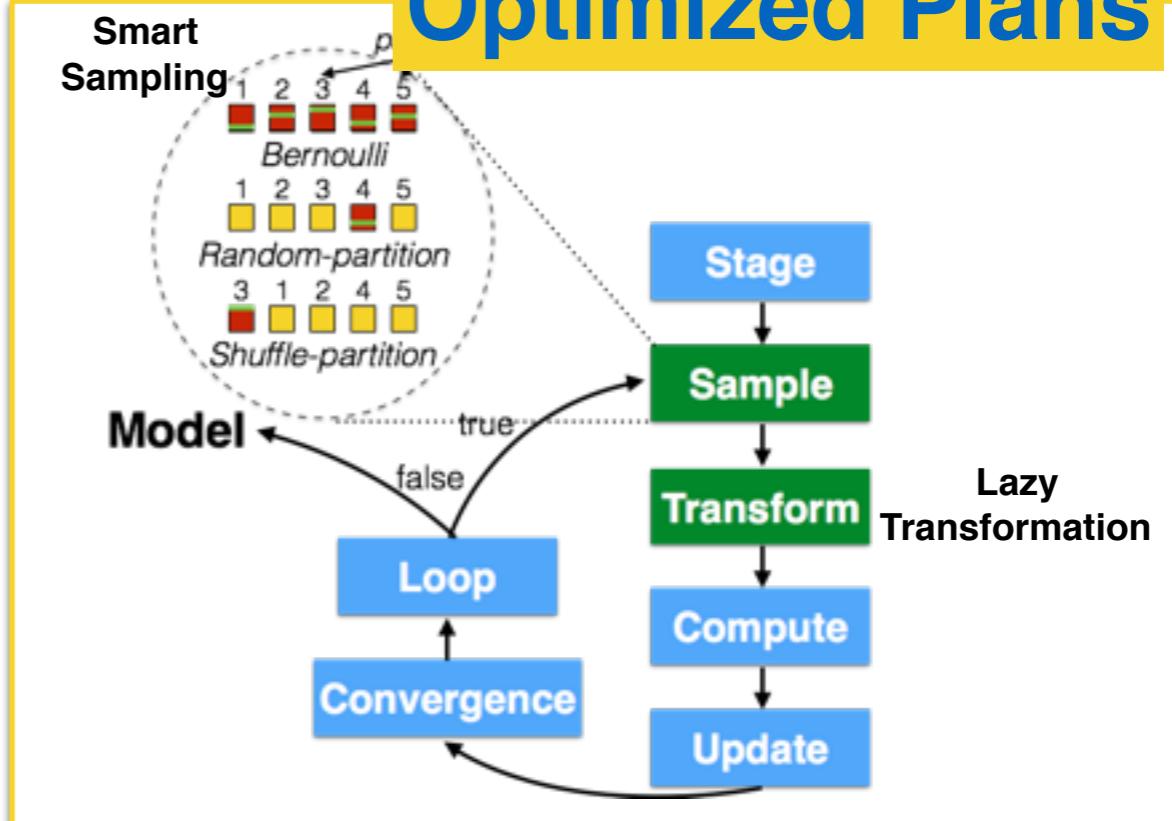
As good as the pure Spark code!

