

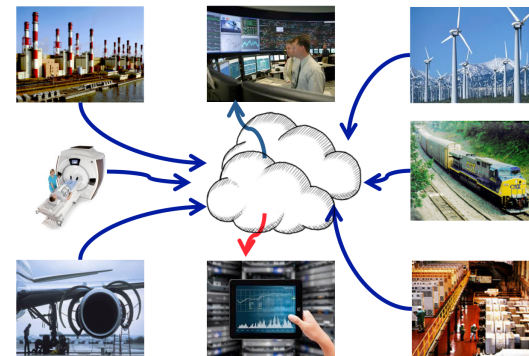
"Divide and Conquer" Machine Learning to Exploit Big Data Knowledge Discovery

Una-May O'Reilly
The Alfa Group

MIT ICT Conference
April 24, 2013



Lots of Data Everywhere



HELP



Agenda

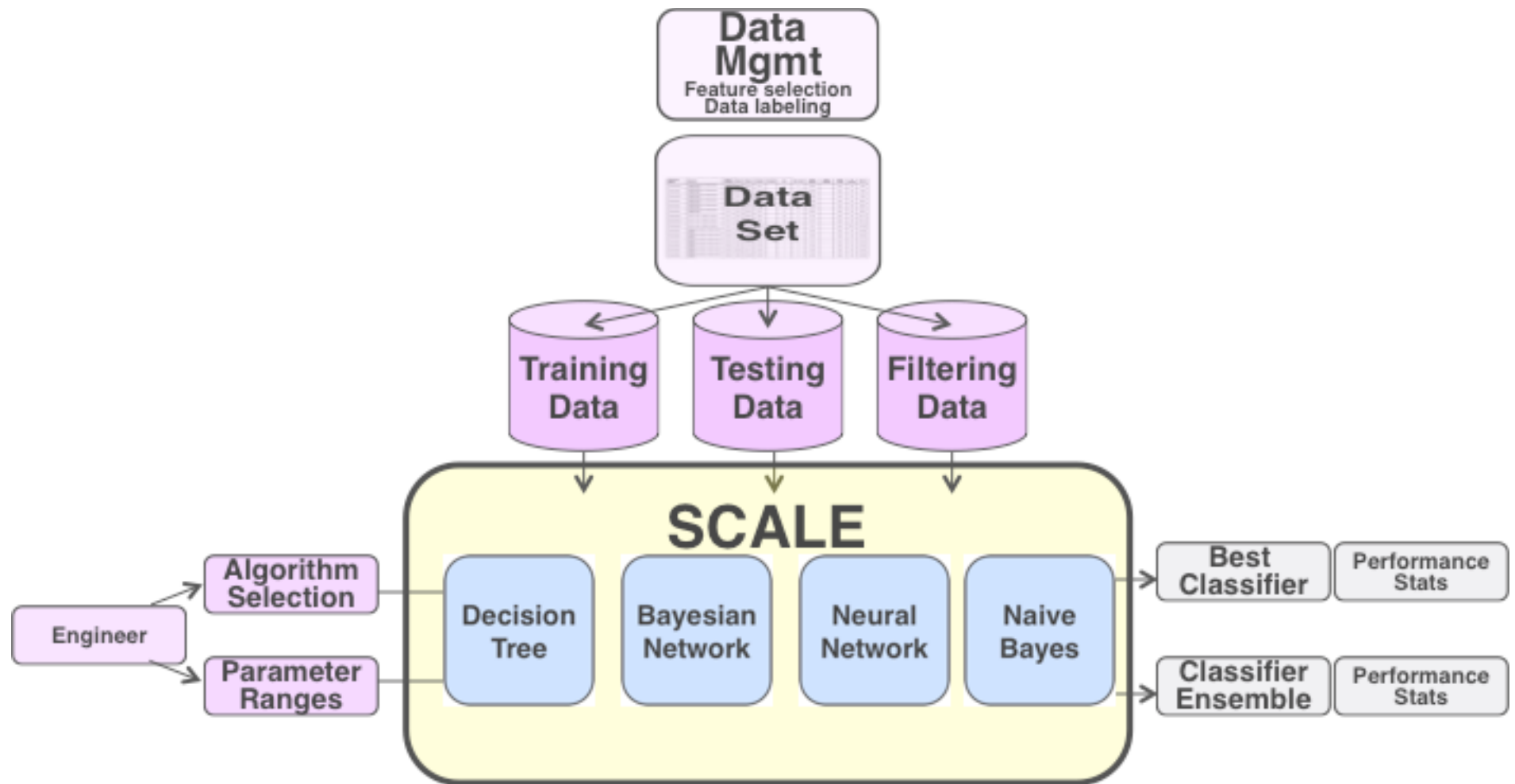
For each of SCALE, FlexGP and EC-Star

- **System Layer:**
 - Resource and Task Management/Mapping
 - Task = execute a ML algorithm on a distributed system
- **ML Algorithm Layer**
 - Algorithm scaling:
 - Divide and Conquer data strategy
 - » Factor
 - » Filter,
 - » Fuse Look at all the data everywhere
 - Experiments related to divide and conquer with the data

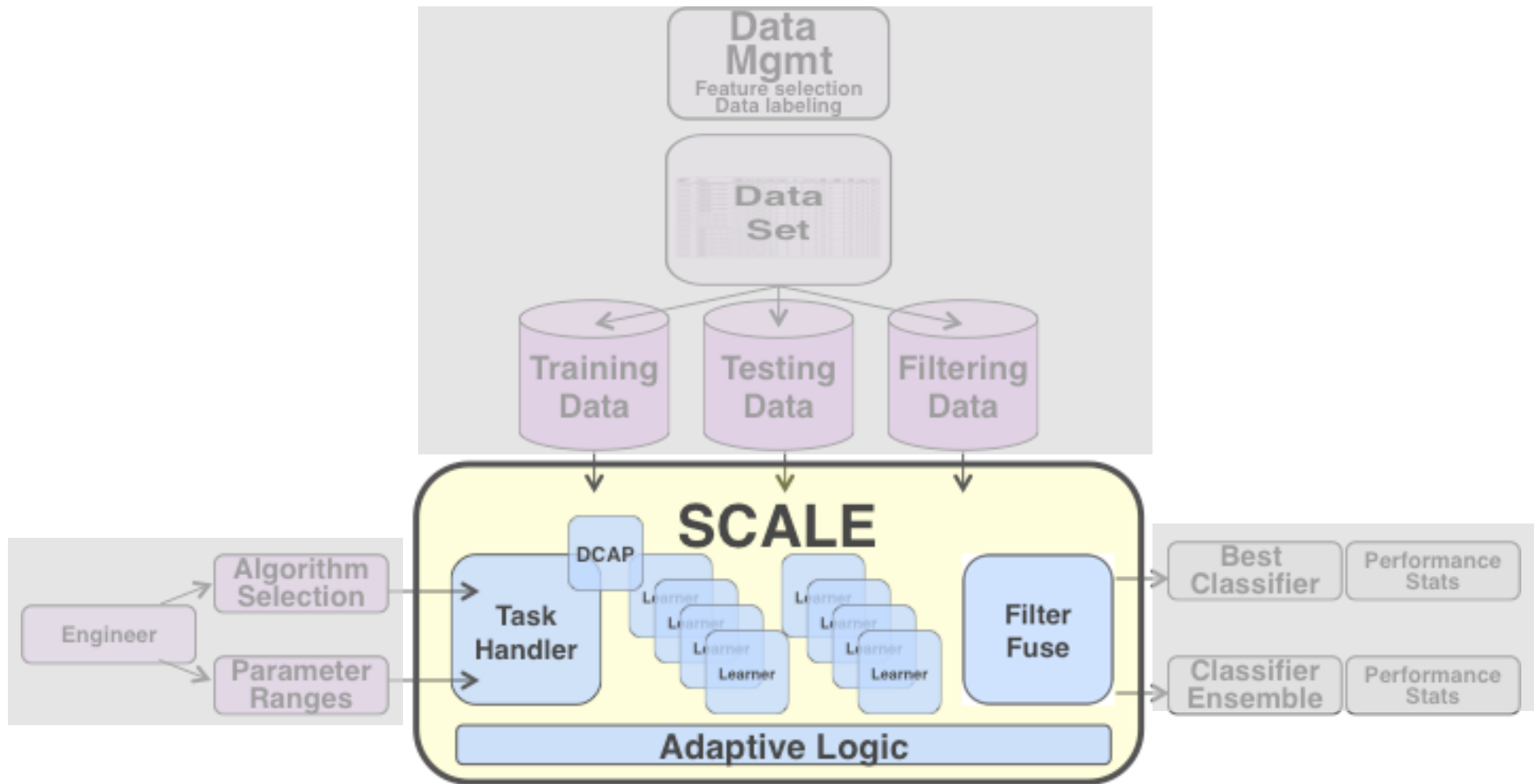
Endfor

- Compare and contrast
- Going Forward

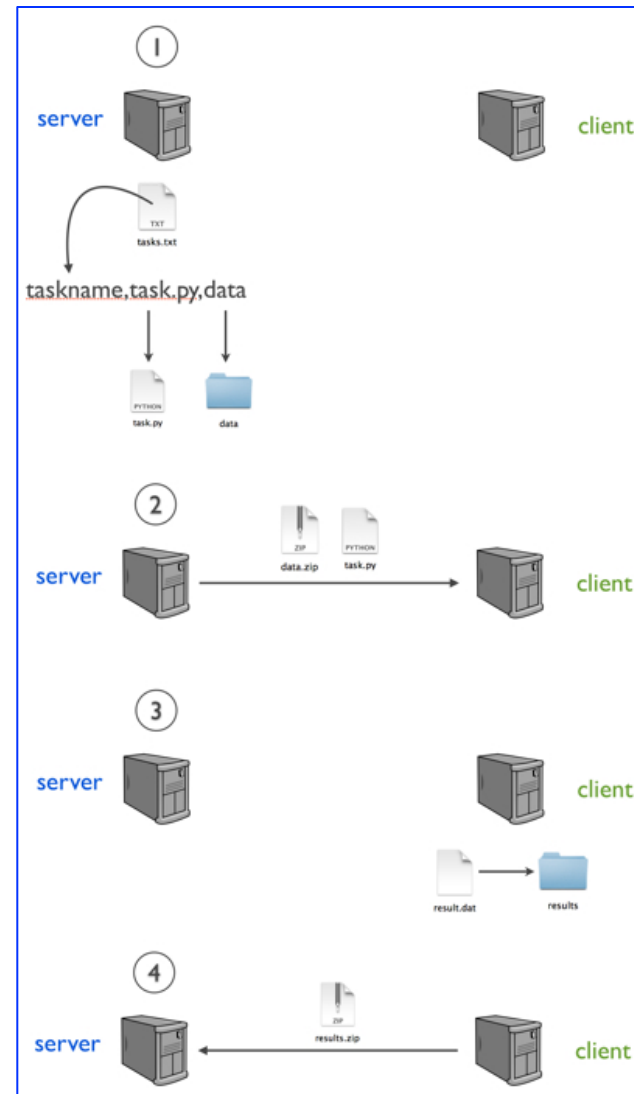
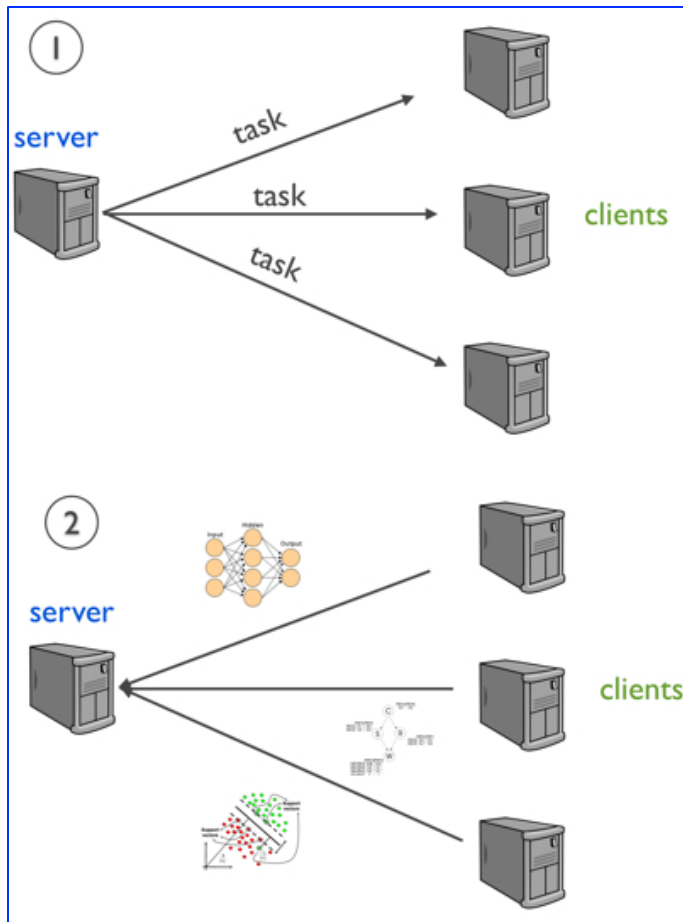
SCALE Introduction



