Activity Analysis: Finding Explanations for Sets of Events

Dima Damen, David Hogg

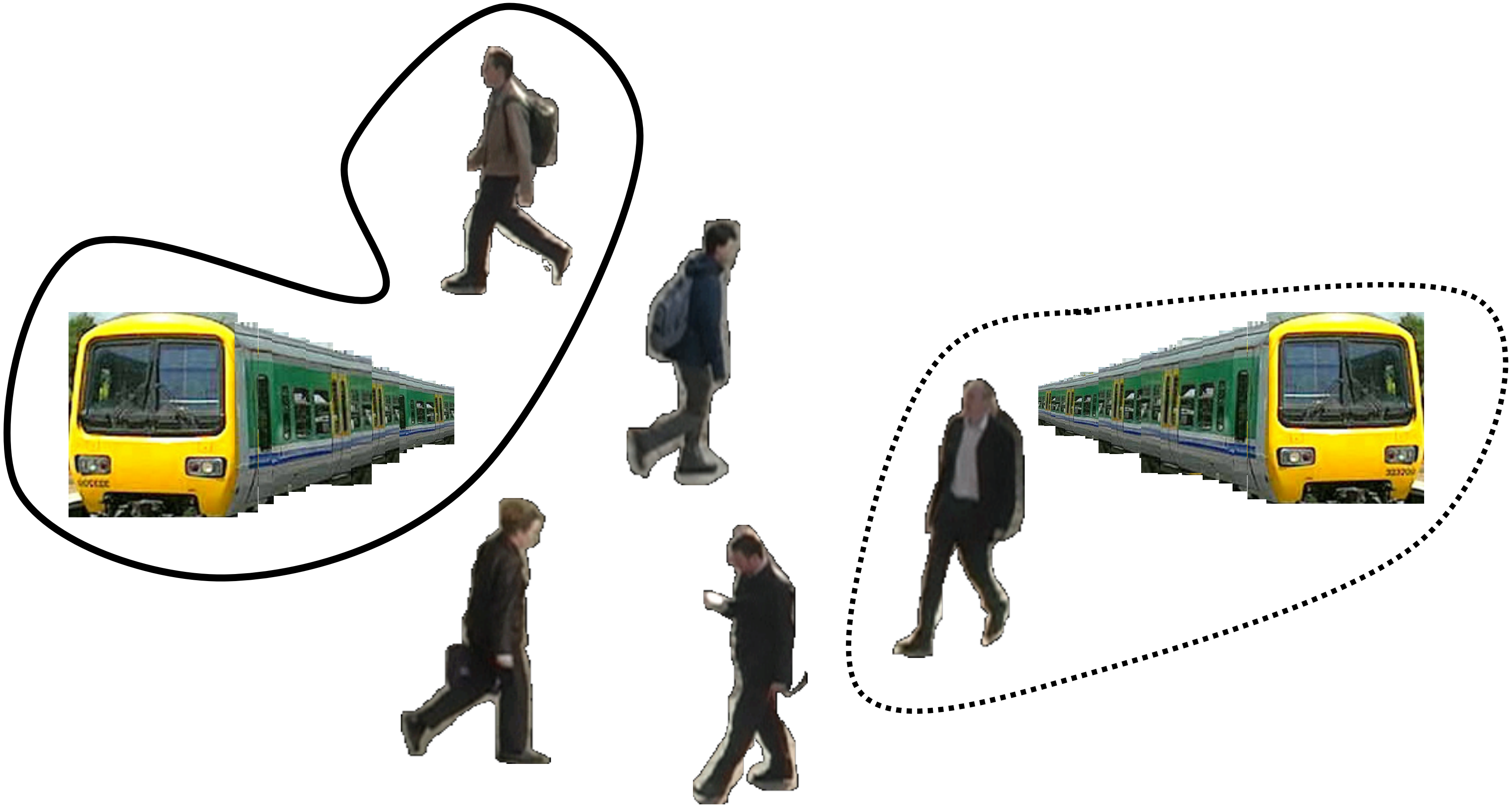
Computer Vision Group

UNIVERSITY OF LEEDS

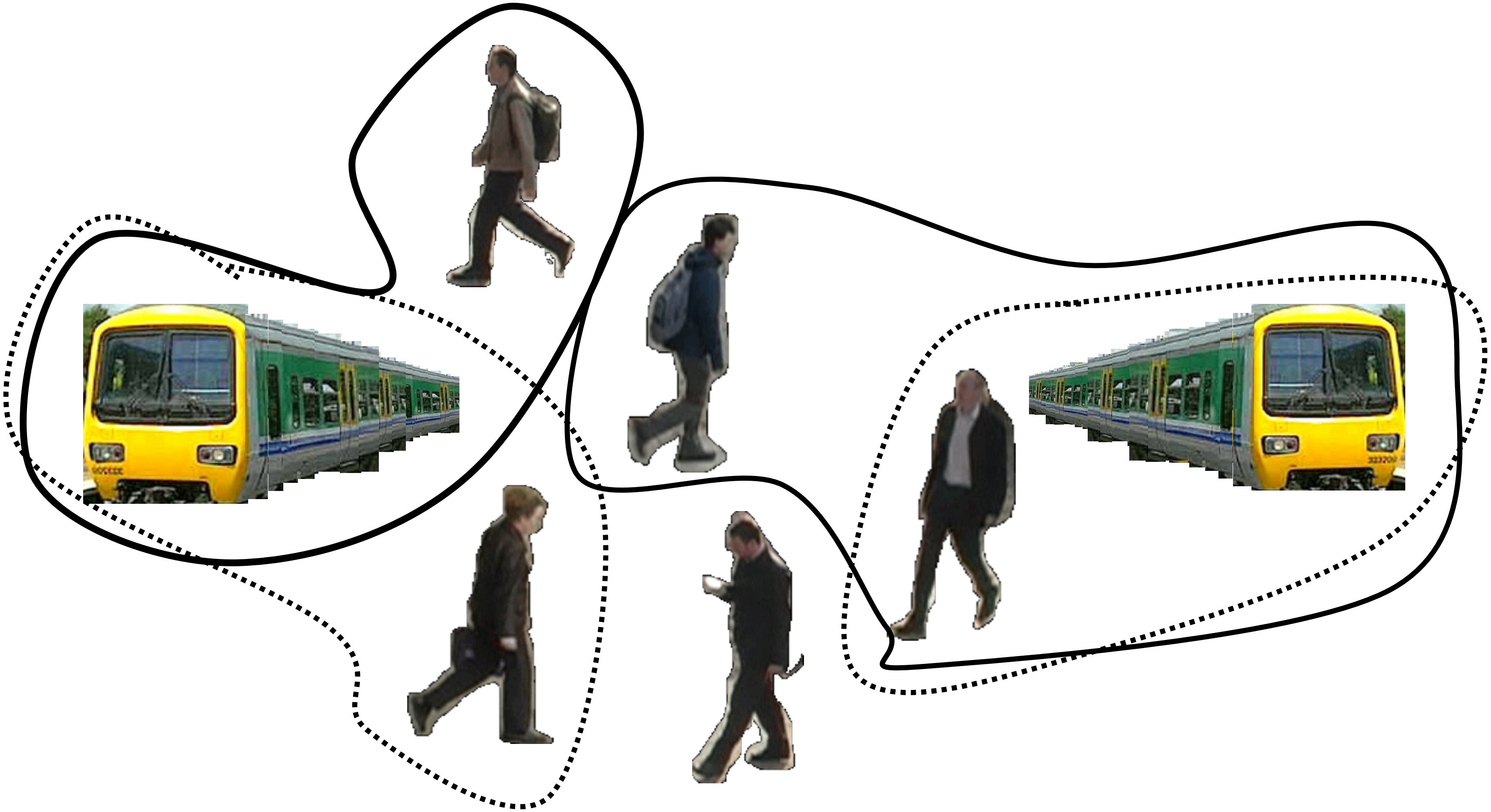
Activity Recognition

Activity Recognition

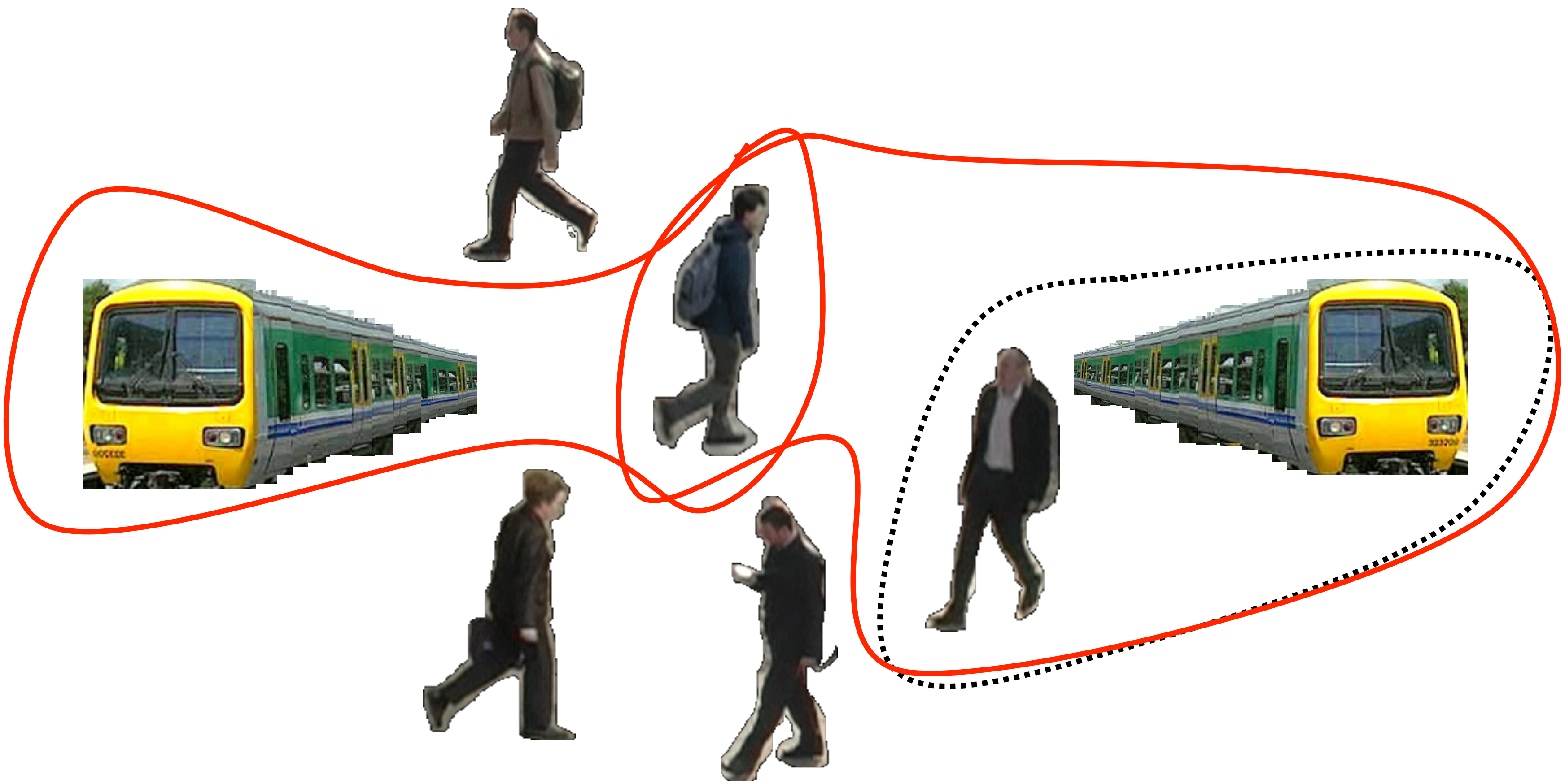
Activity Recognition



Activity Recognition

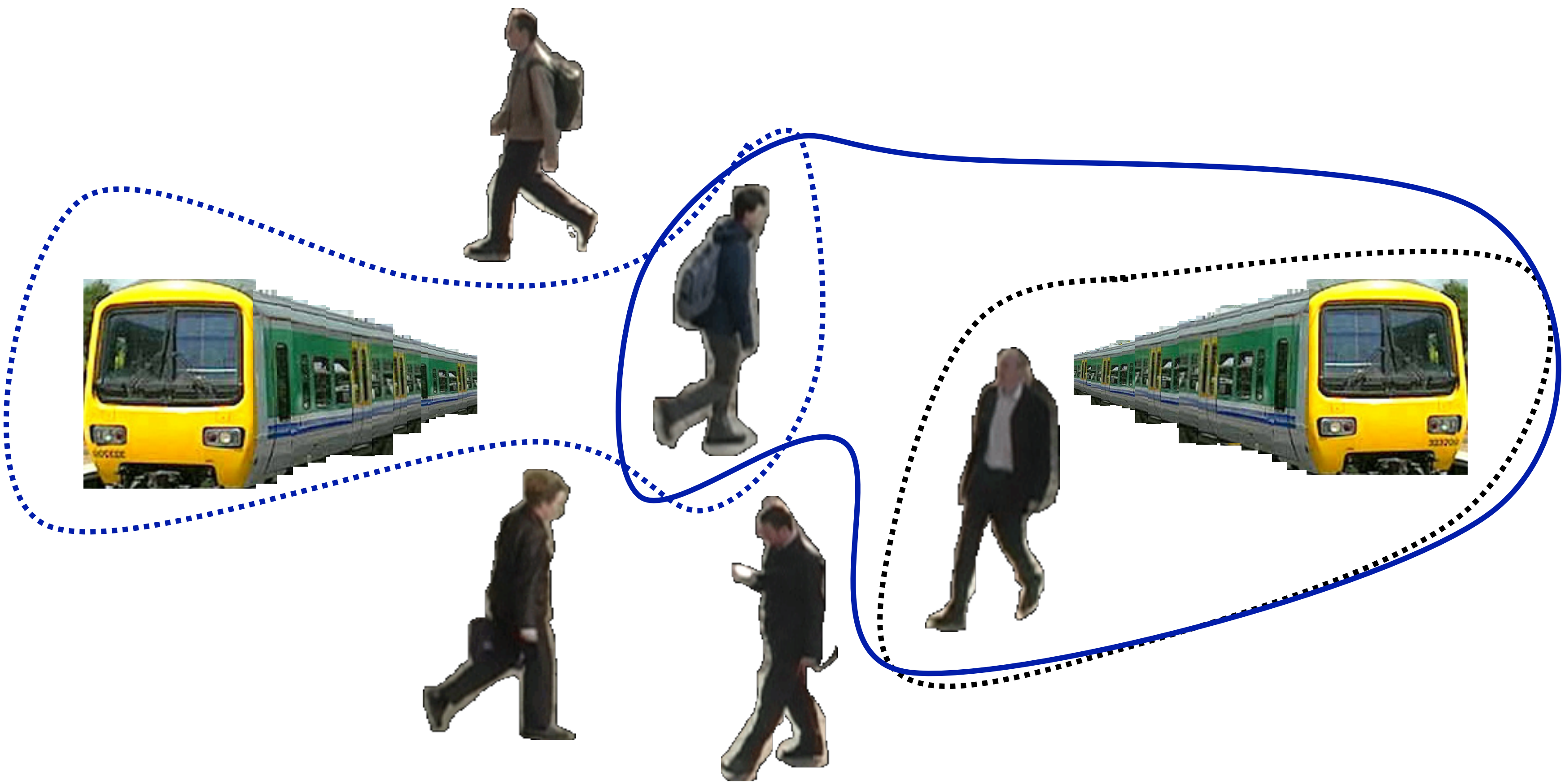


Activity Recognition



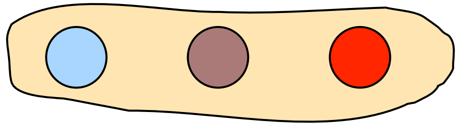
UNIVERSITY OF LEEDS

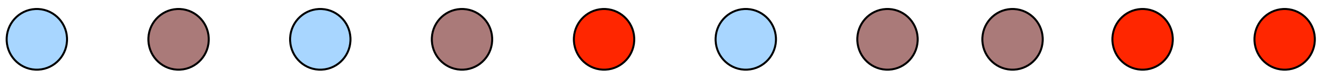
Activity Recognition



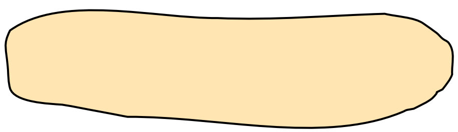
Activity Recognition

Event Definition



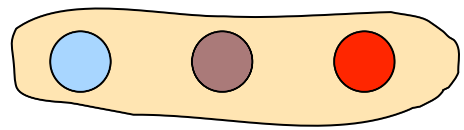