# Summary

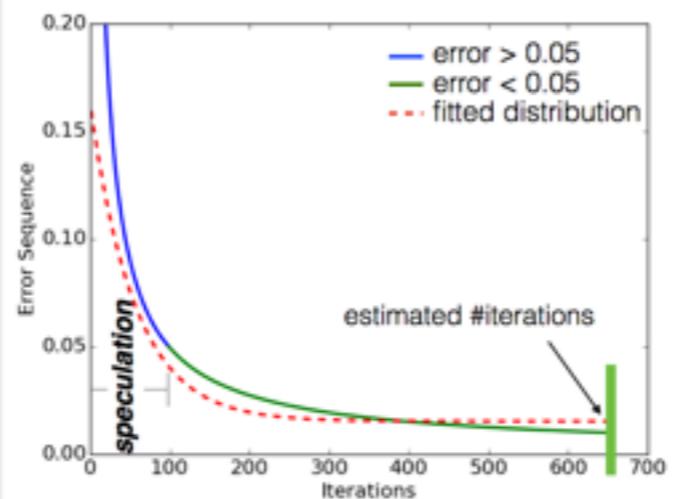


**M4 ALL**

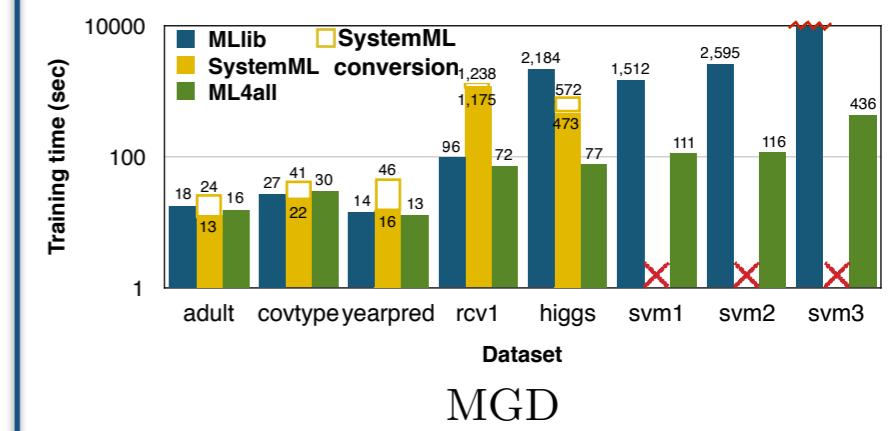
## Optimized Plans



## Speculative Estimator



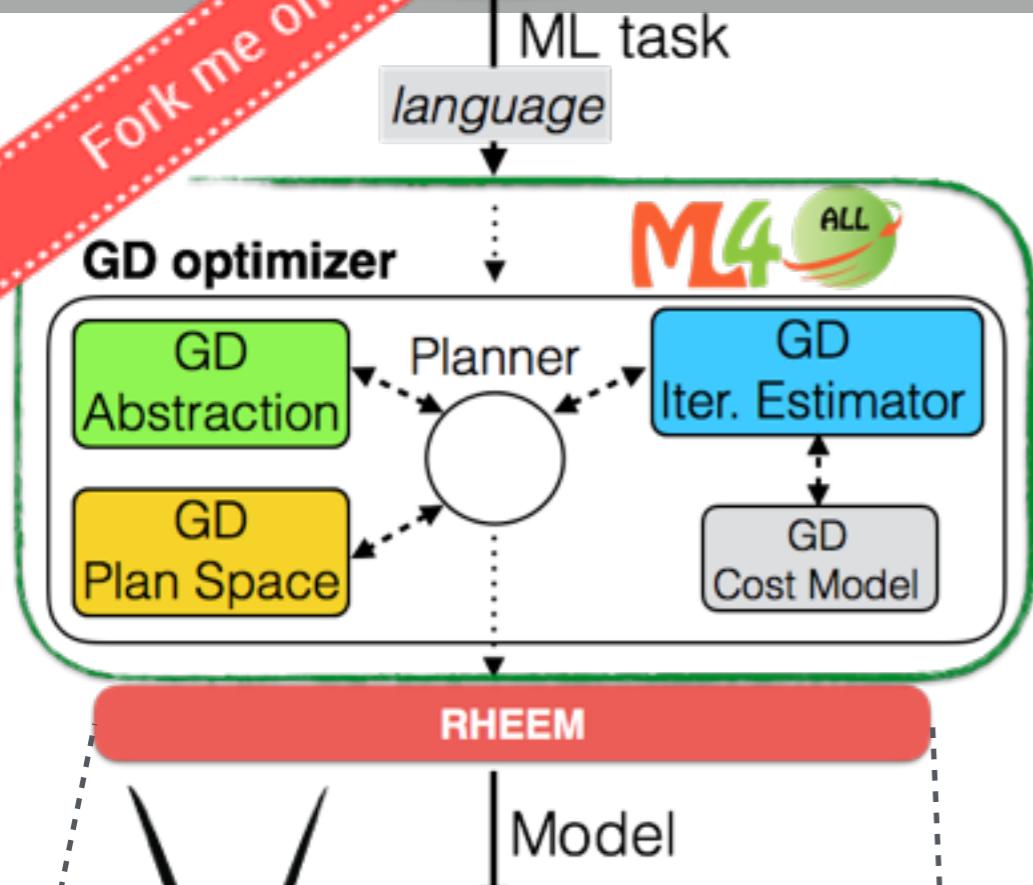
## High performance



MGD

# Summary

Fork me on GitHub

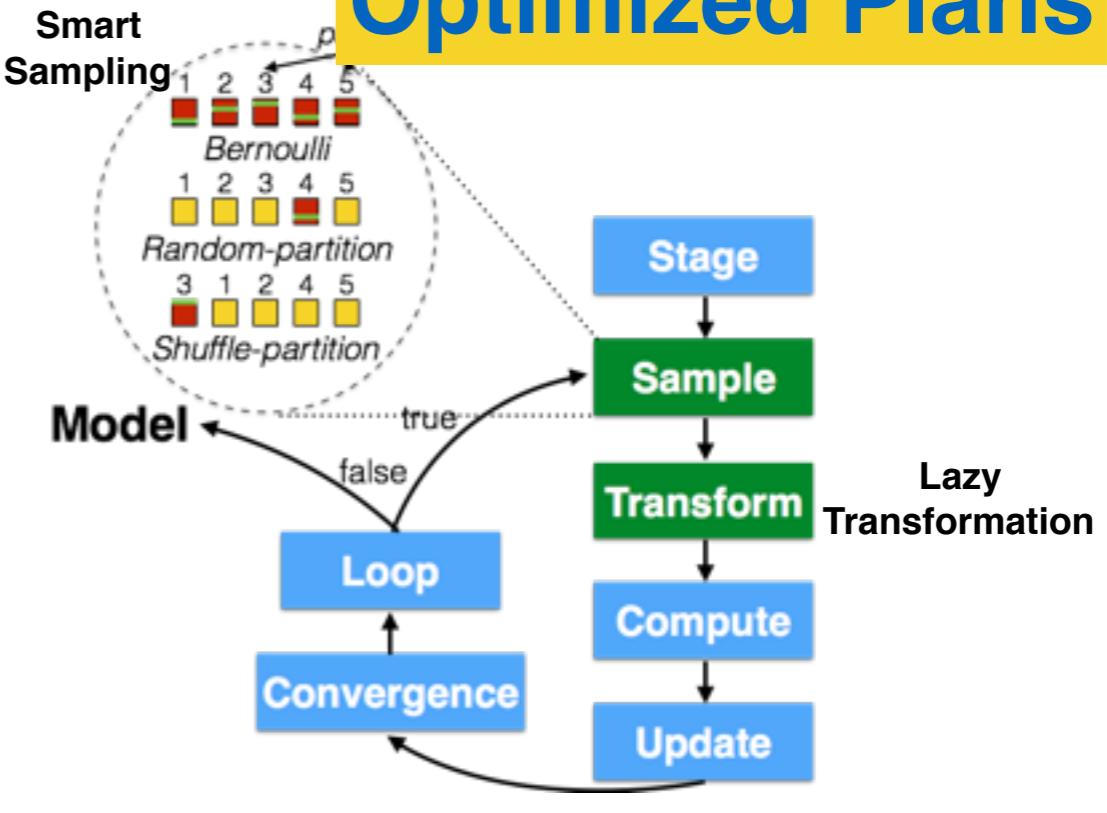