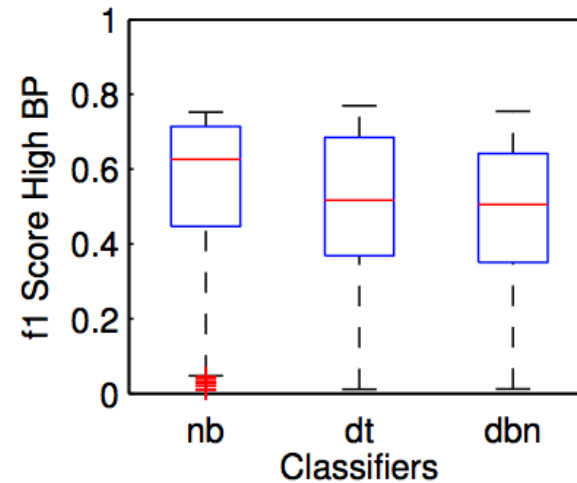
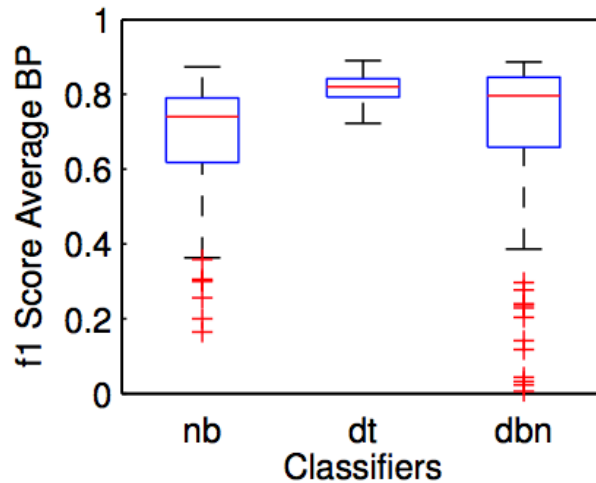
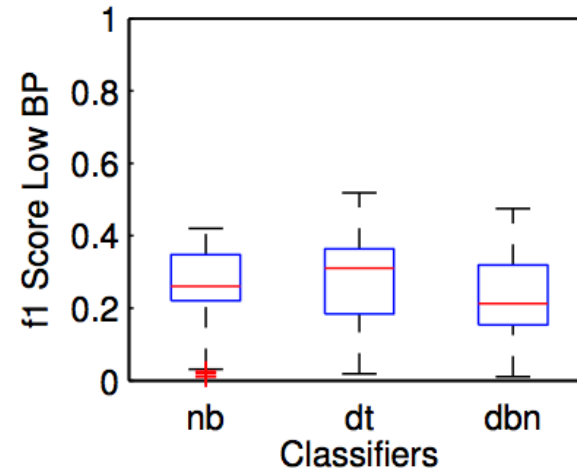
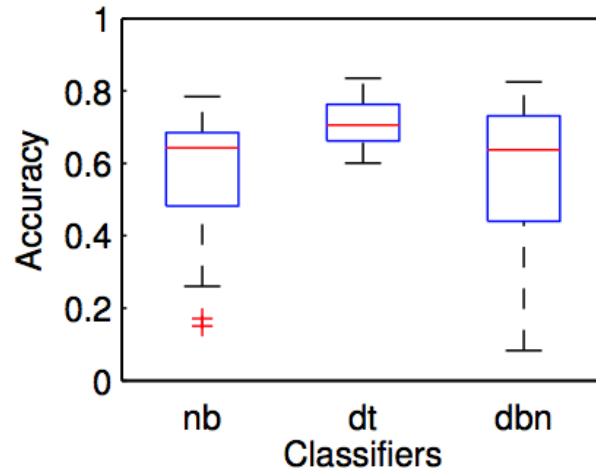
SCALE



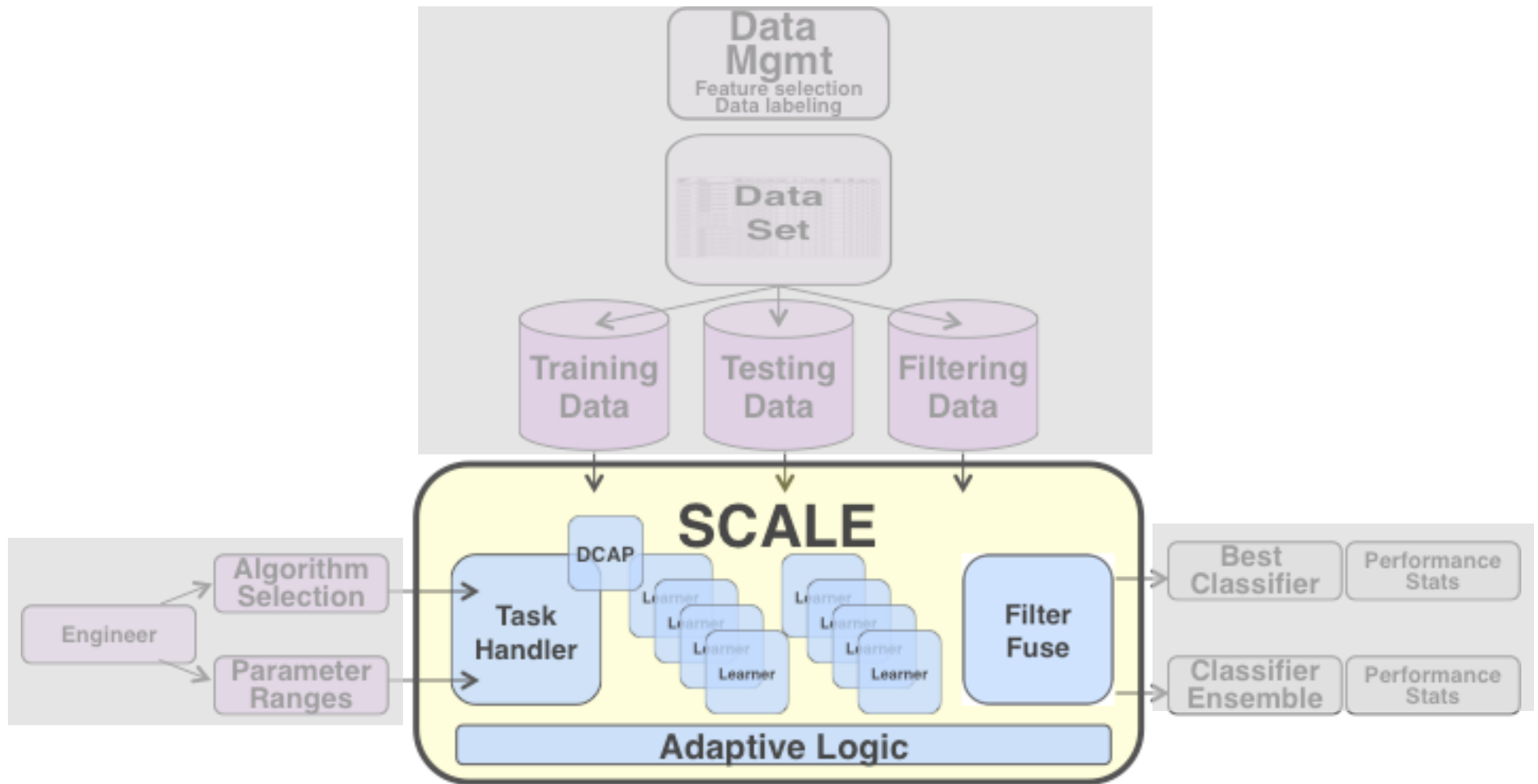
SCALE Server-Client Architecture



SCALE Results



SCALE



Scaling Up to 1000s: from SCALE to FlexGP

SCALE:

- Every learner has to know IP of task handler
- Task handler is a bottleneck and central point of failure
- No communication to accelerate learning
- Inelastic

FlexGP

- No central task handler or point of failure
- Learners gossip to learn each others IP
- Elastic

Scaling up to 1000's: from SCALE to FlexGP

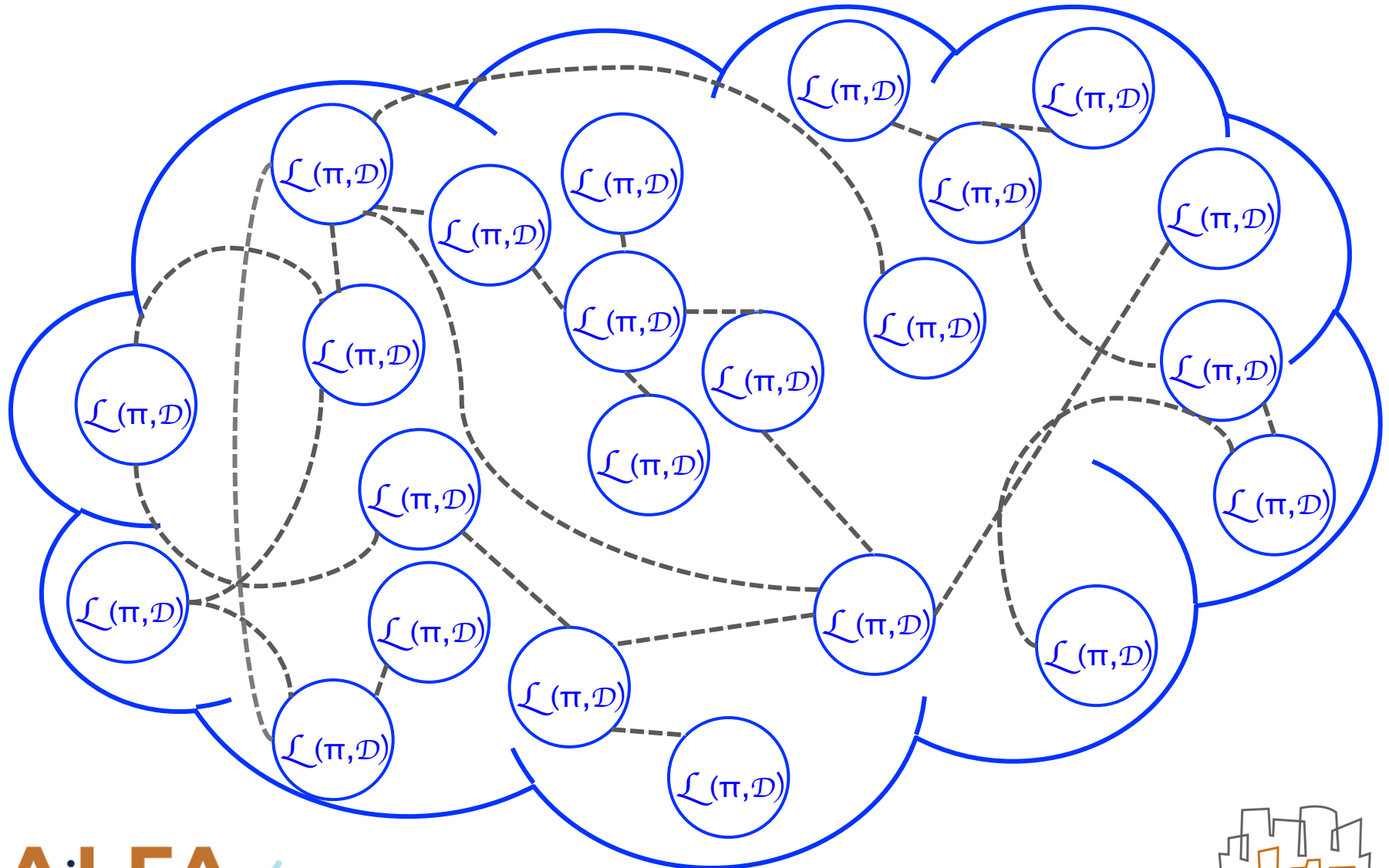
SCALE

- Modest # of features
- Data must fit into RAM
- Explicit algorithm tasks
- Small scale, serial algorithms
- No learner communication