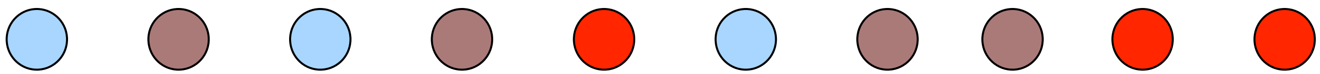
1 event



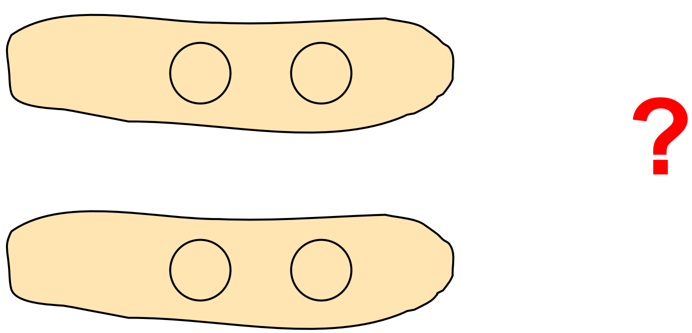
Activity Recognition

Event Definition



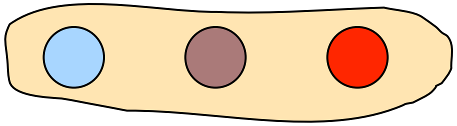


Event Threads

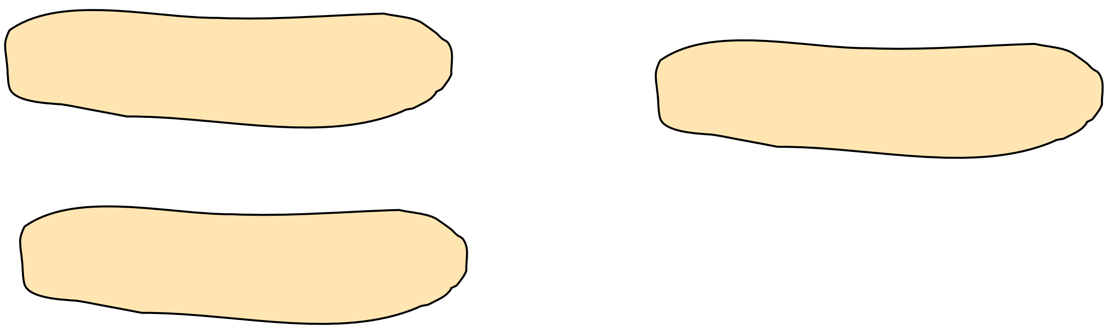


Activity Recognition

Event Definition

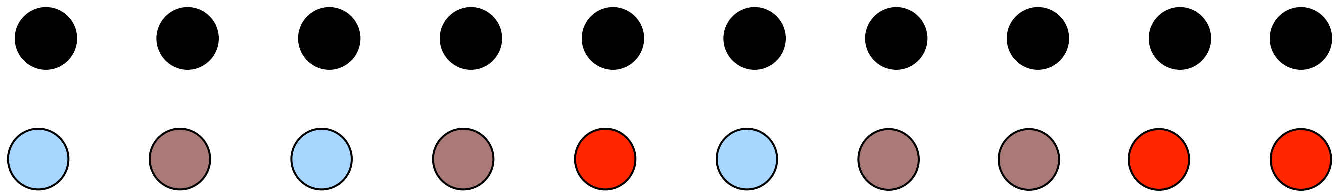


Global Explanation

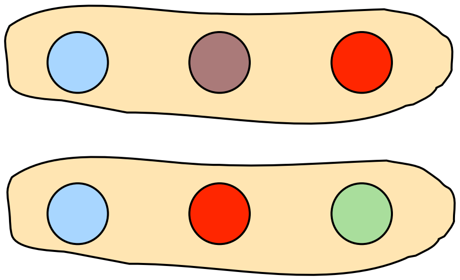


Activity Recognition

- Uncertain detections

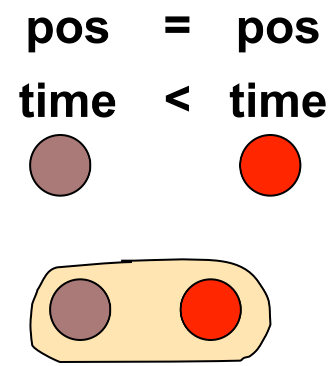


- Multiple definitions

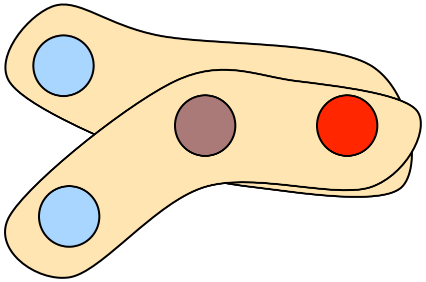


Activity Recognition

- Intra-activity constraints
  - Temporal Constraints
  - Spatial Constraints
  - Other Geometric Constraints