A Cross-Platform System

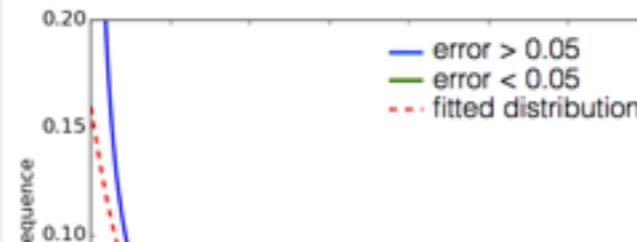
Visit our web page:  
<http://da.qcri.org/rheem>

Join us at the  
*Spark Summit 2017*

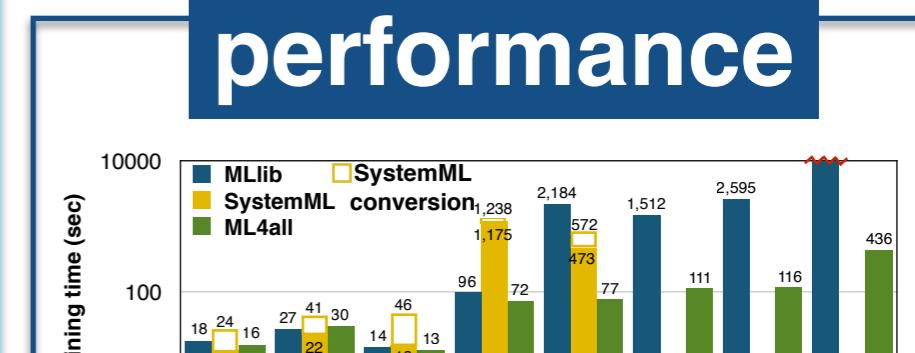
## Optimized Plans



## Speculative Estimator



## High performance



## ML4all:

<https://github.com/rheem-ecosystem/ml4all>