FlexGP

- Factor
 - 100s of features-
- Big Data
- Statistically directs algorithm islands
- Large scale, distributed algorithm
 - Local algorithms coordinated
- Learners share and integrate progress

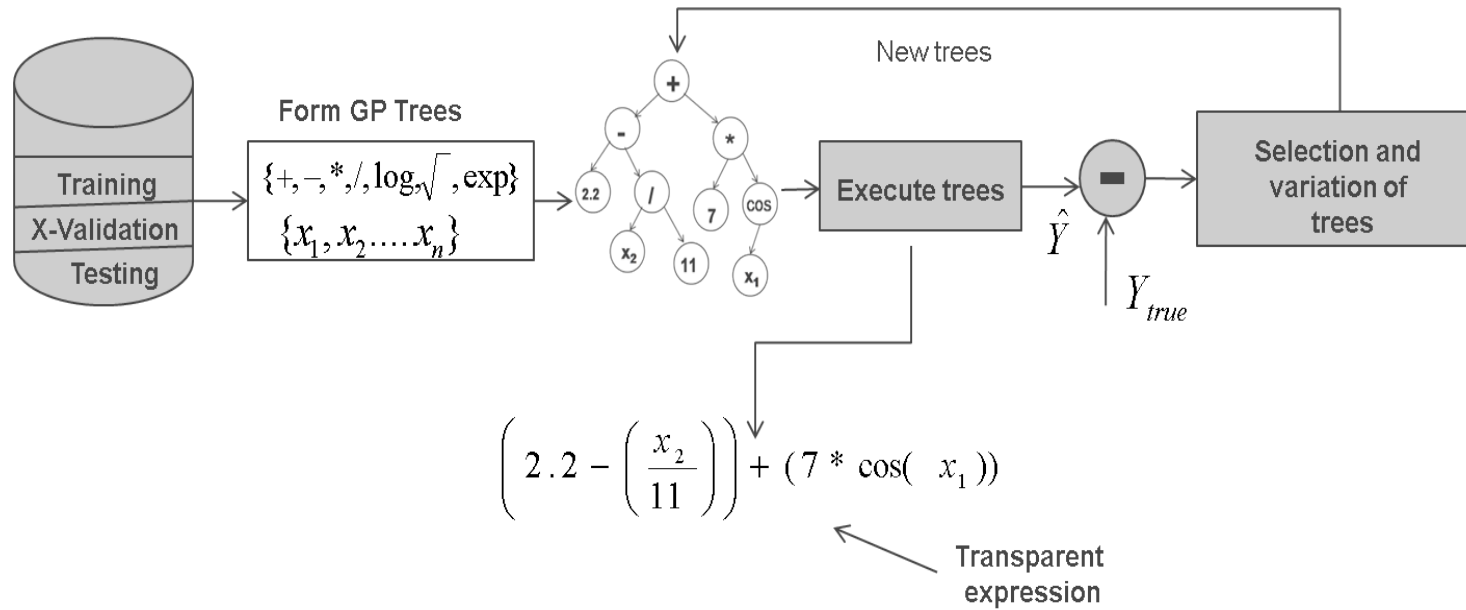
FlexGP



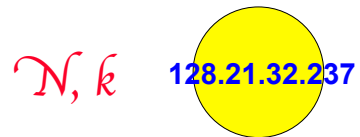
Introduction

FlexGP

Goal: Model $y = f(x_1, x_2, \dots, x_n)$

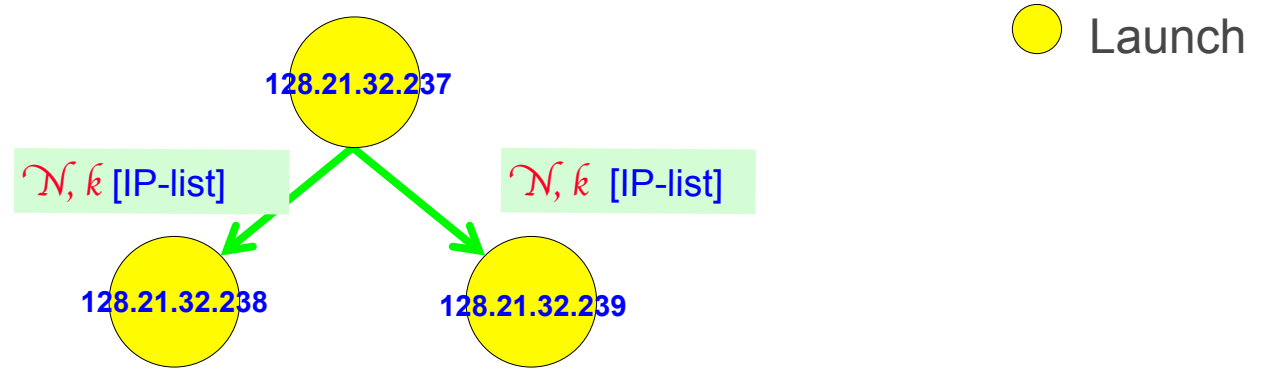


FlexGP: launch and IP discovery

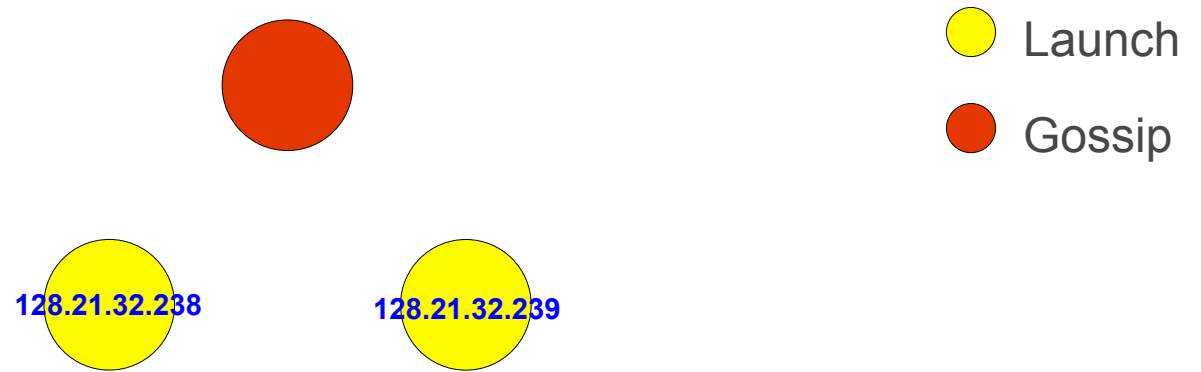


● Launch

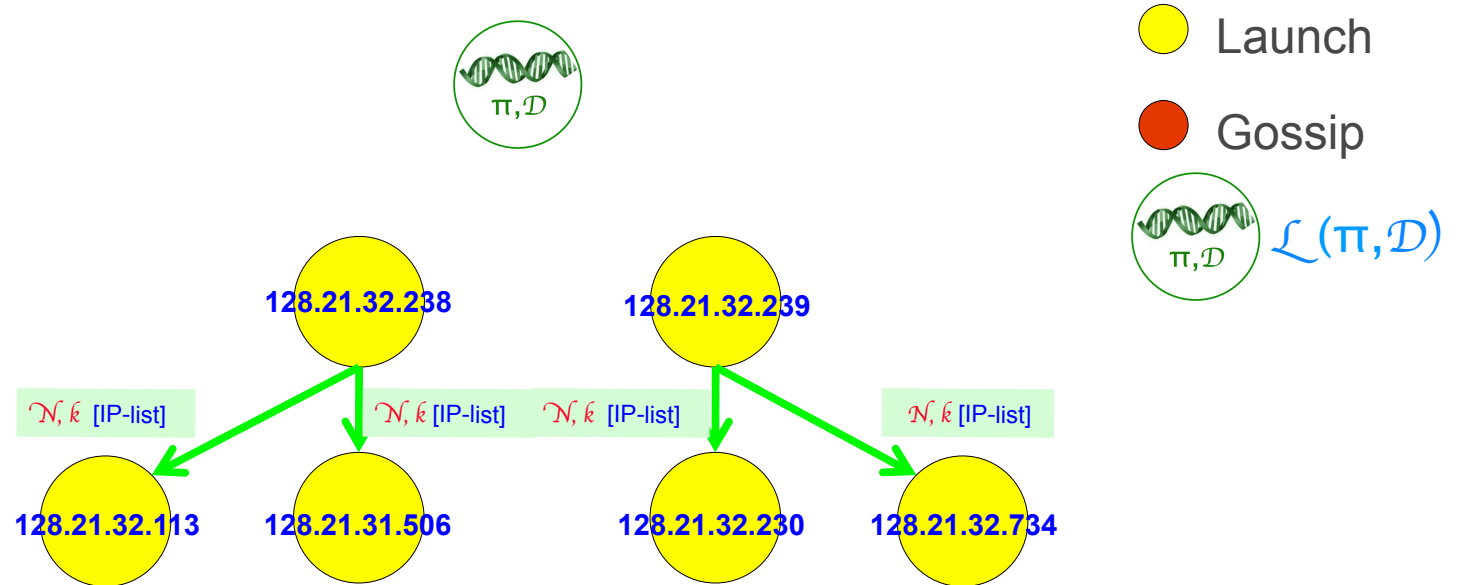
FlexGP: launch and IP discovery



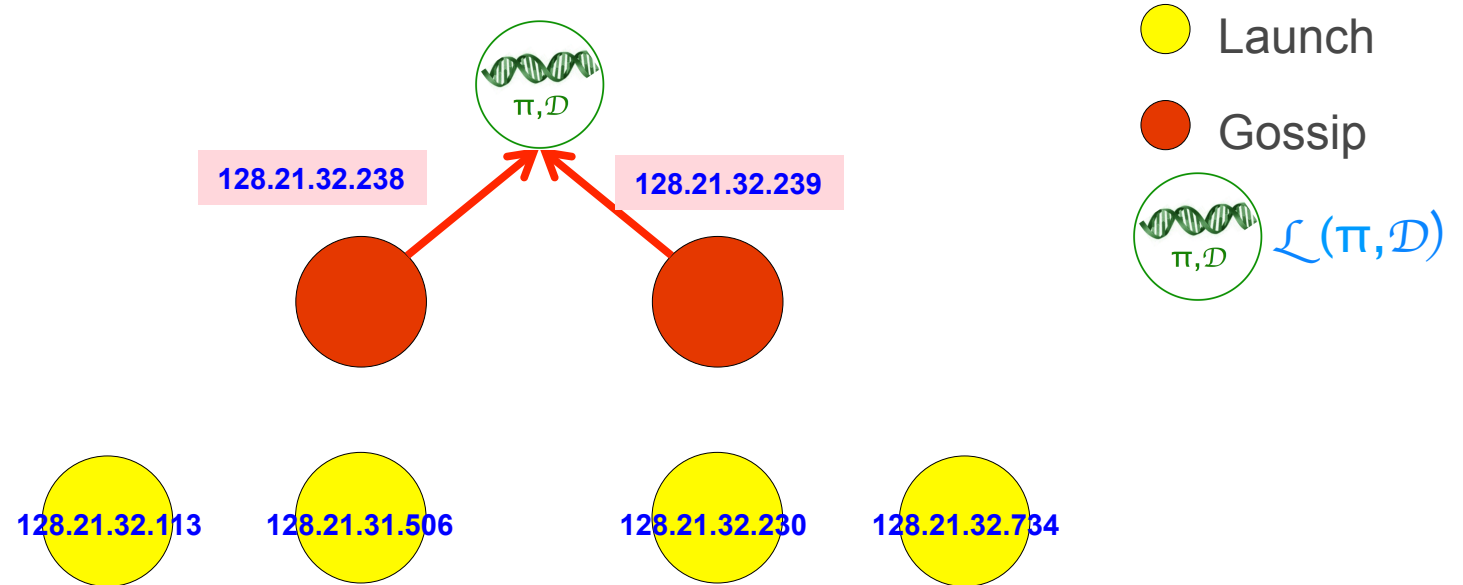
FlexGP: launch and IP discovery