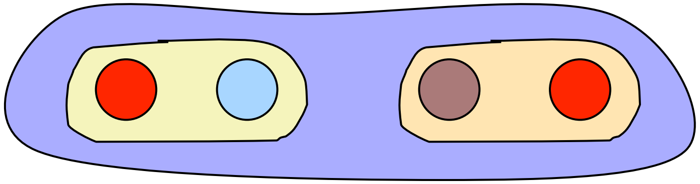


- Inter-activity constraints

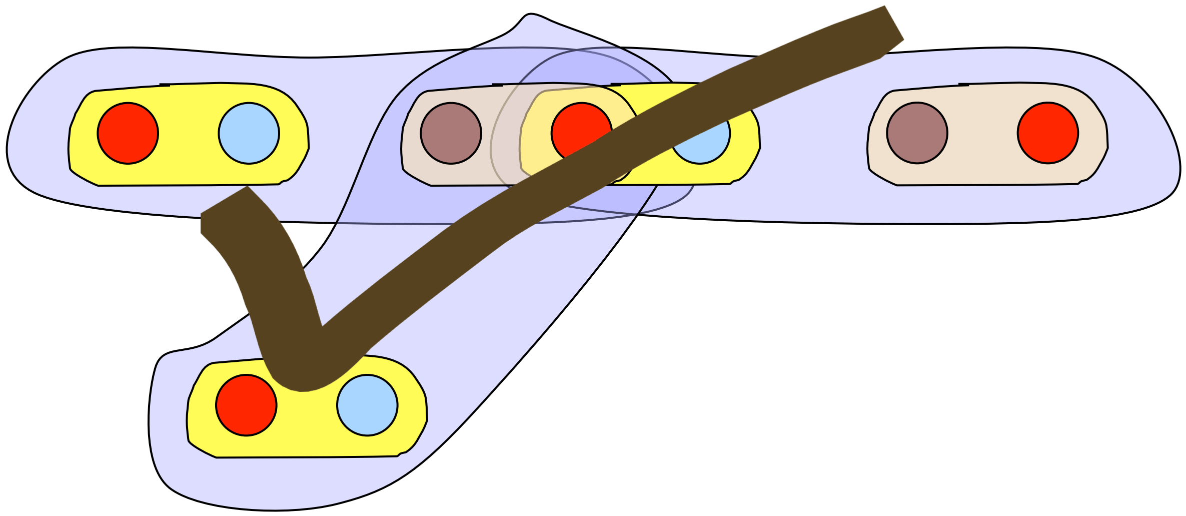


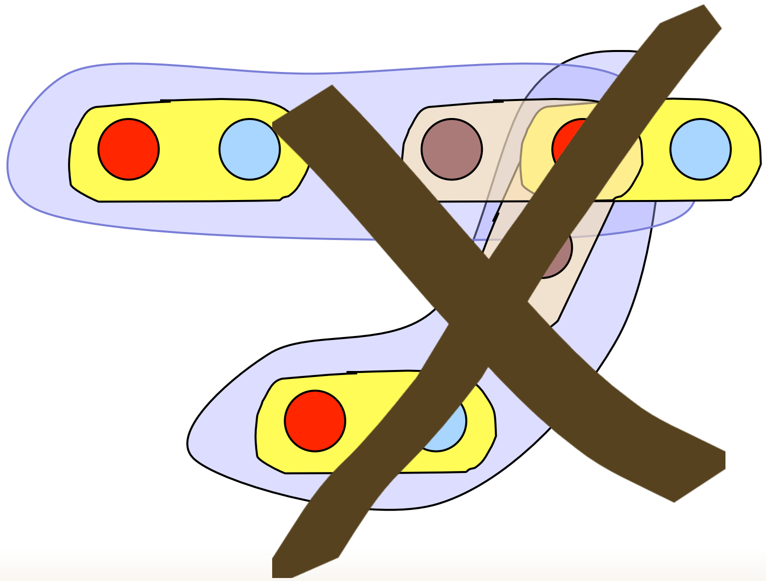
Activity Recognition

- Complex events

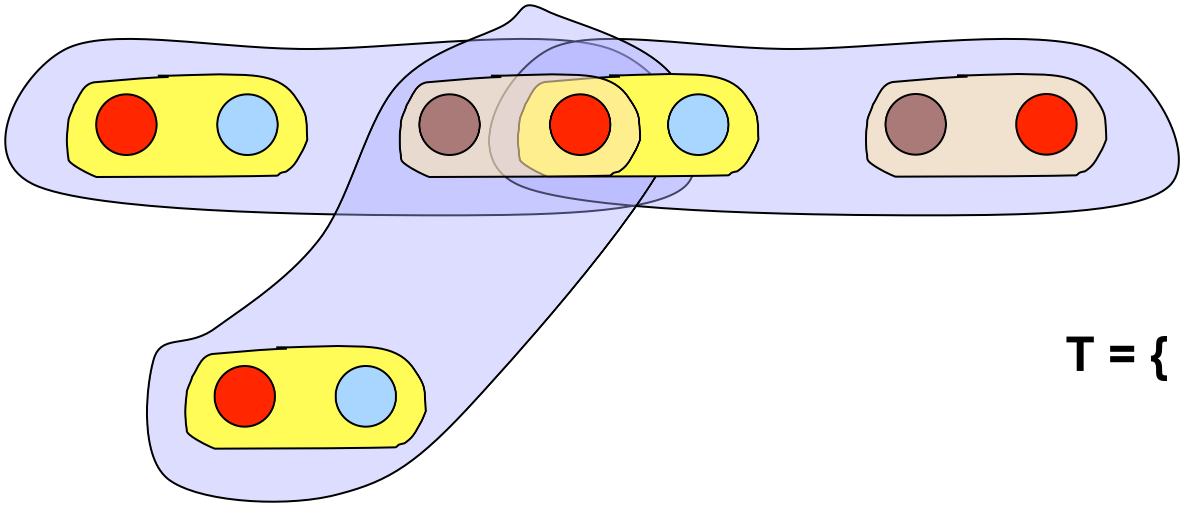


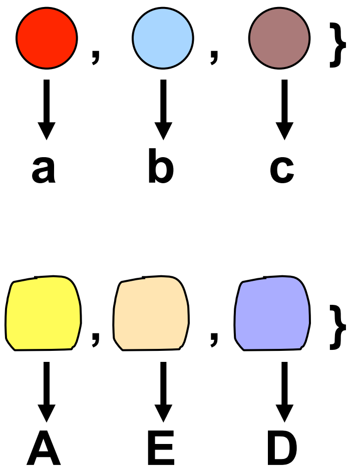
Definition using AMG





Definition using AMG



T = {  , , }

a b c

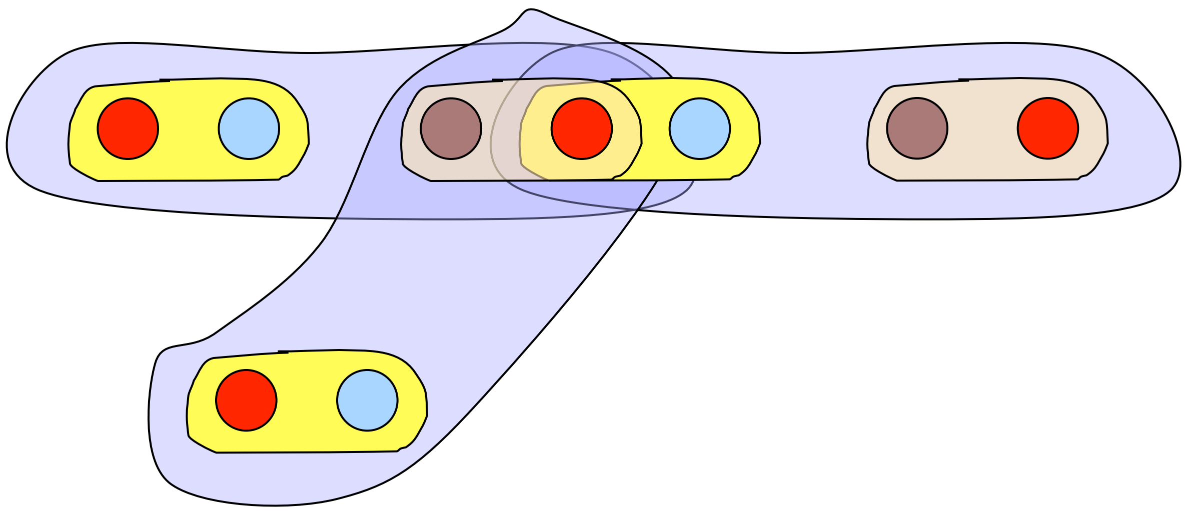
N = { , , }

A E D

Synthetic Rule | **Attribute Rule** | **Attribute Constraints**

A → a, b | **…** | **a.time < b.time**

Definition using AMG

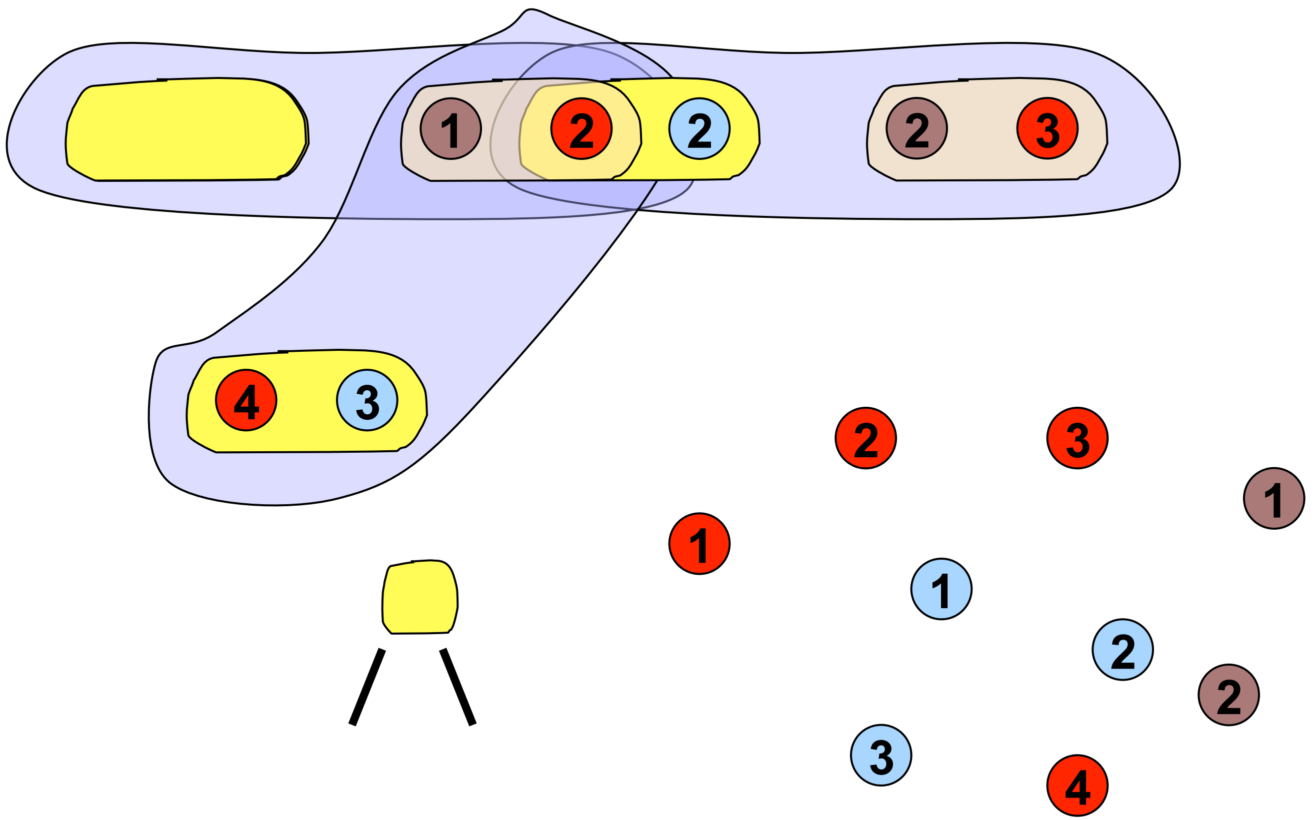


E → a, c c.count < 1

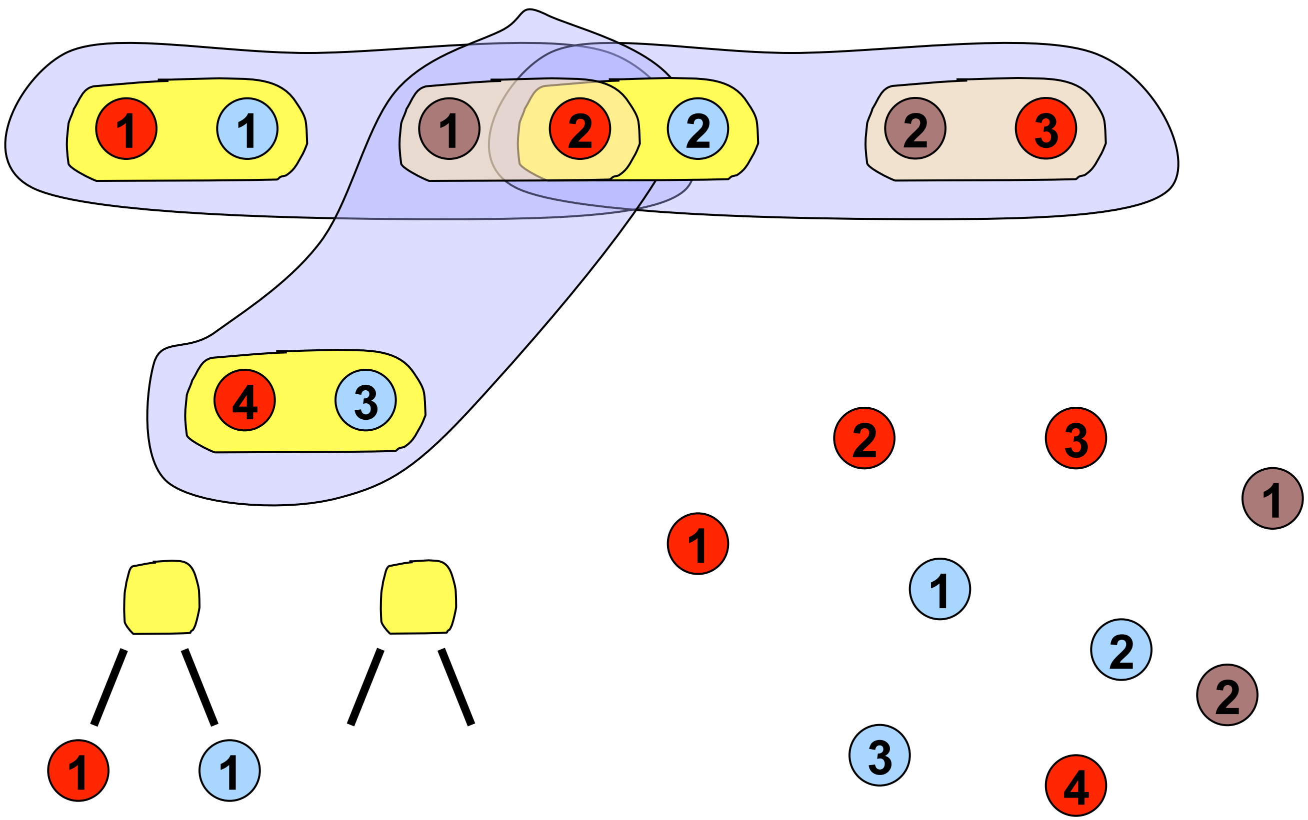
D → A, E ….