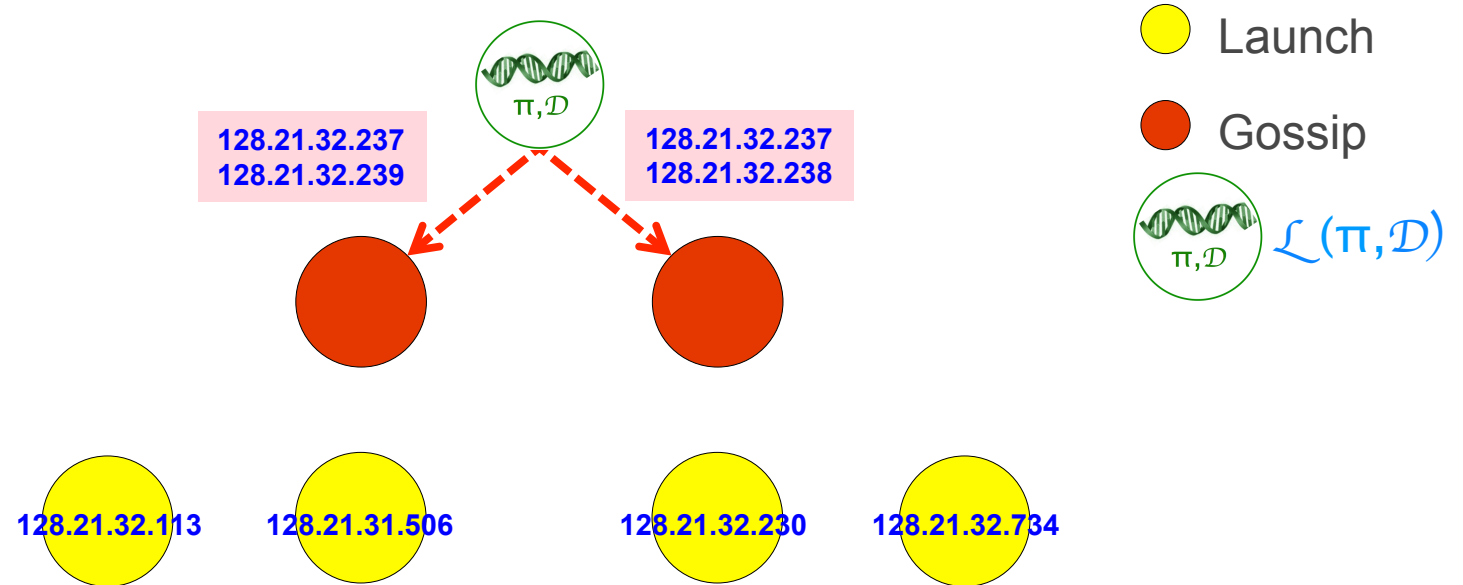
FlexGP: launch and IP discovery



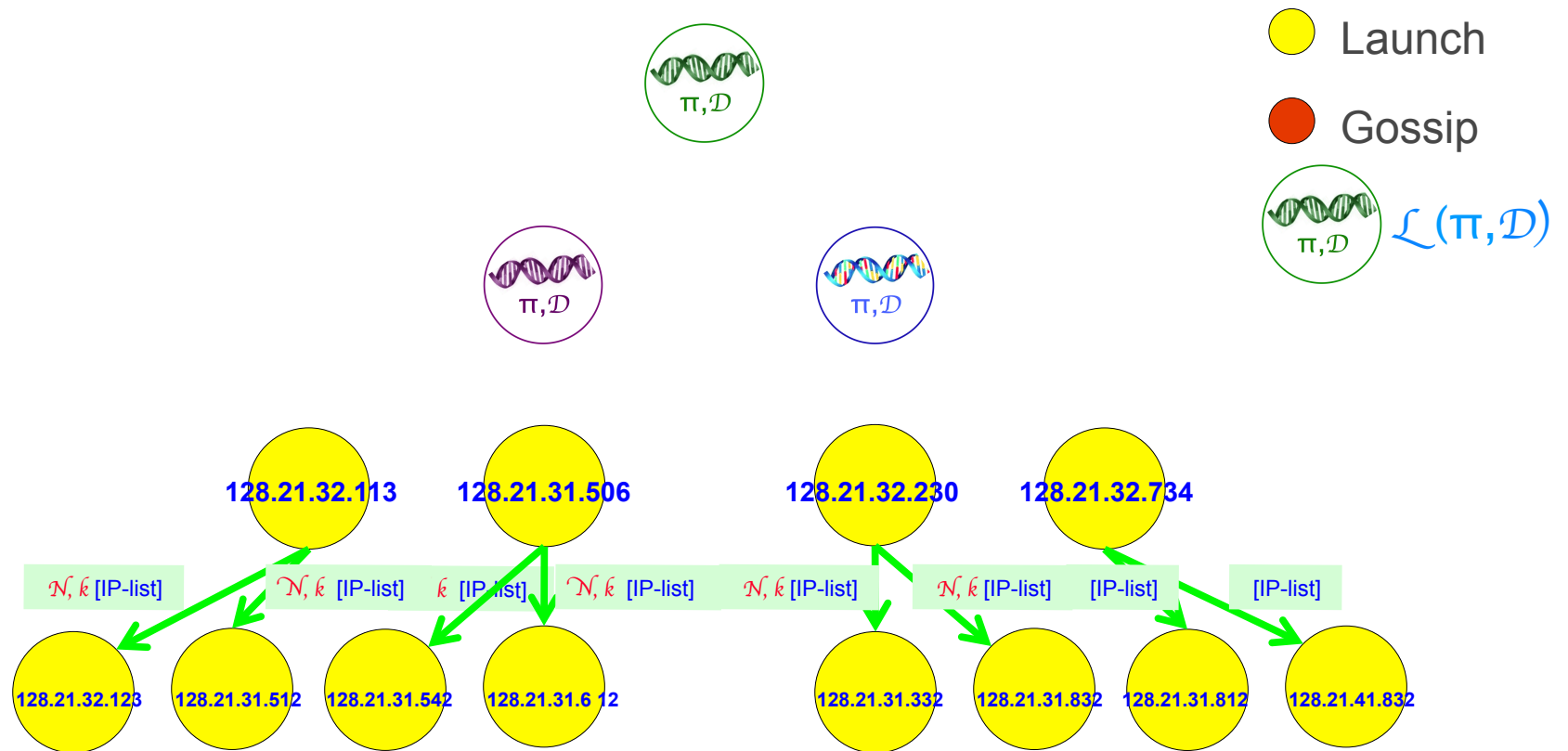
FlexGP: launch and IP discovery



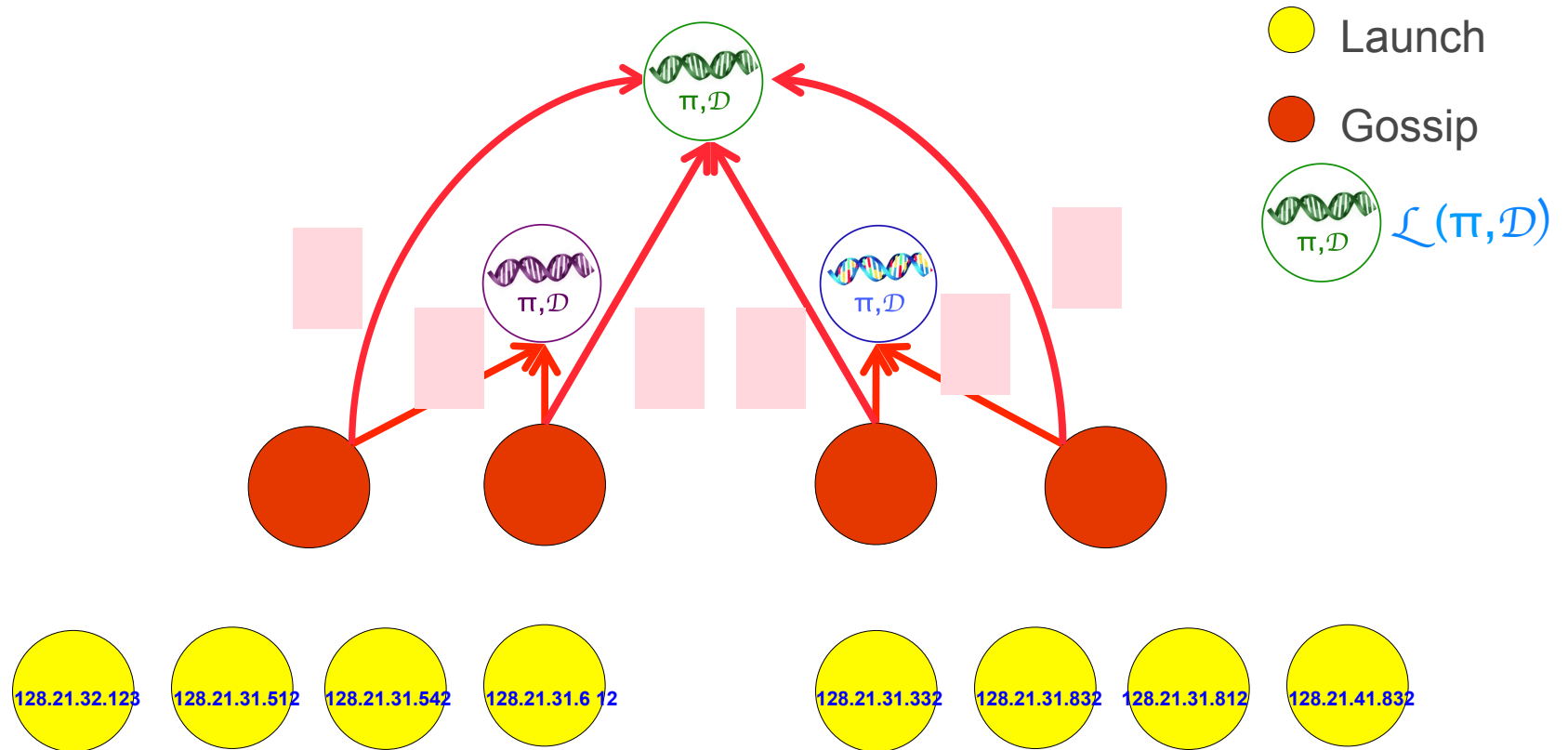
FlexGP: launch and IP discovery



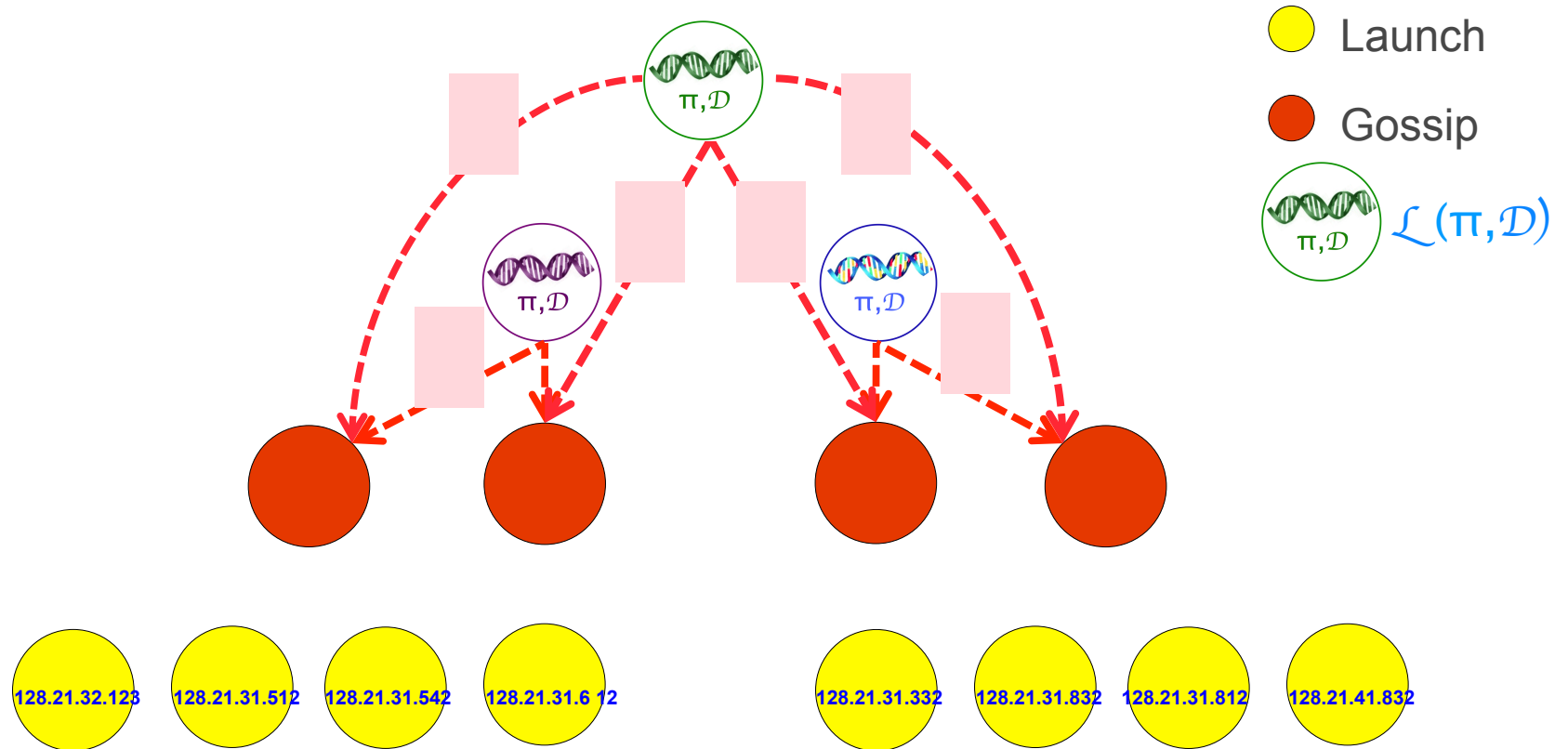
FlexGP: launch and IP discovery



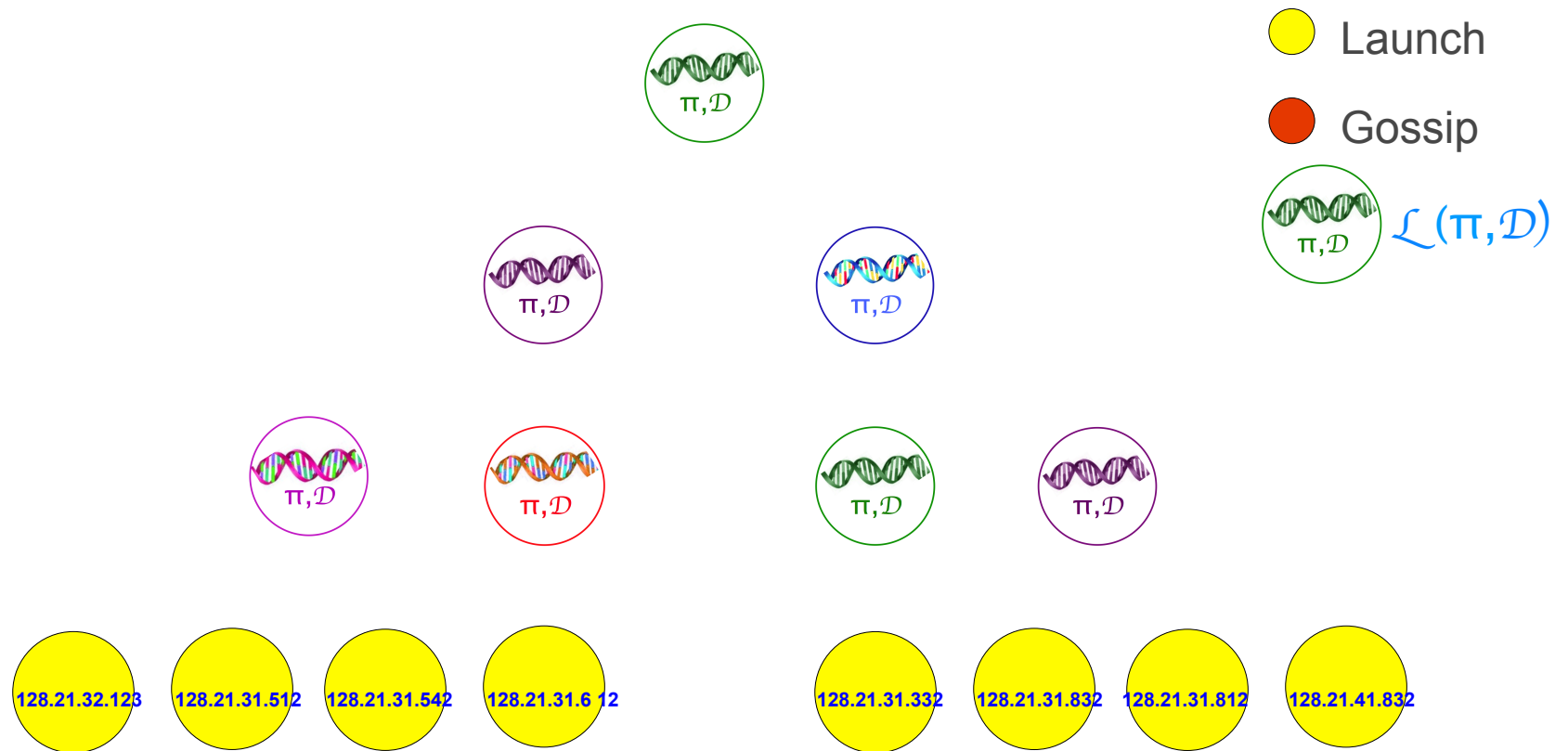
FlexGP: launch and IP discovery



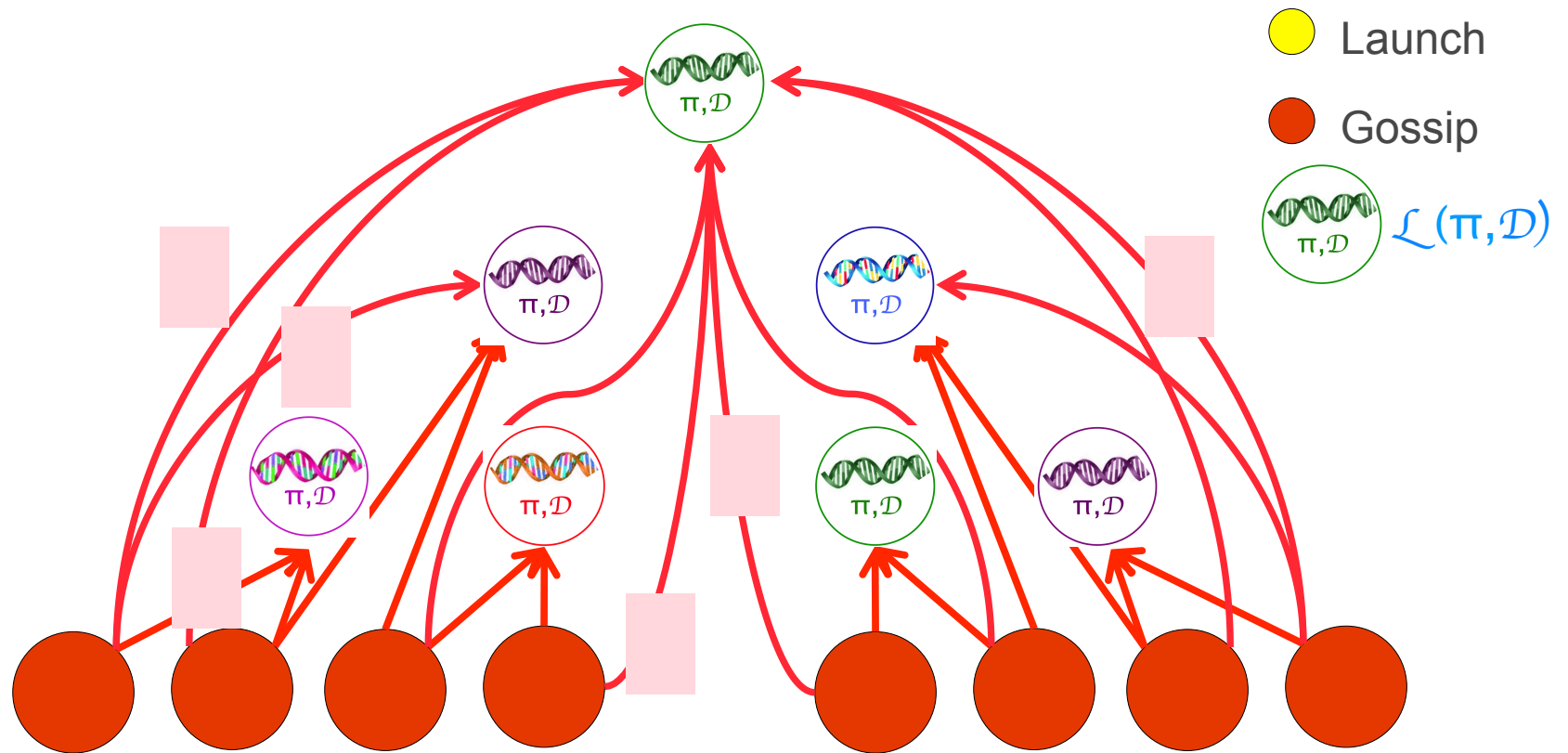
FlexGP: launch and IP discovery



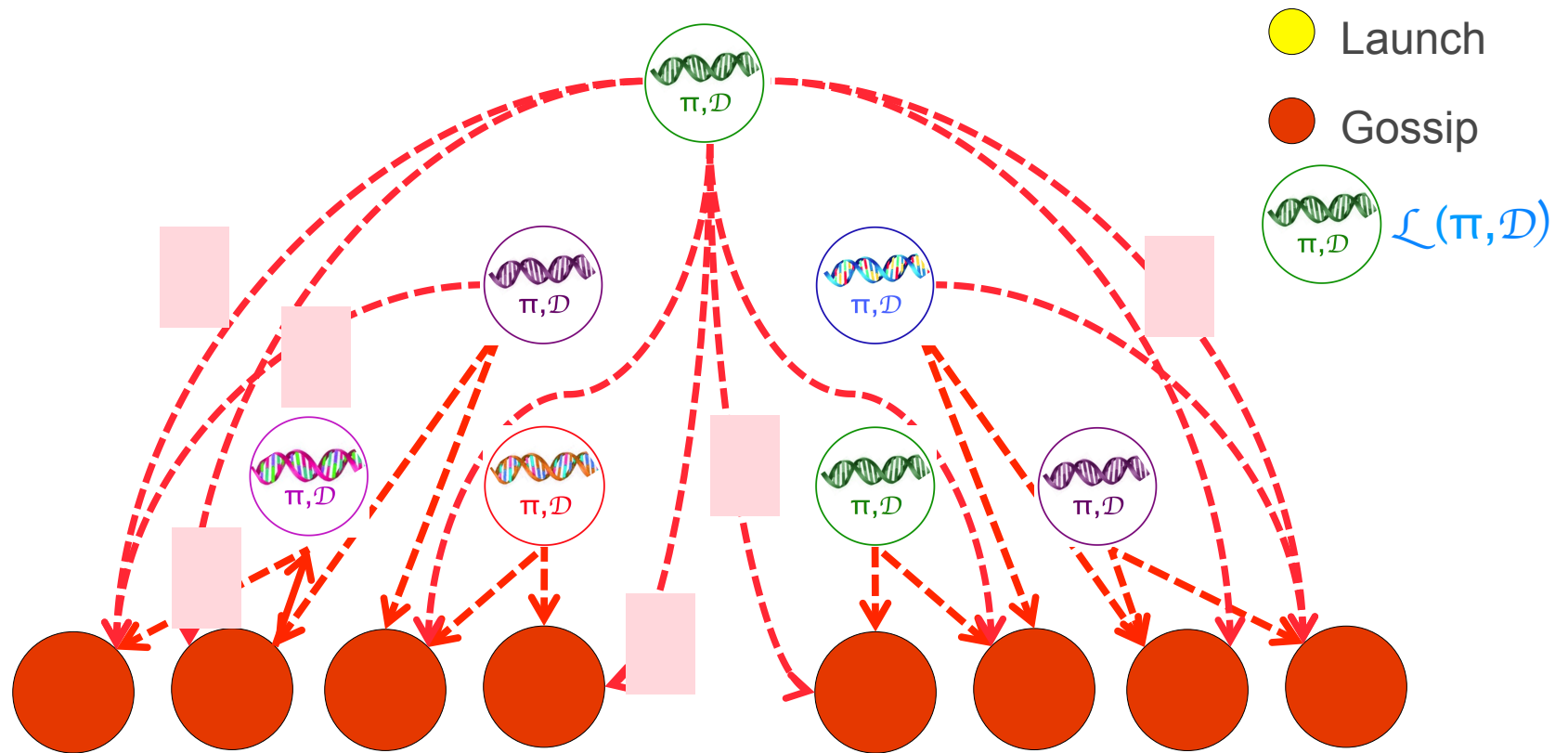
FlexGP: launch and IP discovery



FlexGP: launch and IP discovery

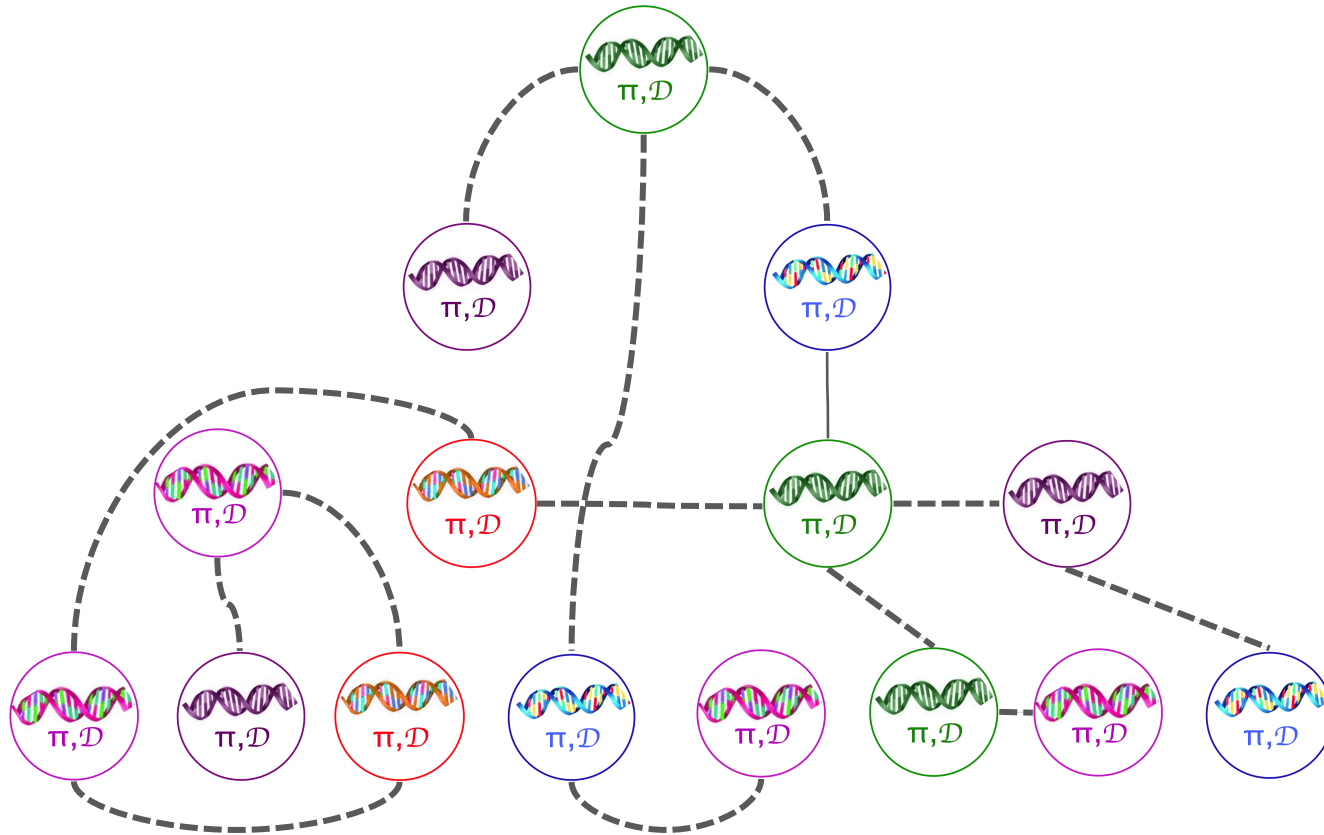


FlexGP: launch and IP discovery