Synthetic Rule	Attribute Rule	Attribute Constraints
A → a, b	…	a.time < b.time

Definition using AMG



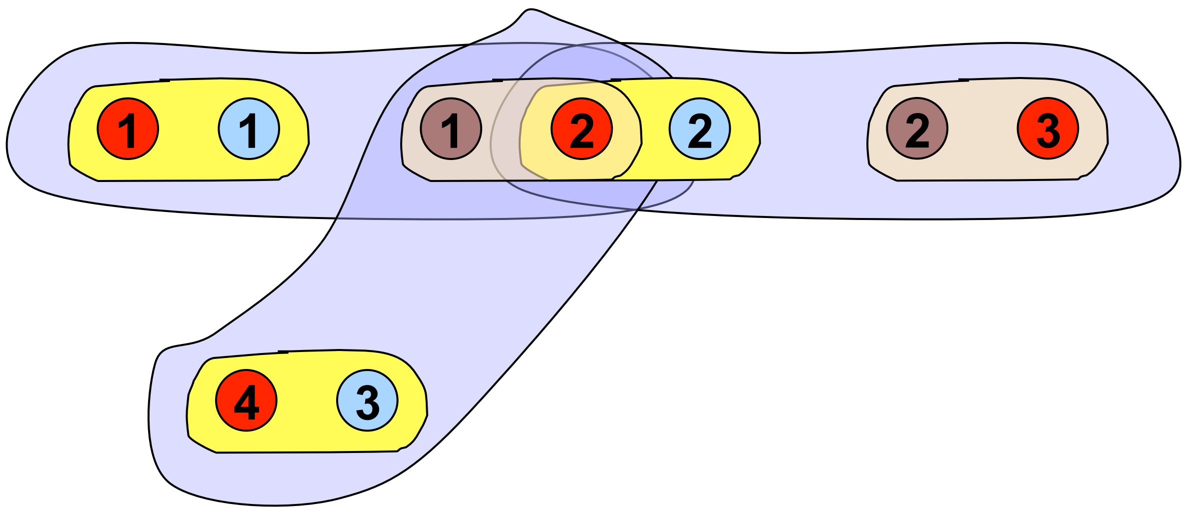
Definition using AMG

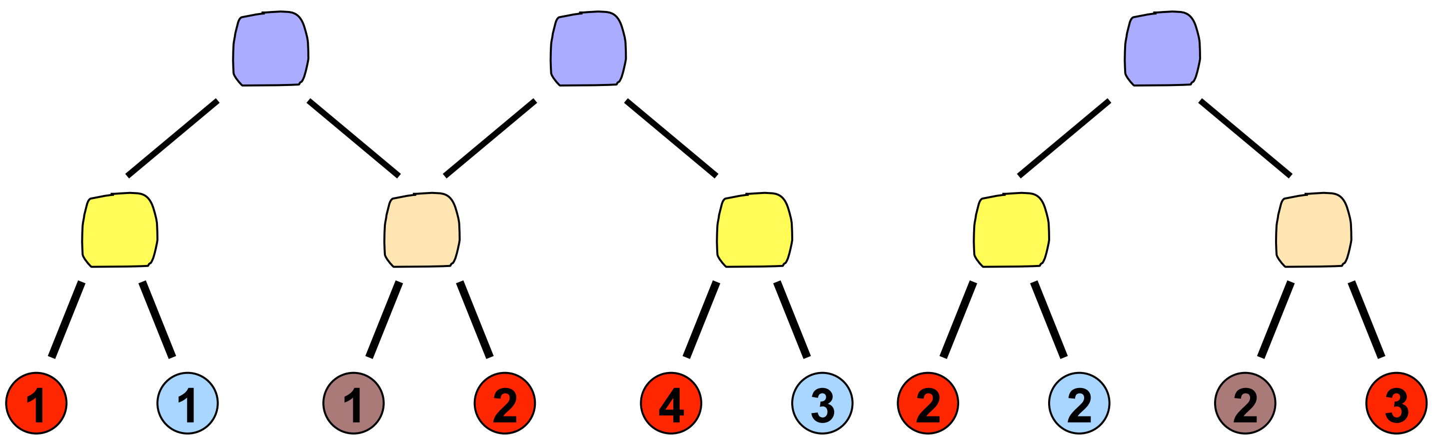


Syntactic Rule | **Attribute Rule** | **Attribute Constraints**

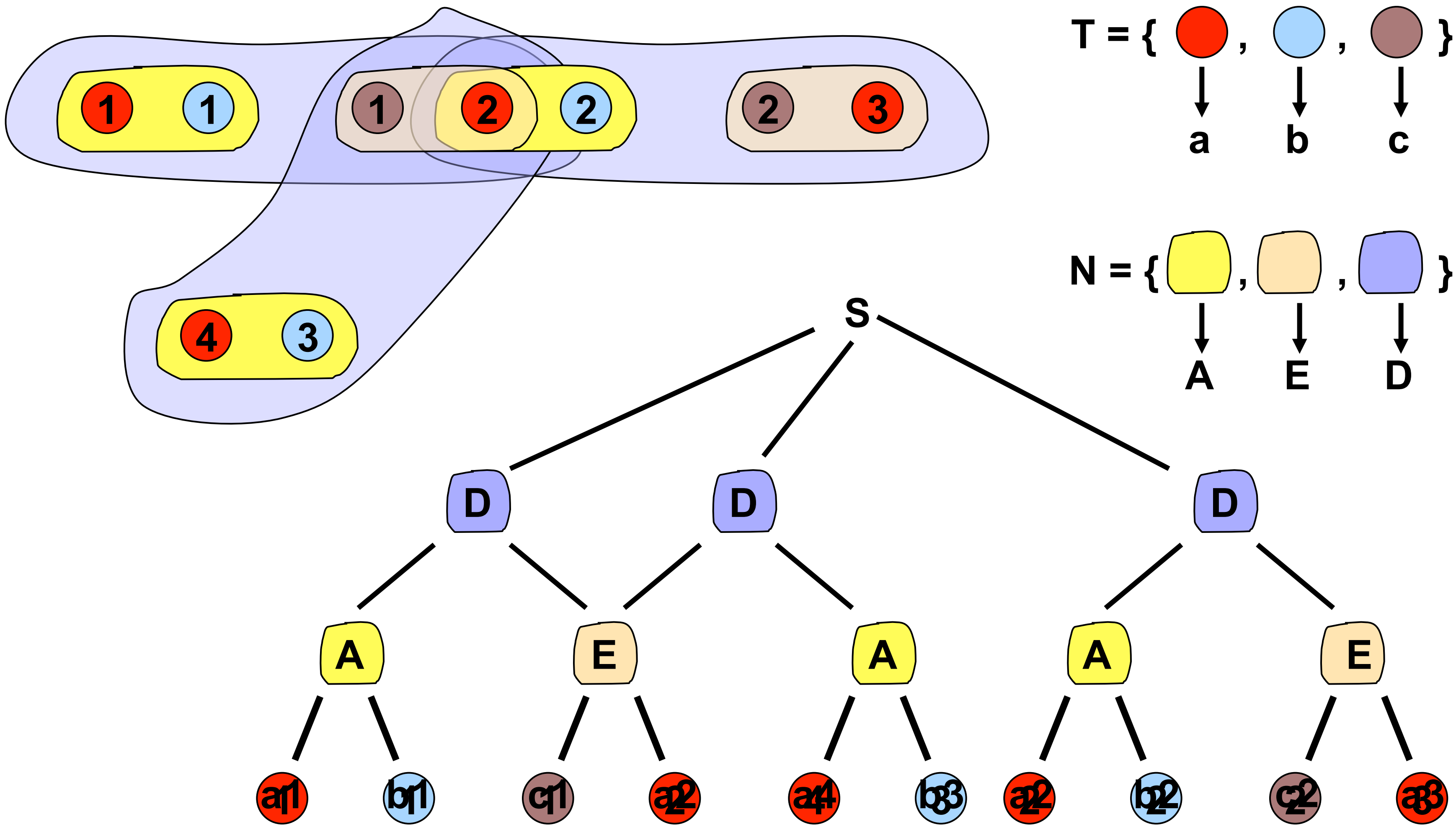
A → a, b **A.dist = a.pos – b.pos** **a.time < b.time**

Definition using AMG





Definition using AMG

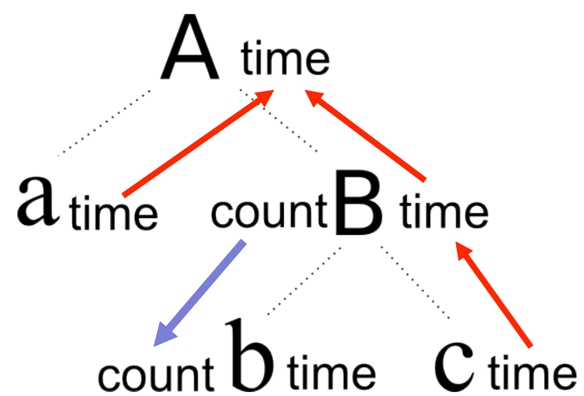


Definition using AMG

- Attribute Multiset Grammars

$$G = (N,\ T,\ S,\ A,\ P)$$

detections

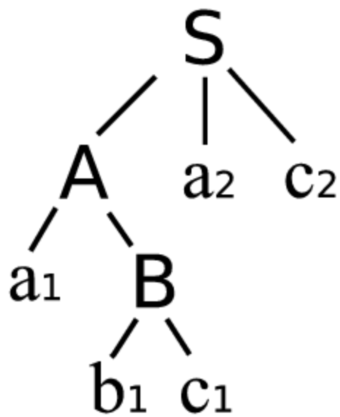


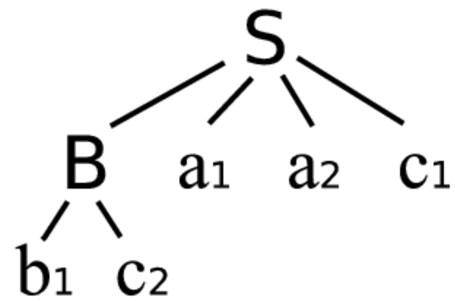
Production Rules (P):

rule	Syntactic Rule (r)			Attribute Rules (M)			Attribute Constraints (C)		
p_1	S	$\rightarrow$	$A^\star, B^\star, a^\star, c^\star$						
p_2	A	$\rightarrow$	a, B	A.time	=	a.time+B.time	a.time	$<$	B.time
				B.count	=	1	B.count	$\neq$	1
p_3	B	$\rightarrow$	b, c	B.time	=	c.time	b.time	$<$	c.time
				b.count	=	B.count	b.count	$\neq$	1