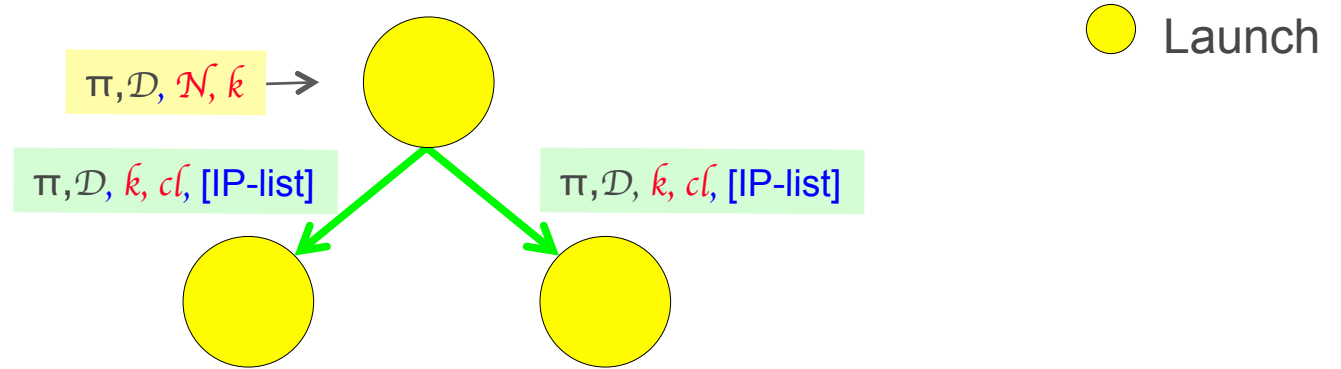


Launch complete!

... and ready to expand or contract
.... and gossiping intermittently



Statistically Parameterized Factoring

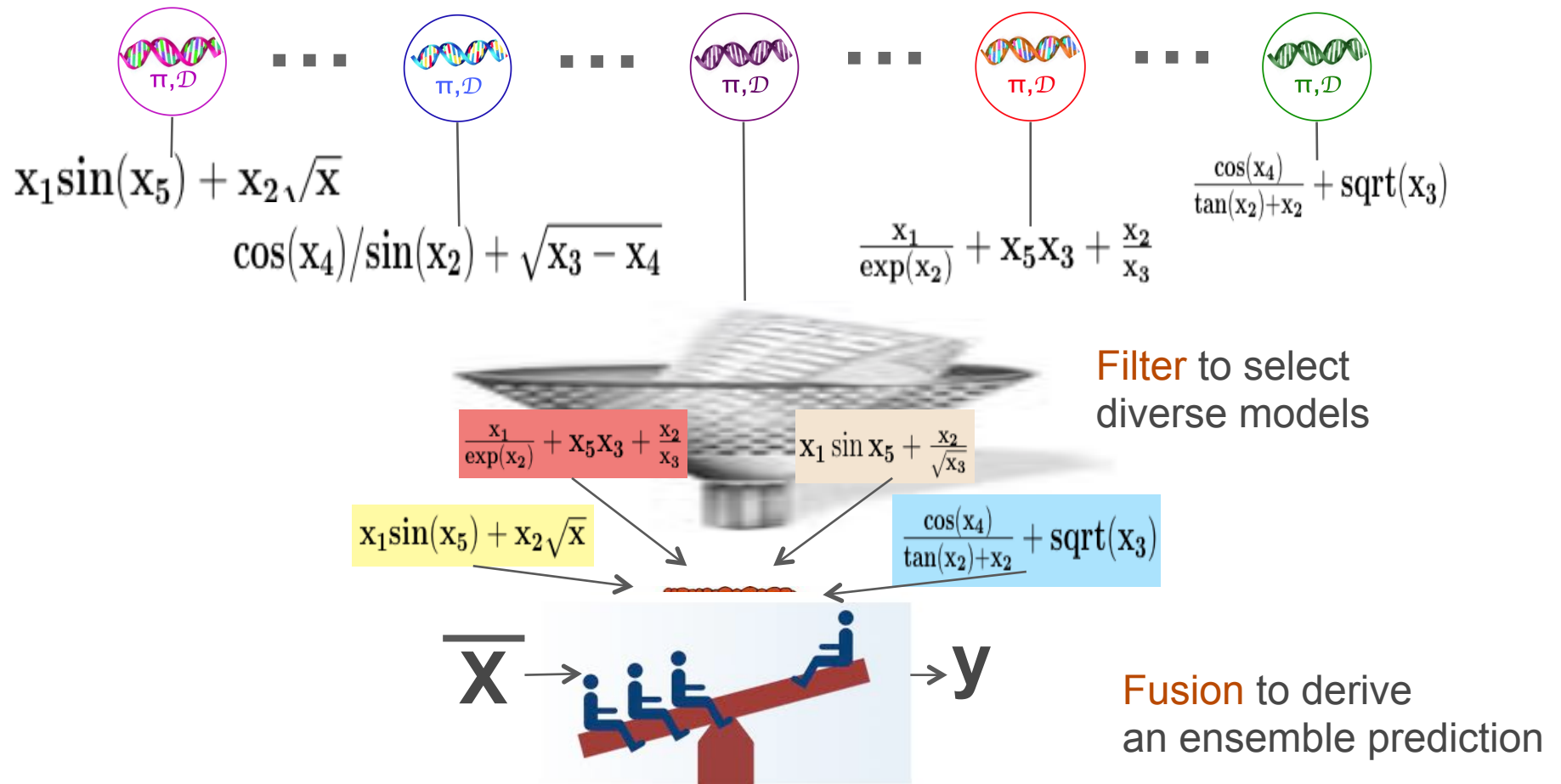


Π : Probability of feature, objective function, operator
 $\mathcal{D}$: factoring of the data

Divide and Conquer

- **Factor:**
 - Random subsets using statistical distribution
 - » Demonstrate independent learners -> weak learners
 - Features
- **Filter:**
 - uncorrelated and accurate models
- **Fuse**
 - How far can we pare down the dataset?
 - » Effectiveness of divide and conquer
 - Show results and conclude whether it is sensitive to dataset?
 - harvest "best to date" results from the system as it continues to work away

FlexGP Filter and Fusion



Divide and Conquer

- Factor: Random subsets using statistical distribution
 - Demonstrate independent learners -> weak learners
- Filter:
 - uncorrelated and accurate models
- Fuse
 - Which fusion algorithm is best?
 - Is fusion better than best?
 - How far can we pare down the dataset?
 - » Effectiveness of divide and conquer
 - Show results and conclude whether it is sensitive to dataset?
 - Harvest "best to date" results from the system as it continues to work away

FlexGP Fusion: Better than Best?