Recognition using AMG

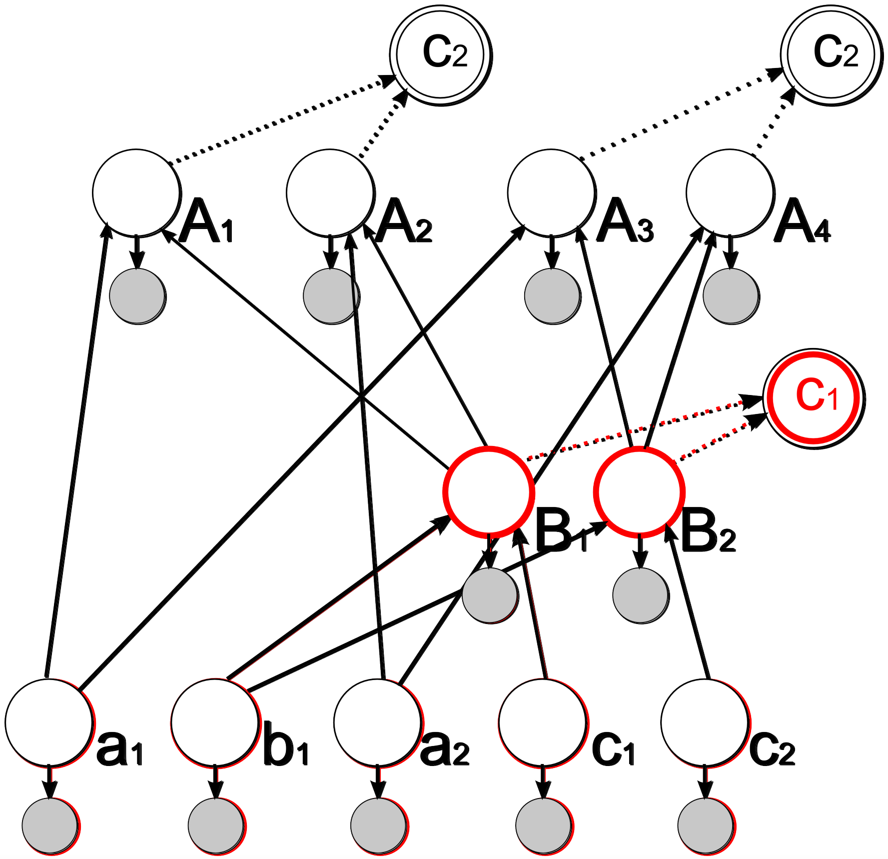
$D = \{a_1, a_2, b_1, c_1, c_2\}$





Recognition using AMG

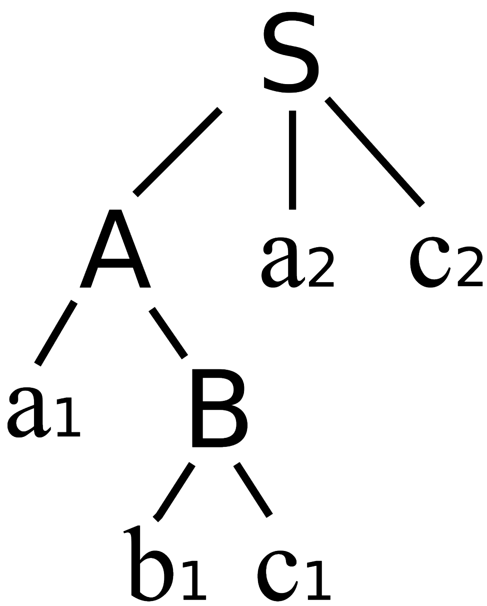
$D = \{a_1, a_2, b_1, c_1, c_2\}$

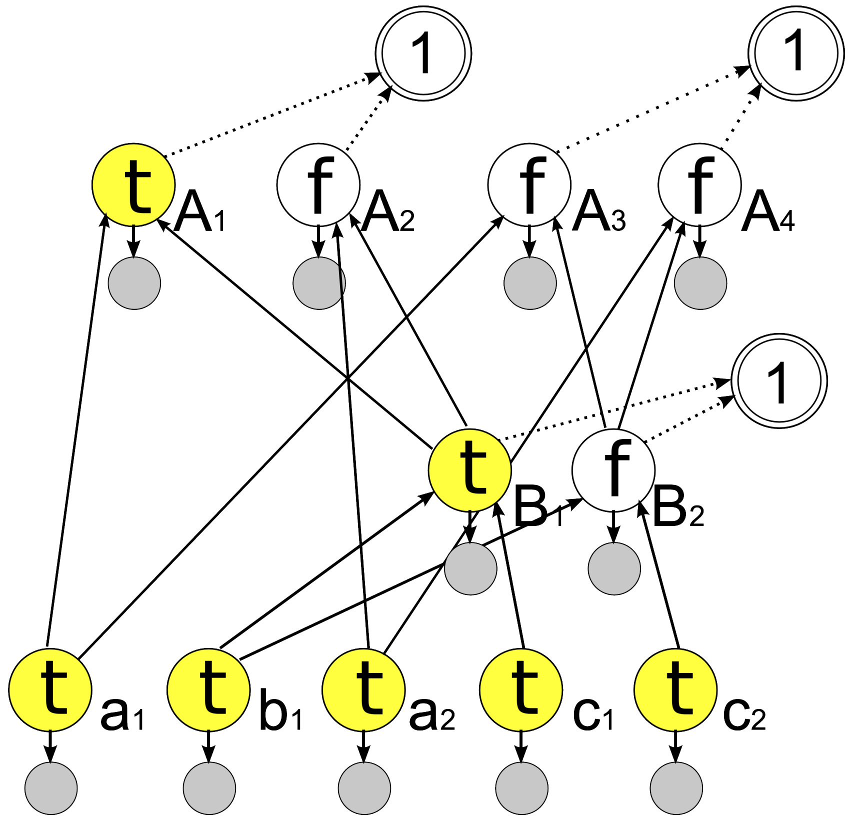


Algorithm

Recognition using AMG

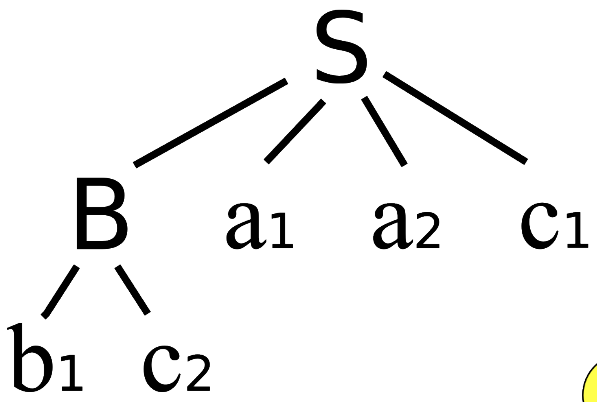
$D = \{a_1, a_2, b_1, c_1, c_2\}$

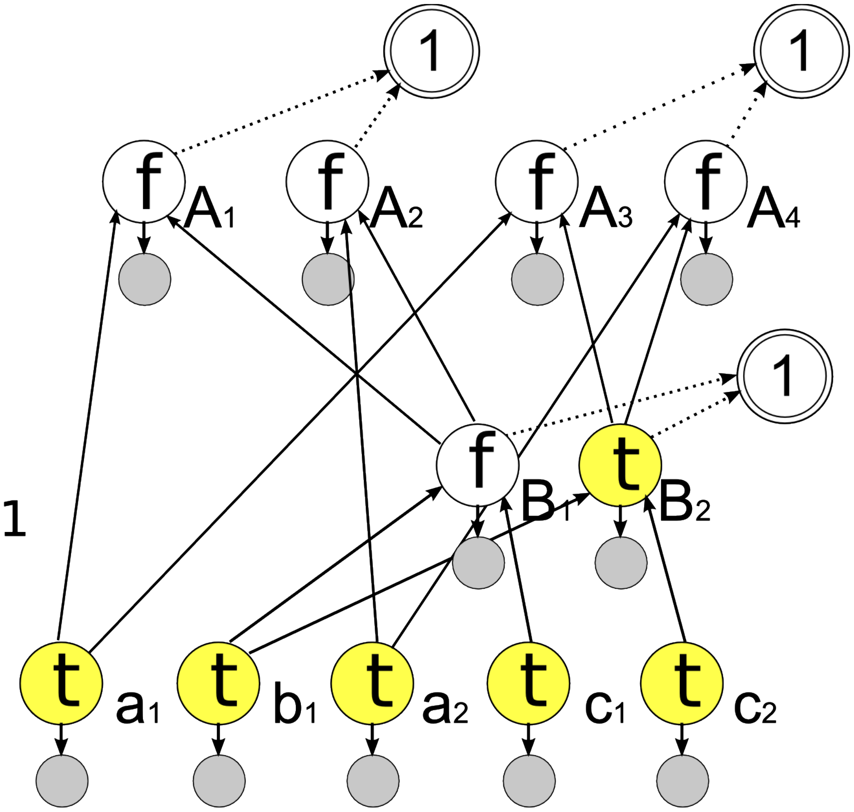




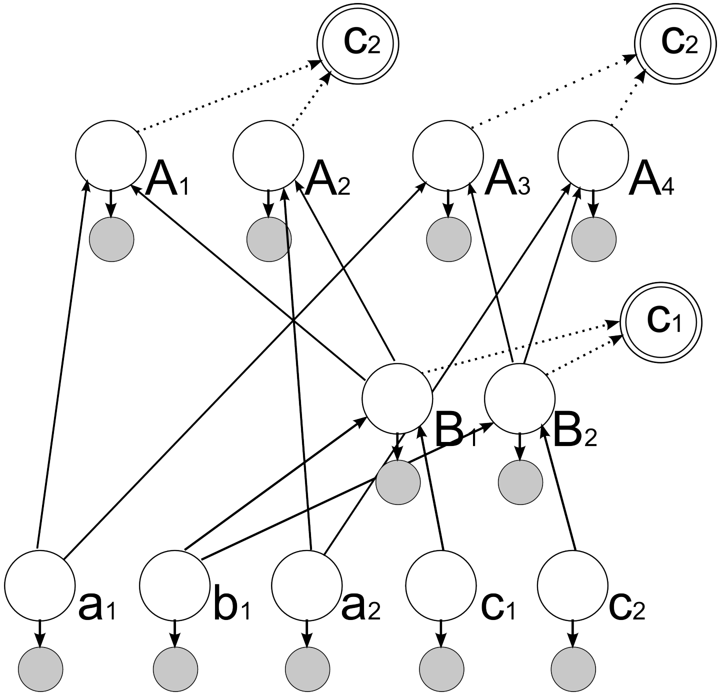
Recognition using AMG

$D = \{a_1, a_2, b_1, c_1, c_2\}$





Recognition using AMG



Searching the space of explanations

- **Greedy Search**
- **Multiple Hypotheses Tree [BMVC 07]**
- **Reversible Jump Markov Chain [CVPR 09]**
  Monte Carlo
- **<u>Integer Programming</u>**

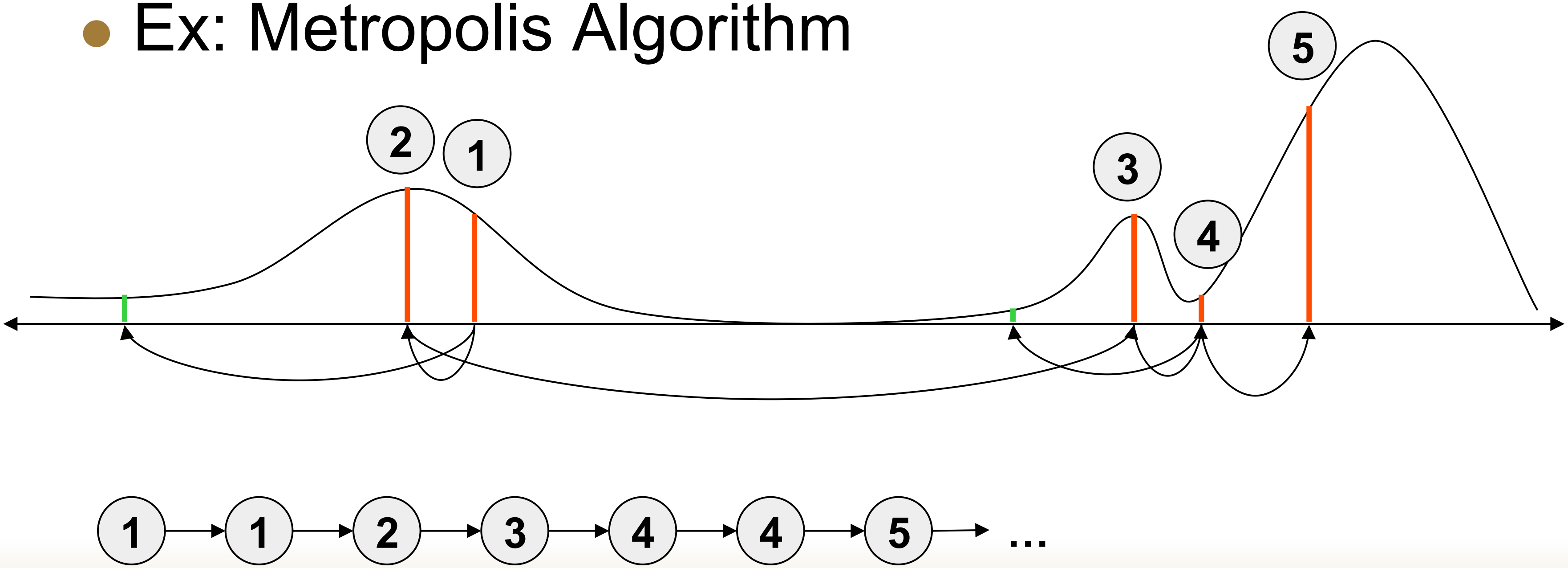
Searching the space of Explanations

$$\omega^\star = \arg\max_\omega p(\omega|Y)$$

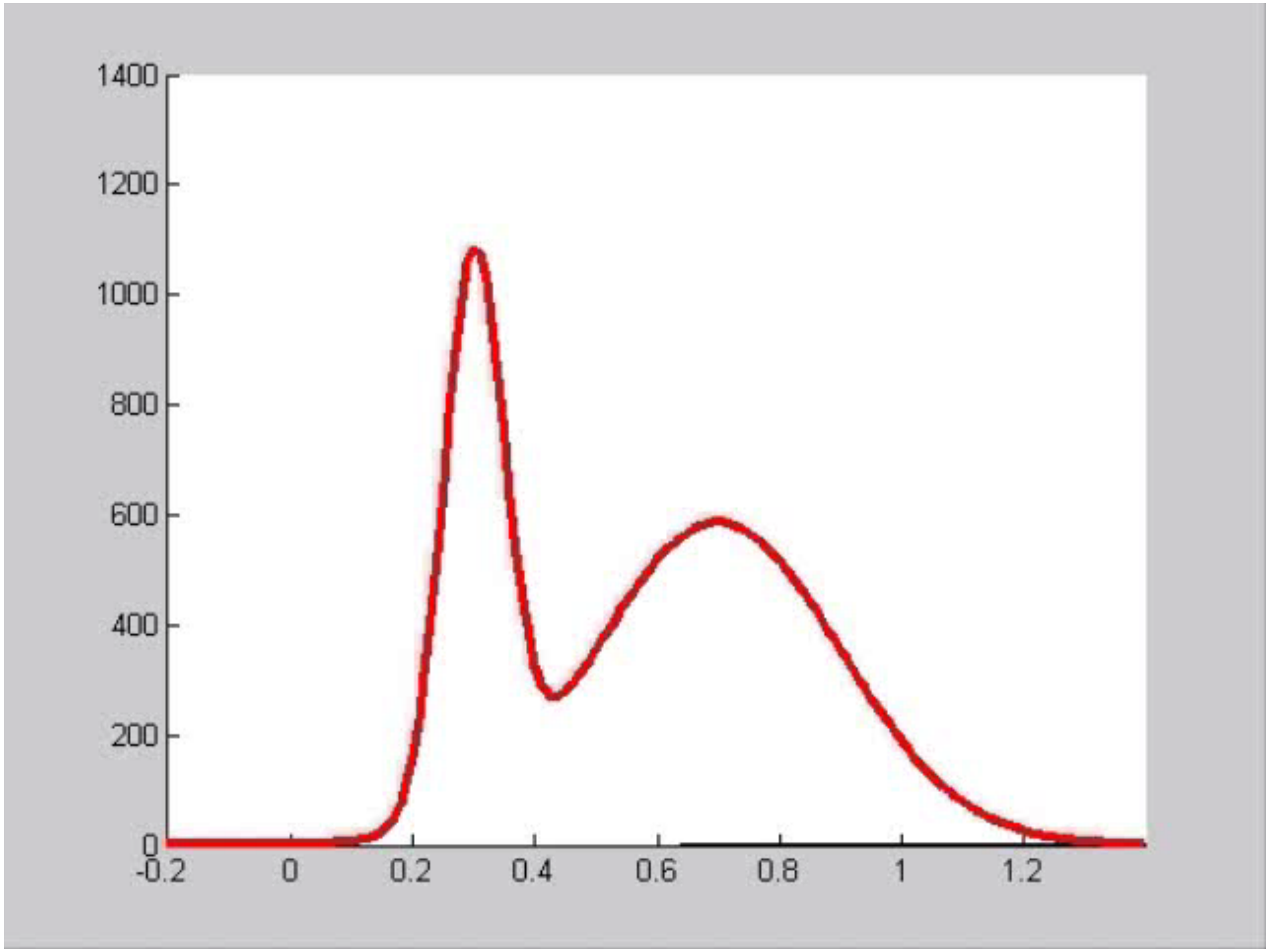
- MCMC samples the space focusing on where posterior is concentrated

Introduction to MCMC

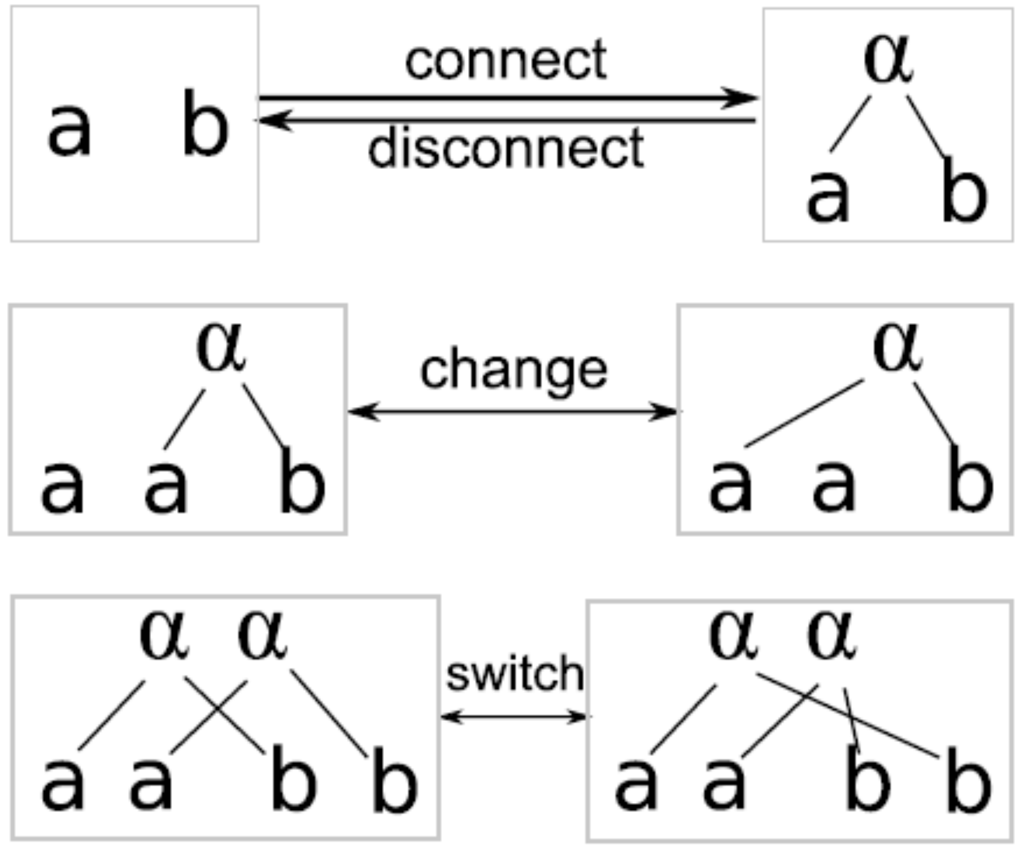
- MCMC – Markov Chain Monte Carlo
- When?
  - You can't sample from the distribution itself
  - Can evaluate it at any point
  - Ex: Metropolis Algorithm



Introduction to MCMC

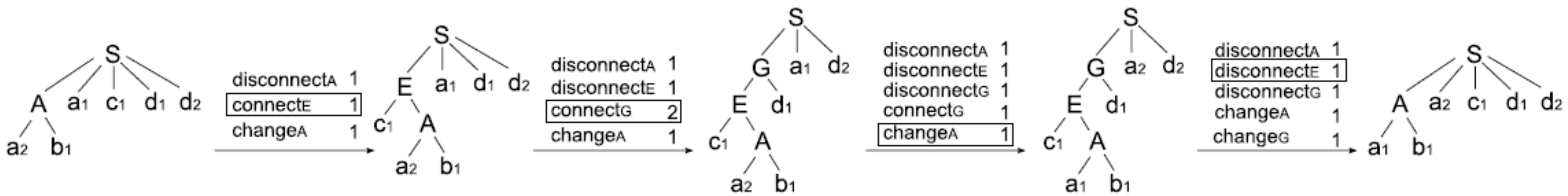


Suggested Moves



UNIVERSITY OF LEEDS

Reversible Moves



MCMC General Algorithm

Markov Chain Monte Carlo

```
initialize ω_0

for i = 1 to n_mc
  sample m from ξ_i
  sample ω* from Q_m(ω*|ω_{i-1})

  calculate α(ω*|ω_{i-1}) = (π(ω*)/π(ω)) (Q(ω|ω*)/Q(ω*|ω))
  sample u from U[0,1]
  if u < α(ω*|ω_{i-1})
    ω_i = ω*
  else
    ω_i = ω_{i-1}
```

initialize ω_0

for i = 1 to n_{mc}

sample m from ξ_i

sample $\omega^\star$ from $Q_m(\omega^\star|\omega_{i-1})$

calculate $\alpha(\omega^\star|\omega_{i-1}) = \left(\frac{\pi(\omega^\star)}{\pi(\omega)}\right) \quad \frac{Q(\omega|\omega^\star)}{Q(\omega^\star|\omega)}$

sample u from $\mathscr{U}[0,1]$

if u < $\alpha(\omega^\star|\omega_{i-1})$

$\omega_i = \omega^\star$

else

$\omega_i = \omega_{i-1}$

MCMC General Algorithm

```
                              Markov Chain Monte Carlo
initialize ω_0


for i = 1 to n_mc
  sample m from ξ_i
  sample ω* from Q_m(ω*|ω_{i-1})

  calculate α(ω*|ω_{i-1}) = (π(ω*)/π(ω))  Q(ω|ω*)/Q(ω*|ω)
  sample u from U[0,1]
  if u < α(ω*|ω_{i-1})
    ω_i = ω*
  else
    ω_i = ω_{i-1}
```

Markov Chain Monte Carlo

initialize ω_0

for i = 1 to n_{mc}

  sample m from ξ_i

  sample $\omega^\star$ from $Q_m(\omega^\star|\omega_{i-1})$

  calculate $\alpha(\omega^\star|\omega_{i-1}) = \left(\frac{\pi(\omega^\star)}{\pi(\omega)}\right) \quad \frac{Q(\omega|\omega^\star)}{Q(\omega^\star|\omega)}$