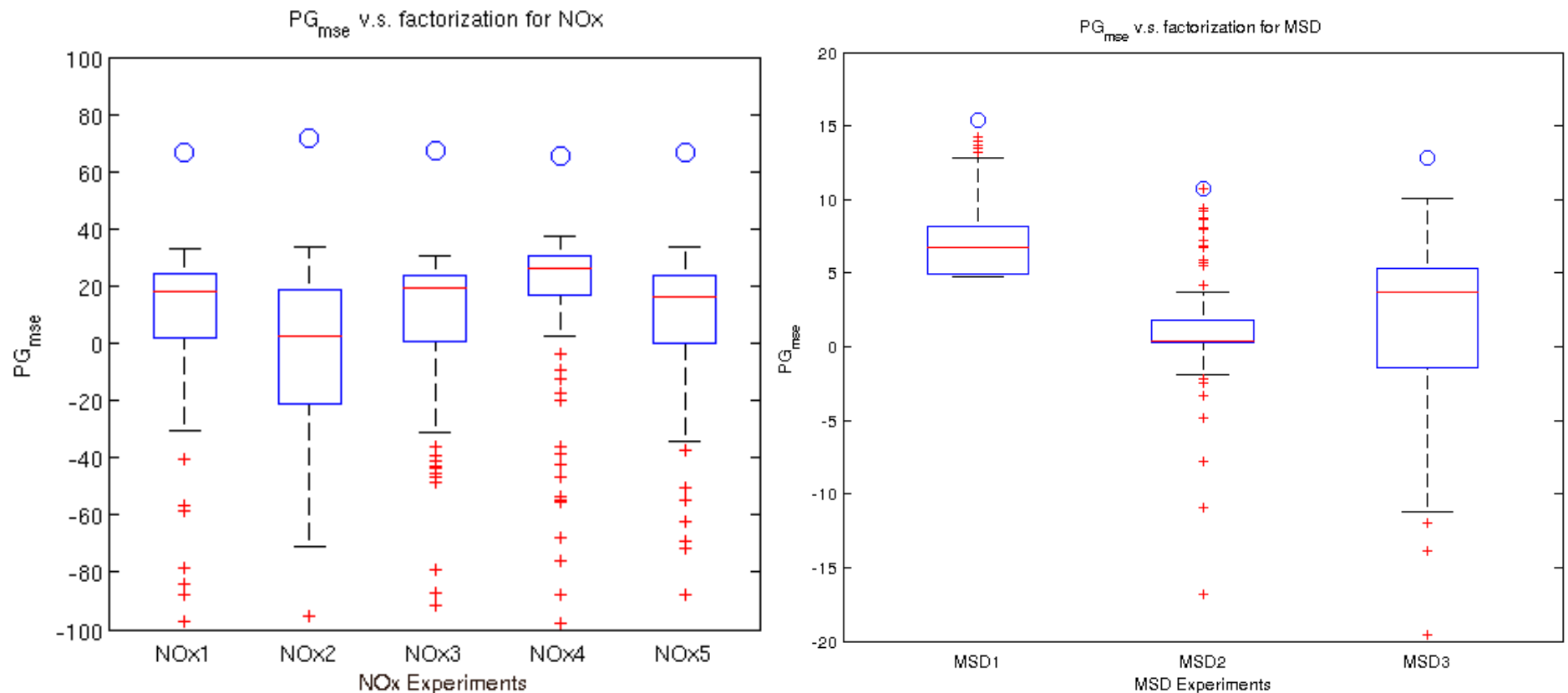
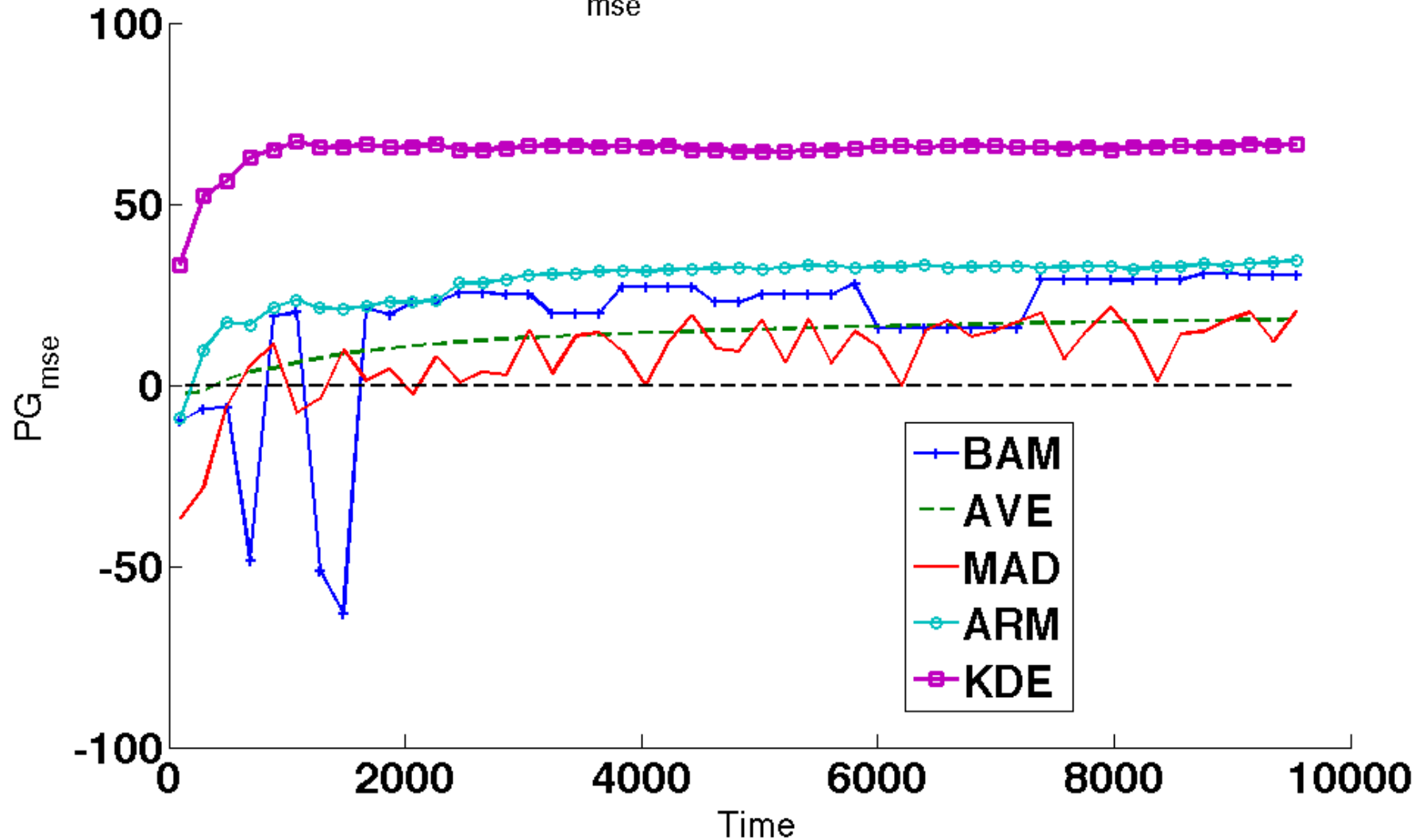


Figure 1: The quartile distribution of PG_{MSE} of models used for fusion in each experiment. The circles represent the best PG_{MSE} from fusion. *Left* Results for NOx experiments; KDE was the best fusion method. *Right* Results for MSD experiments; ARM was the best fusion method.

FlexGP: Best So Far Fusion

PG_{mse} v.s. Time for NOx1

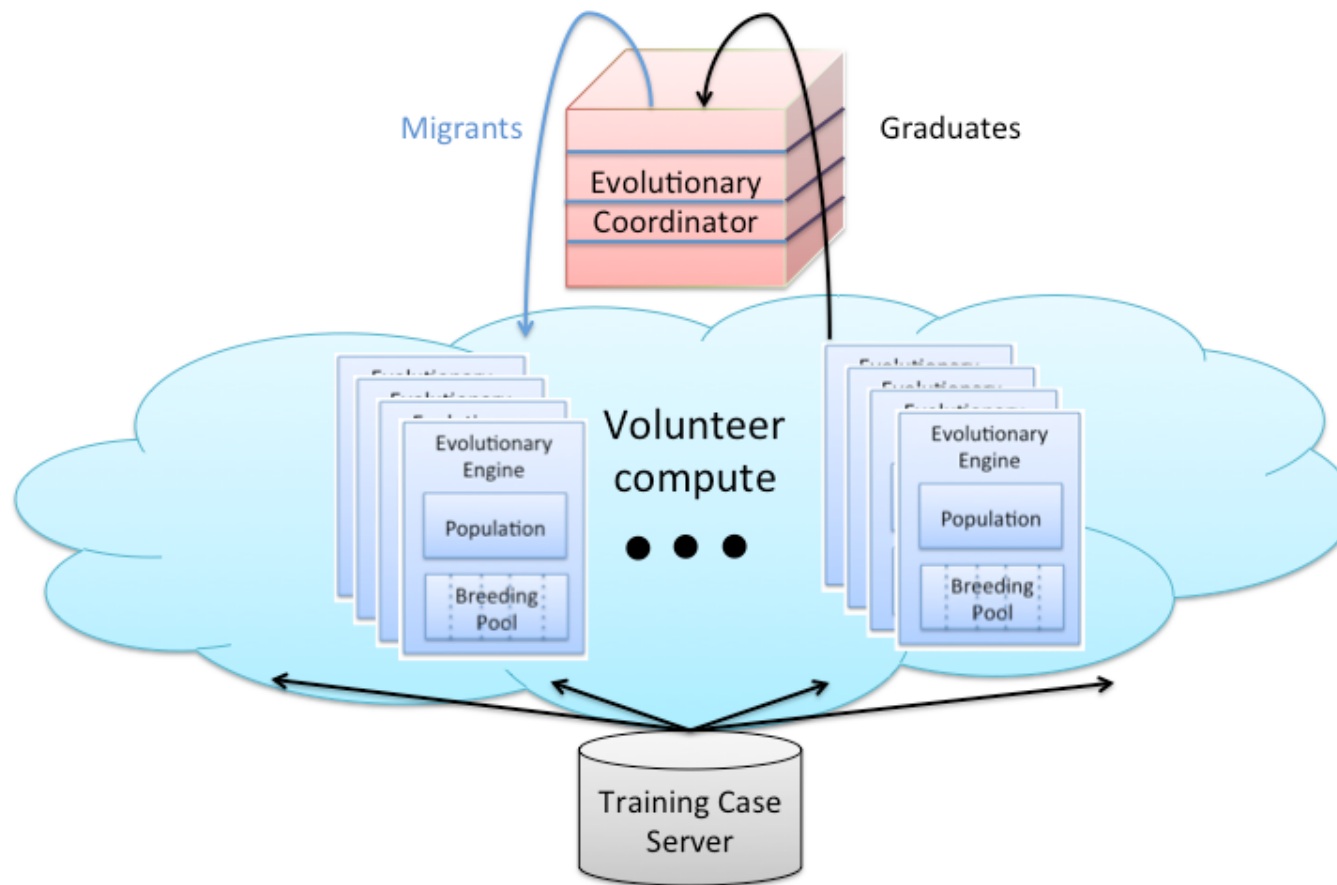


ECStar

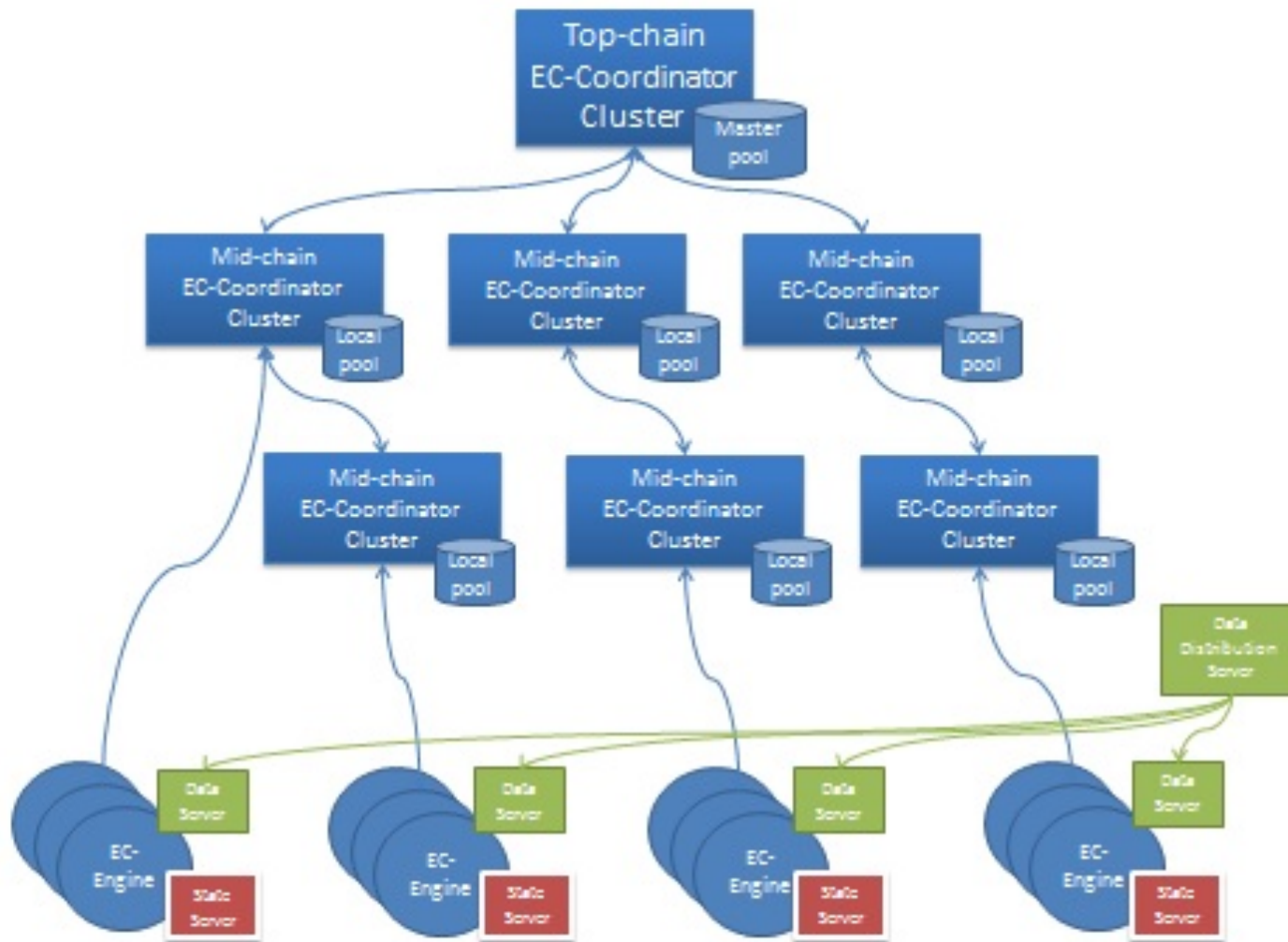
- **Goal: compute very cost effectively on *VAST* number of nodes**
 - Runs on thousand to 10'Ks 100K's million nodes
 - Vast requires cost effective -> volunteer
- **Domain: learn from time series**
 - Finance, medical signals domain
- **Solution is strategy or classifier expressed as rule sets**