  sample u from $\mathscr{U}[0,1]$

  if u < $\alpha(\omega^\star|\omega_{i-1})$

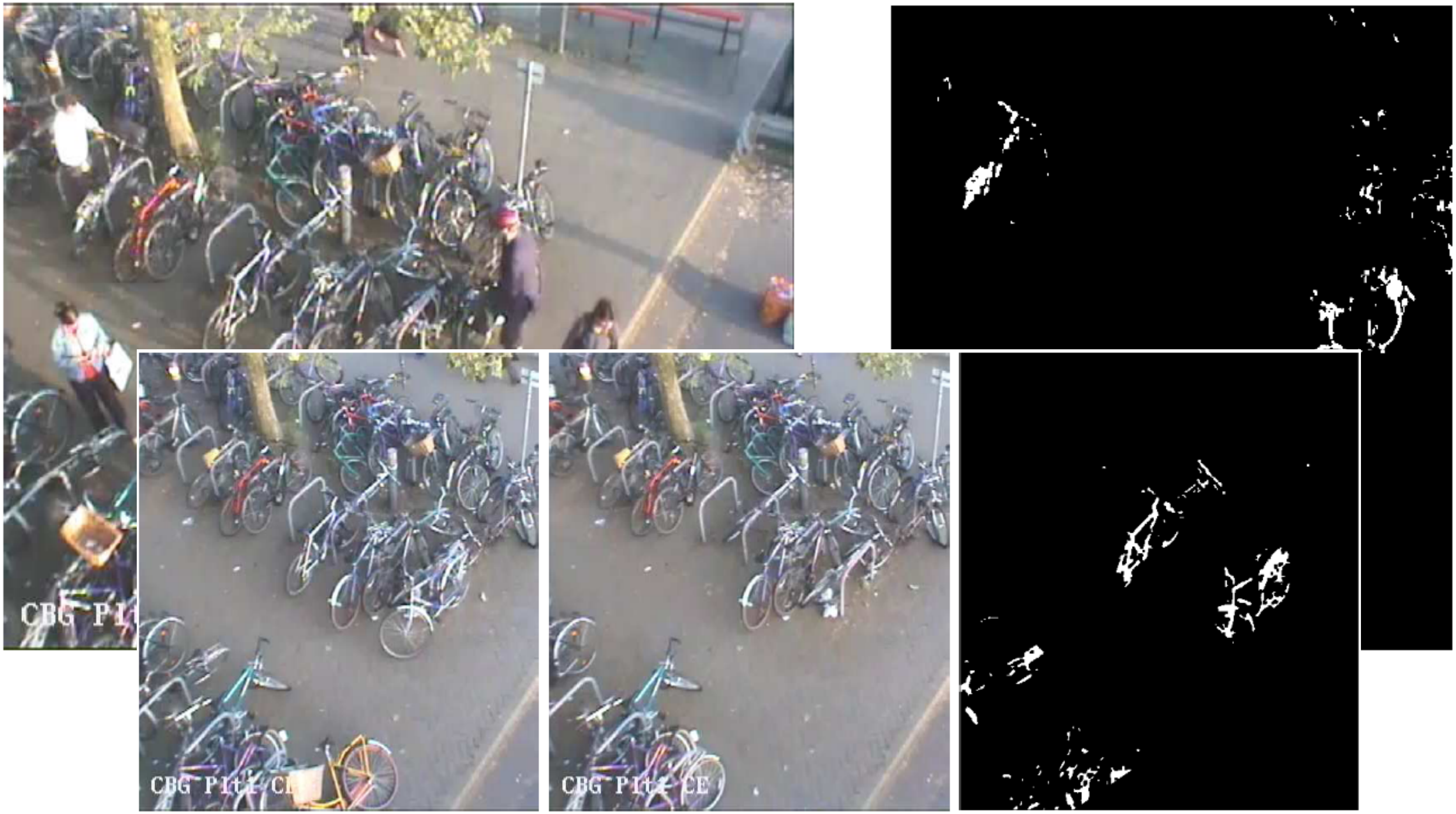
    $\omega_i = \omega^\star$

  else

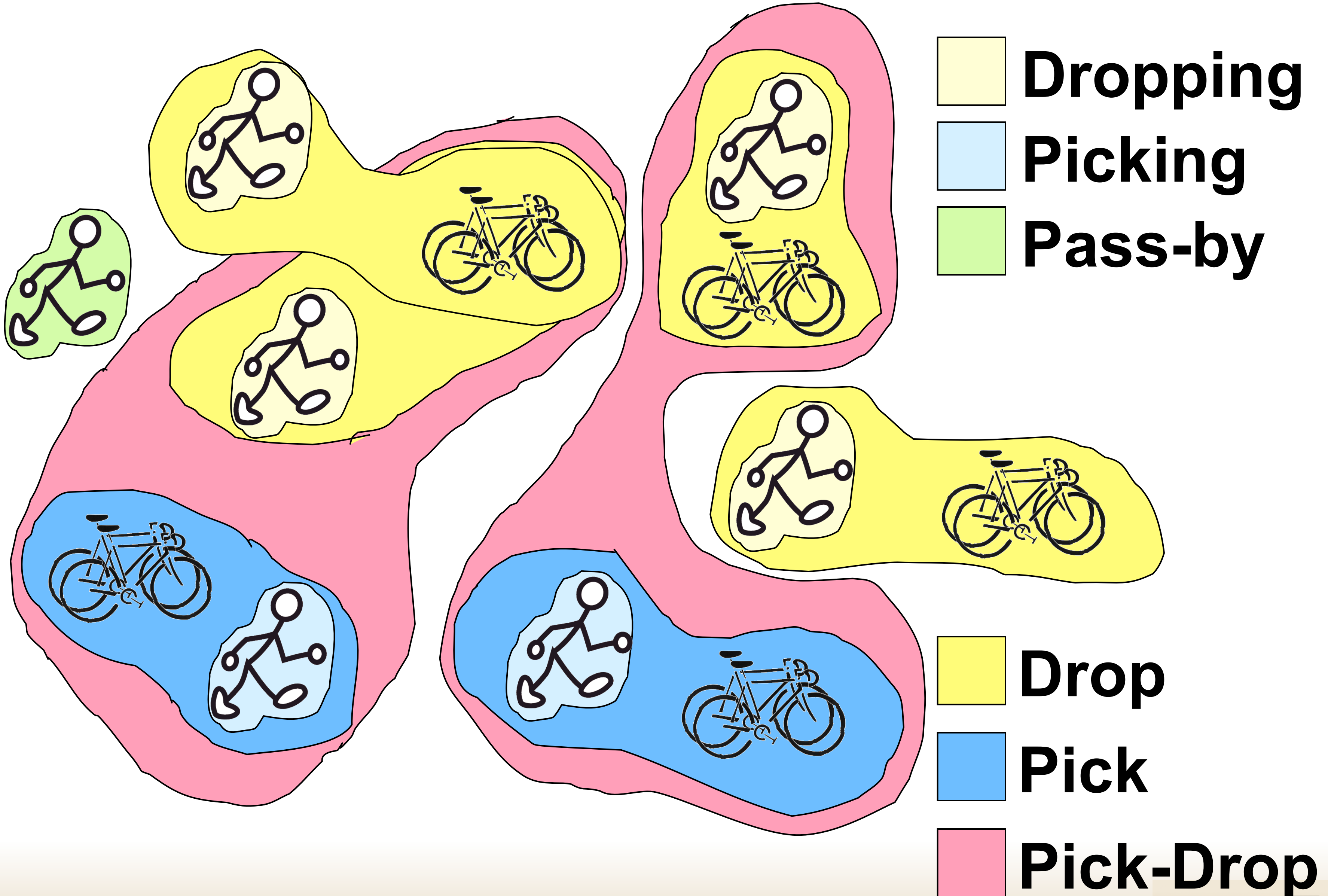
    $\omega_i = \omega_{i-1}$

Case I : The Bicycles Problem

The Bicycles Problem



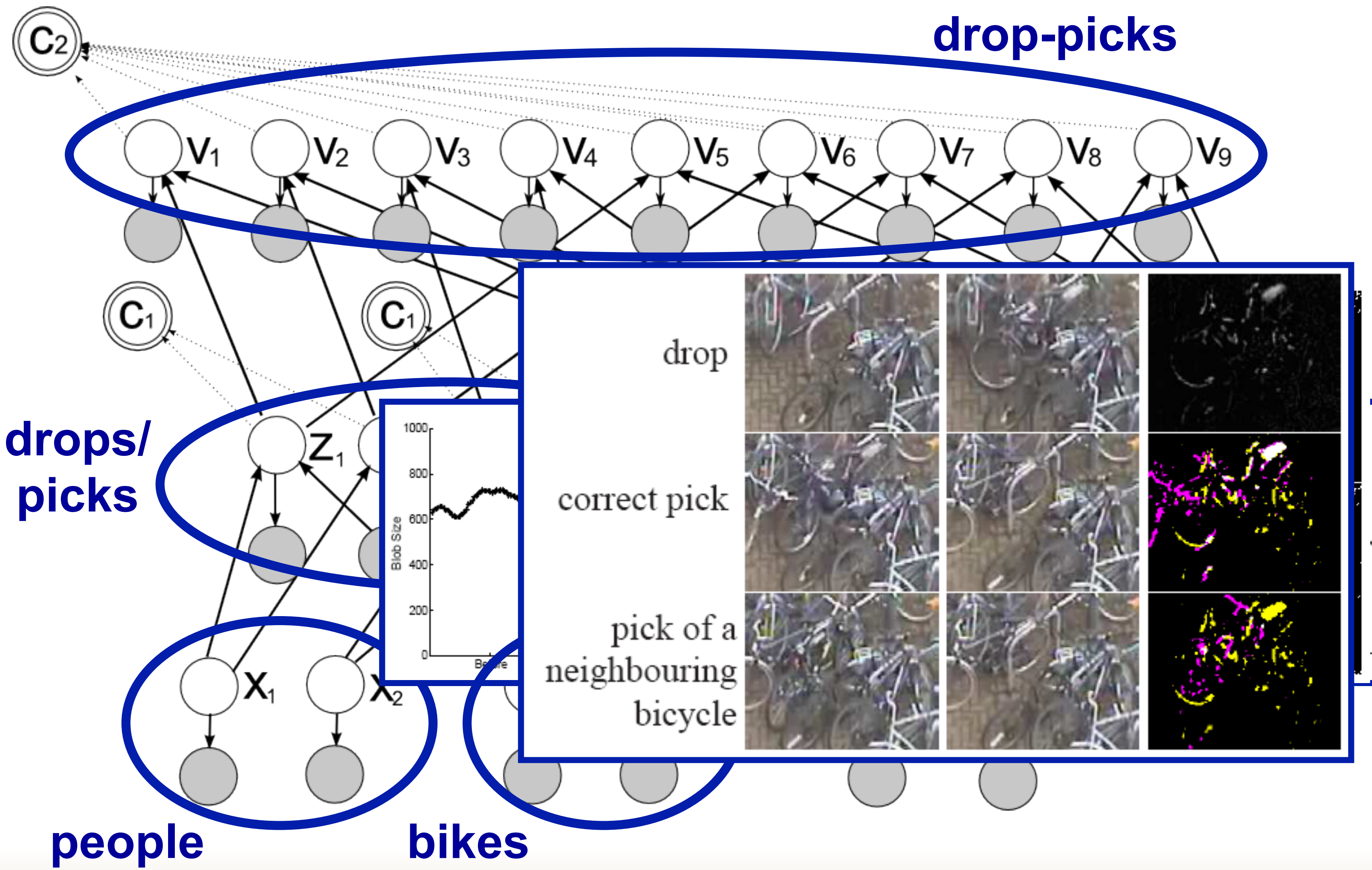
The Bicycles Problem



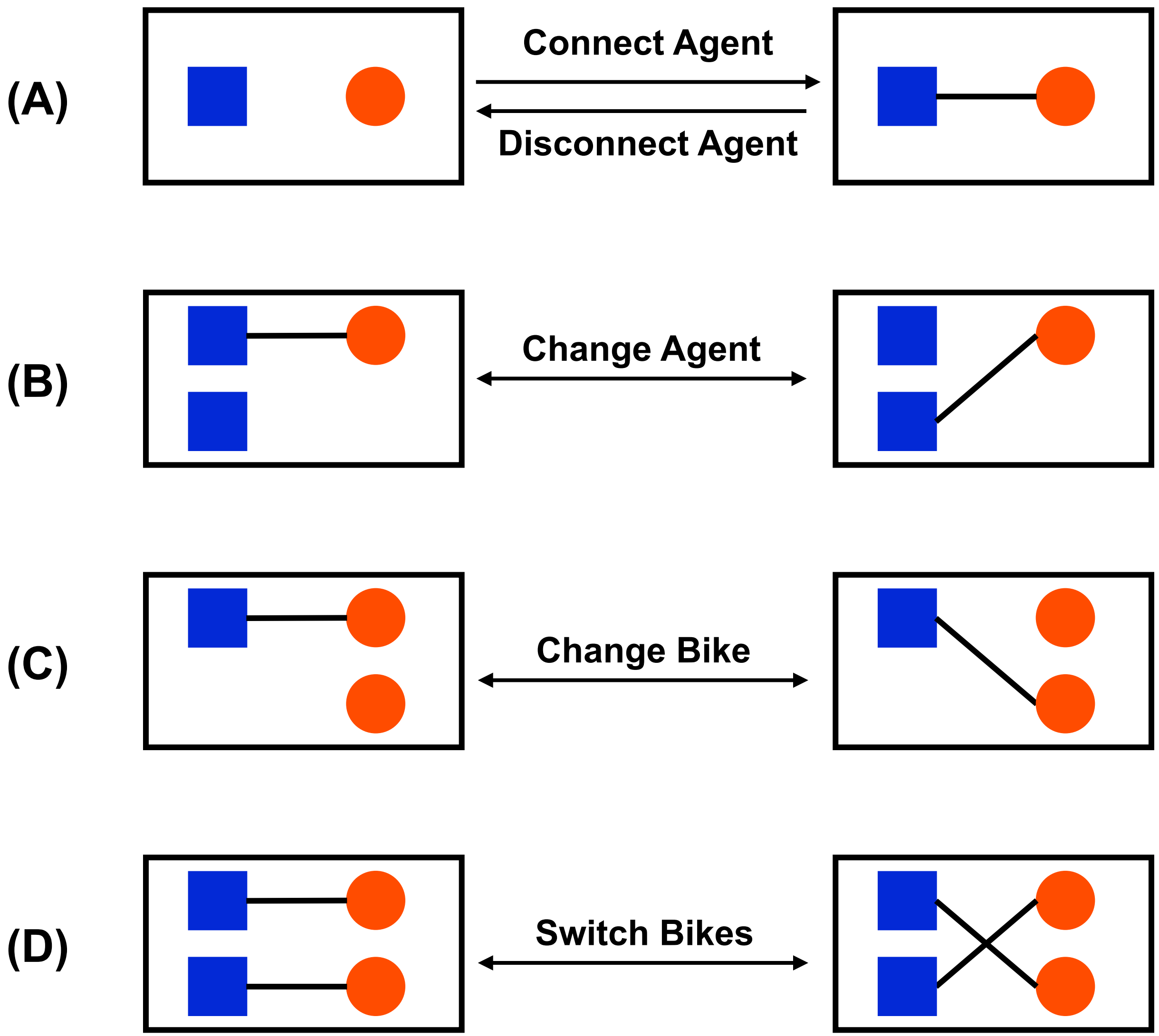
The Bicycles AMG

	Syntactic Rule (r)			Attribute Rules (M)			Attribute Constraints (C)		
p_1	S	$\rightarrow$	$V^\star, x^\star, y^\star$	y.action	=	"noise"	y.count	$<$	1
				x.action	=	"pass-by"	x.count	$\neq$	1
p_2	V	$\rightarrow$	Z_1, Z_2	V.action	=	"drop-pick"	Z_1.au	$<$	Z_2.au
				Z_1.action	=	"drop"	Z_1.count	$\neq$	1
				Z_2.action	=	"pick"	Z_2.count	$\neq$	1
				V.match	=	ψ_V (Z_1.pos, Z_2.pos)			
				Z_1.count	=	Z_2.count = 1			
p_3	V	$\rightarrow$	Z, u	V.action	=	"drop-only"	Z.count	$\neq$	1
				Z.action	=	"drop"			
				Z.count	=	1			
p_4	V	$\rightarrow$	u, Z	V.action	=	"pick-only"	Z.count	$\neq$	1
				Z.action	=	"pick"			
				Z.count	=	1			
p_5	Z	$\rightarrow$	x, y	x.action	=	Z.action	x.au	=	y.au
				y.action	=	Z.action	x.count	$\neq$	1
				Z.au	=	x.au			
				Z.pos	=	y.pos			
				Z.match	=	ψ_Z (x.traj, y.pos)			
				x.count	=	1			
				y.count	=	y.count+1			