FromFlexGP to ECStar

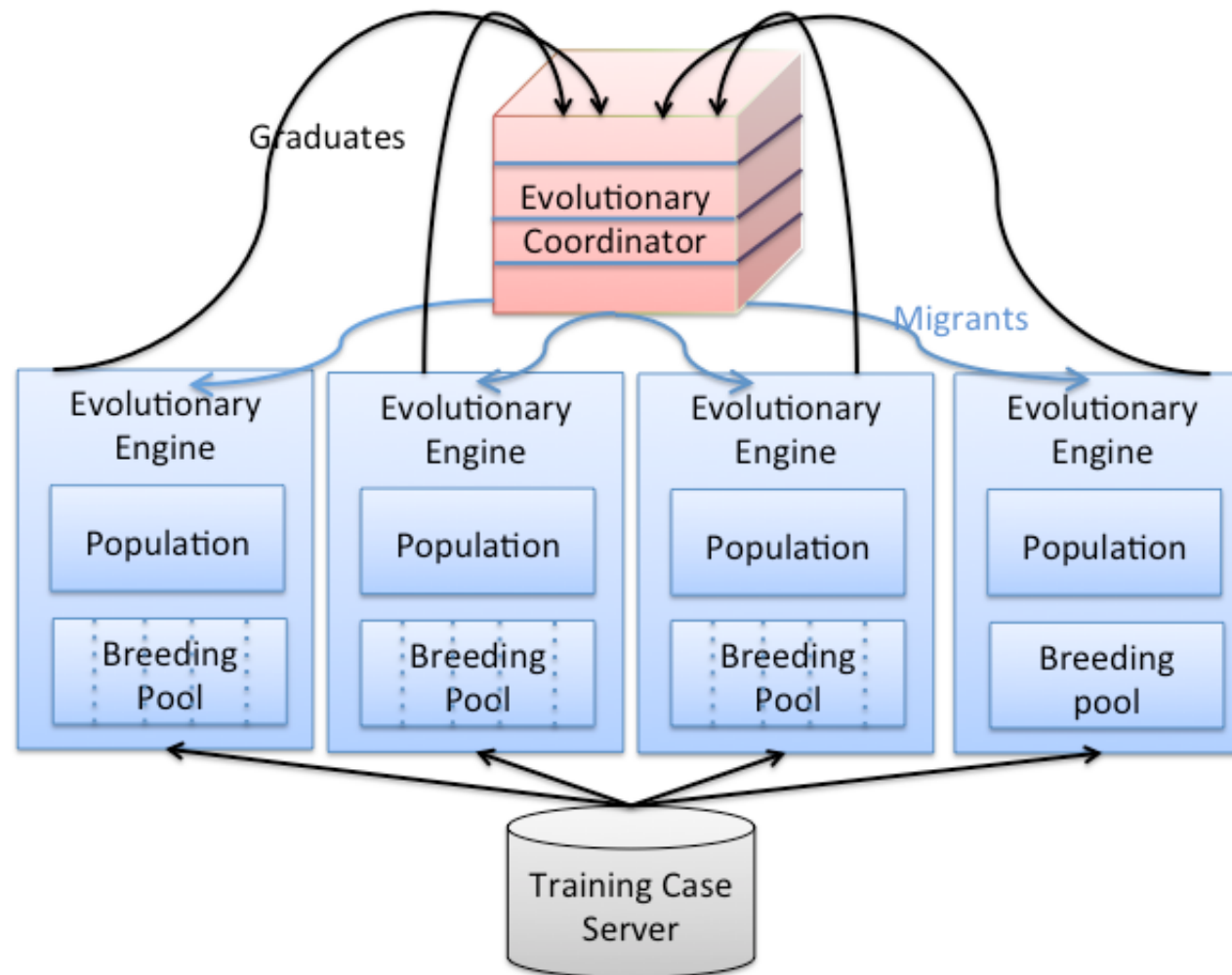
- Clear separation between system layer vs algorithm layer
- The volunteer compute nodes of EC-Star change that picture
 - They have unpredictable availability – when they start and stop
 - A client's host can fail
 - Host imposes
 - » Small memory footprint,
 - » need to save and migrate state
 - » client-to-client communication ban
 - » Design decisions negotiating responsibilities between VCN and dedicated servers
 - » Result is a distributed algorithm with divide and conquer strategy for data handling that is
 - tightly integrated with the resource layer design



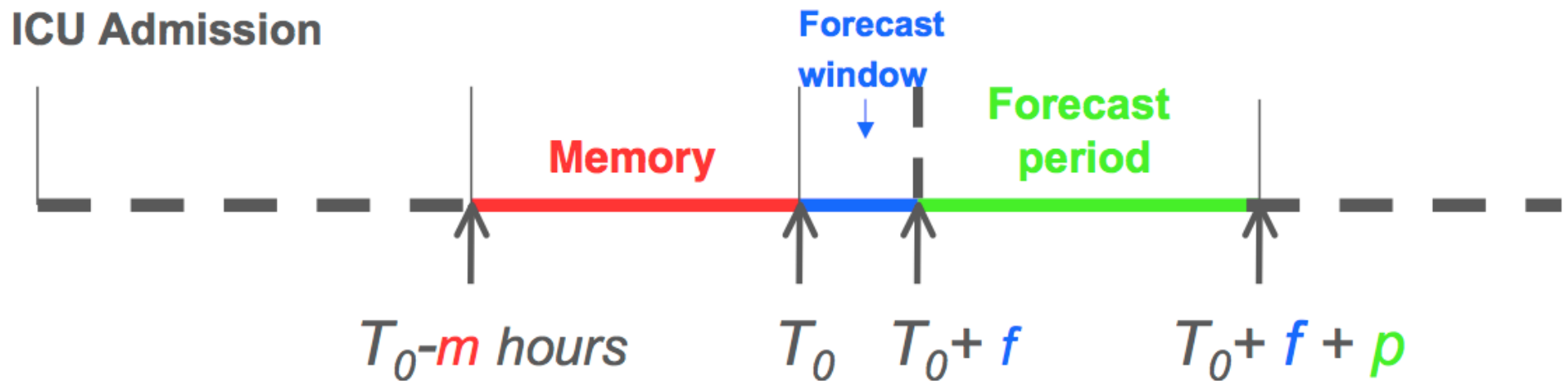
Resources Federation



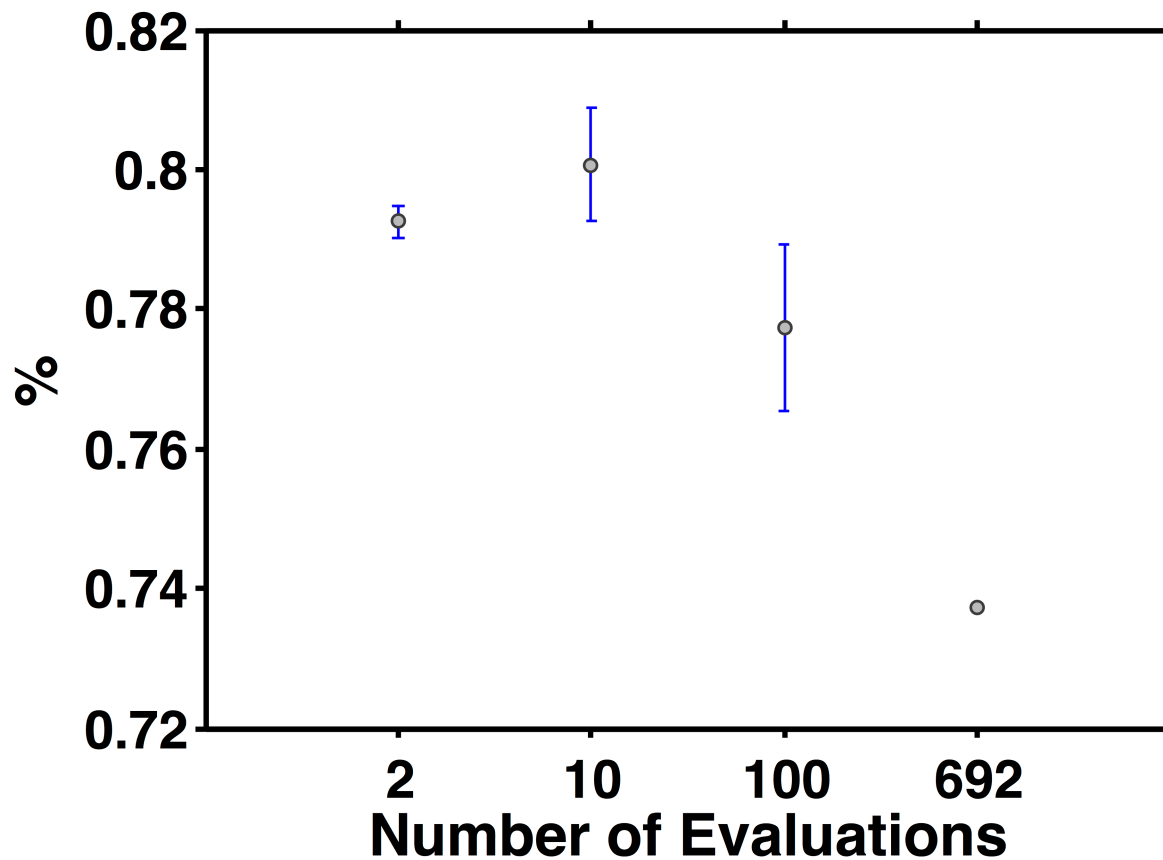
Under to Over Sampling



Blood Pressure Problem



Impact of Partially Evaluating Models



System Layer Comparison

	Scale	FlexGP	EC-Star
ML domain	Classification	Regression Classification	Rule Learning
Resource Scale	10's to 100	100's to 1000	10^3 to 10^6
Resource Type	Cloud	Cloud	Volunteer and Dedicated
Fusion	External	External	Integrated
Local Algorithm	Different	Same	Same
Server:Client ratio	1: many	Decentralized	many:many

	SCALE	FlexGP	EC-Star
Factor	Subsets	Subsets	Under to oversampling
Filter	Correlation Accuracy	Correlation Accuracy	Layered competition
Fuse	Voting	Non-parametric output space approaches	Migration and ancestral properties

Automation

- "In the end, the biggest bottleneck is not data or CPU cycles, but human cycles."



ML requires a lot of Human Effort

Domain Knowledge
Analysis and Transfer

Problem Definition

Data Preconditioning

Feature
Identification

Feature Extraction



Algorithm Selection

Algorithm Customization

Parameter Selection

Training and Test Data
Selection

Results Evaluation

Solution Deployment

Thanks for your attention!

Thanks to...

- **ALFA group members**
 - Large team of students
 - Postdoc: Dr. Erik Hemberg
 - Research Scientist: Dr. Kalyan Veeramachaneni
- **Our collaborators and sponsors**



李嘉誠基金會
LI KA SHING FOUNDATION



imagination at work



We bring good things to life.