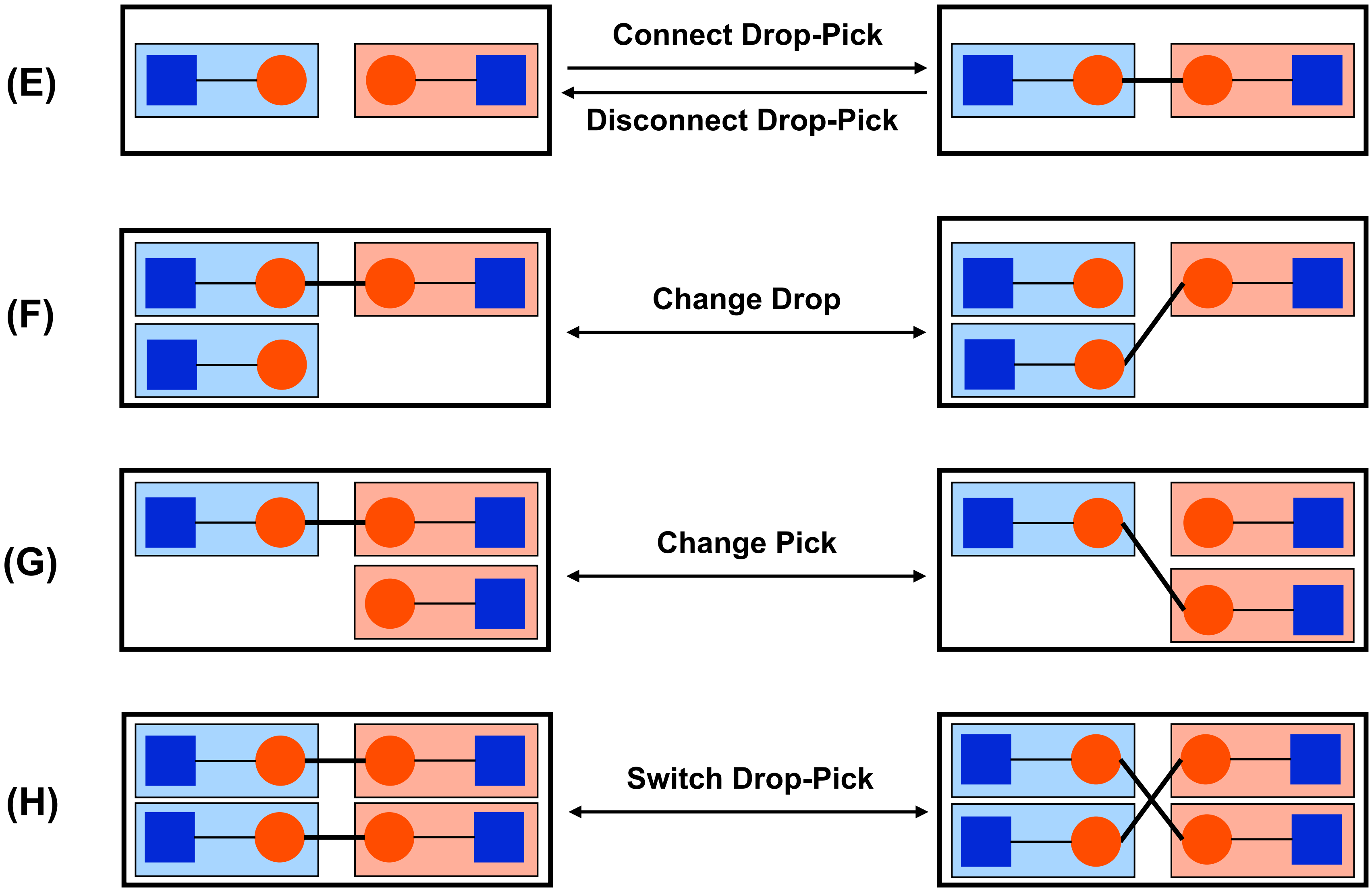
The Bicycles BN



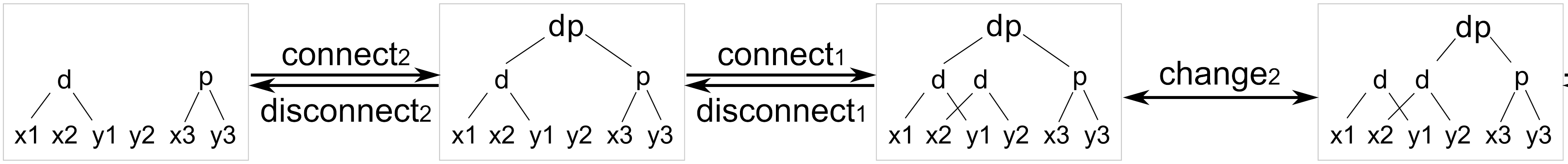
Suggested Moves – Bicycles 1



Suggested Moves – Bicycles 2

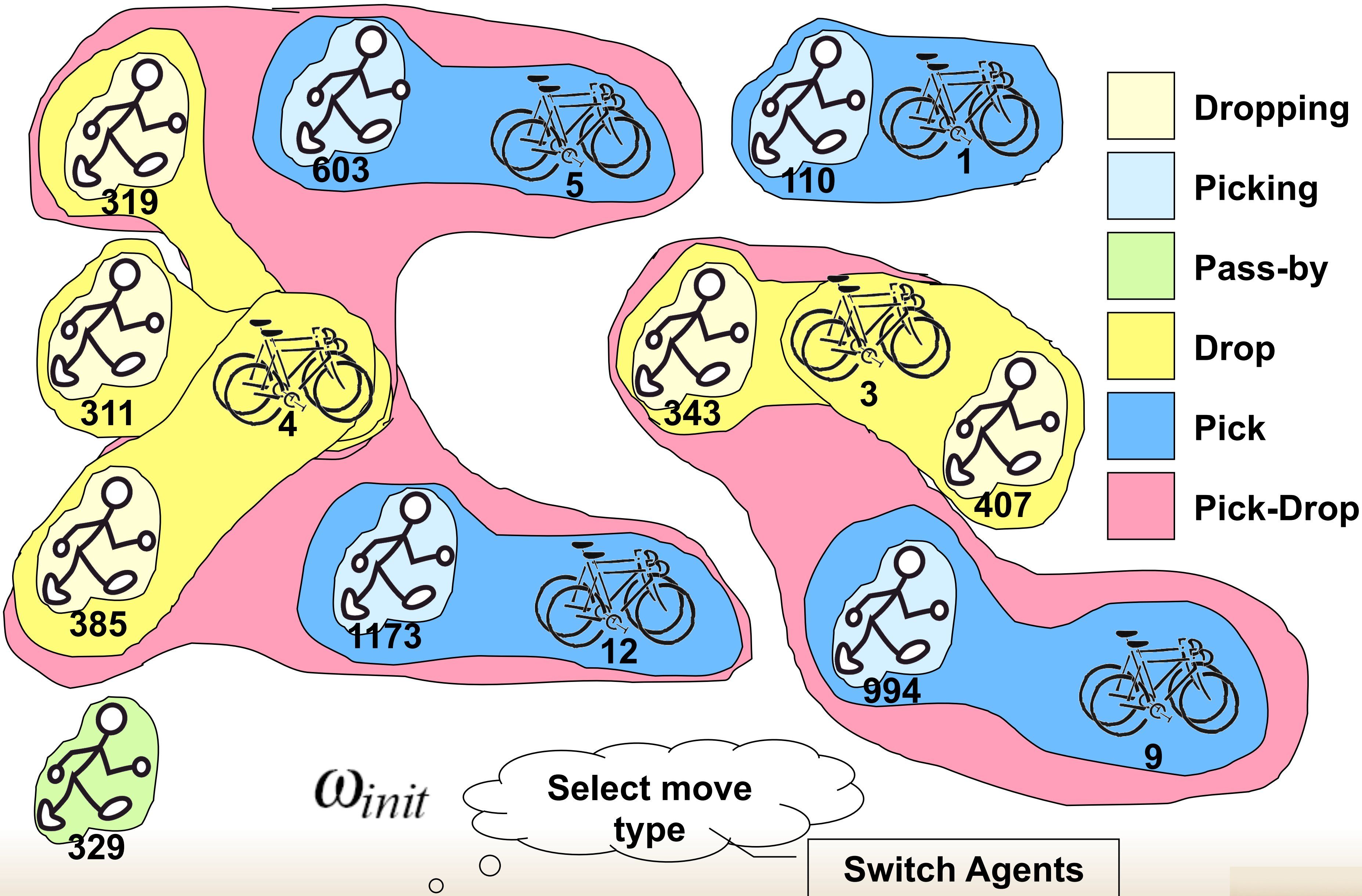


RJMCMC

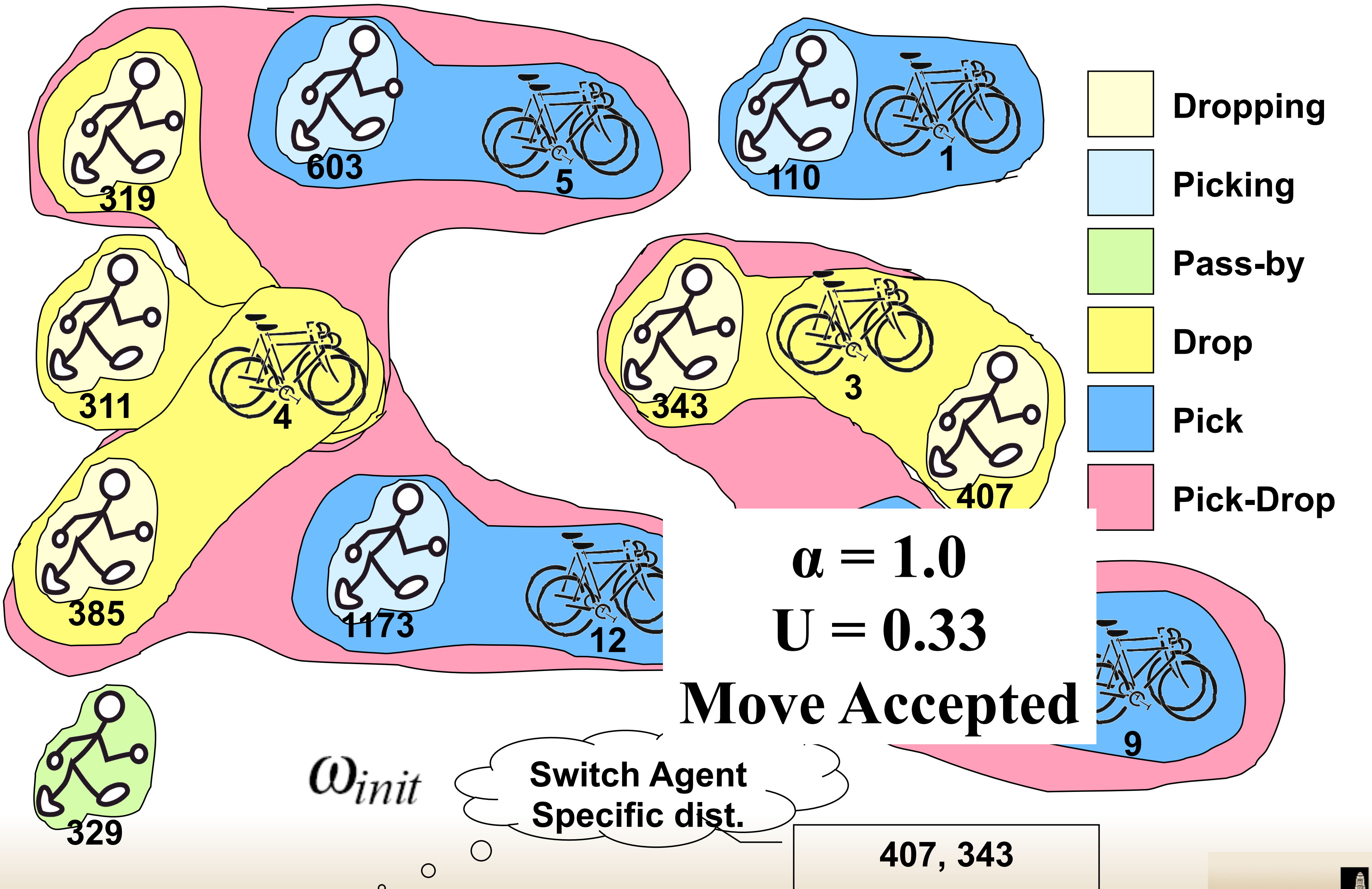


d
p
x1 x2 y1 y2 x3 y3
connect2
disconnect2
dp
d
p
x1 x2 y1 y2 x3 y3
connect1
disconnect1
dp
d d
p
x1 x2 y1 y2 x3 y3
change2
dp
d d
p
x1 x2 y1 y2 x3 y3

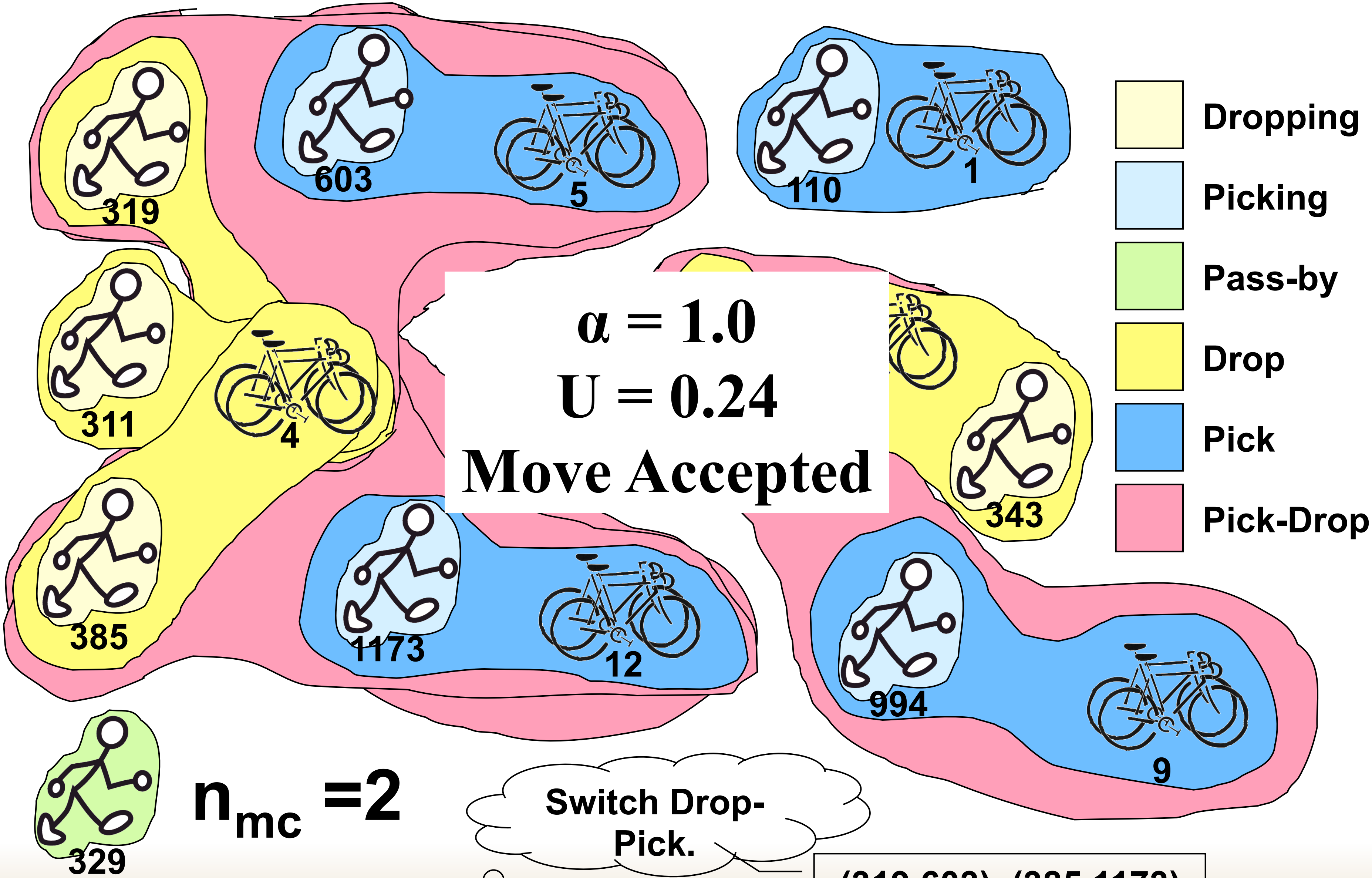
Examples


319
603
5
110
1
311
4
343
3
407
385
1173
12
994
9
329
Dropping
Picking
Pass-by
Drop
Pick
Pick-Drop
ω_{init}
Select move type
Switch Agents

Examples


319
603
5
110
1
311
4
343
3
407
385
1173
12
9
329
Dropping
Picking
Pass-by
Drop
Pick
Pick-Drop
$\alpha = 1.0$
$U = 0.33$
Move Accepted
ω_{init}
Switch Agent
Specific dist.
407, 343

Examples



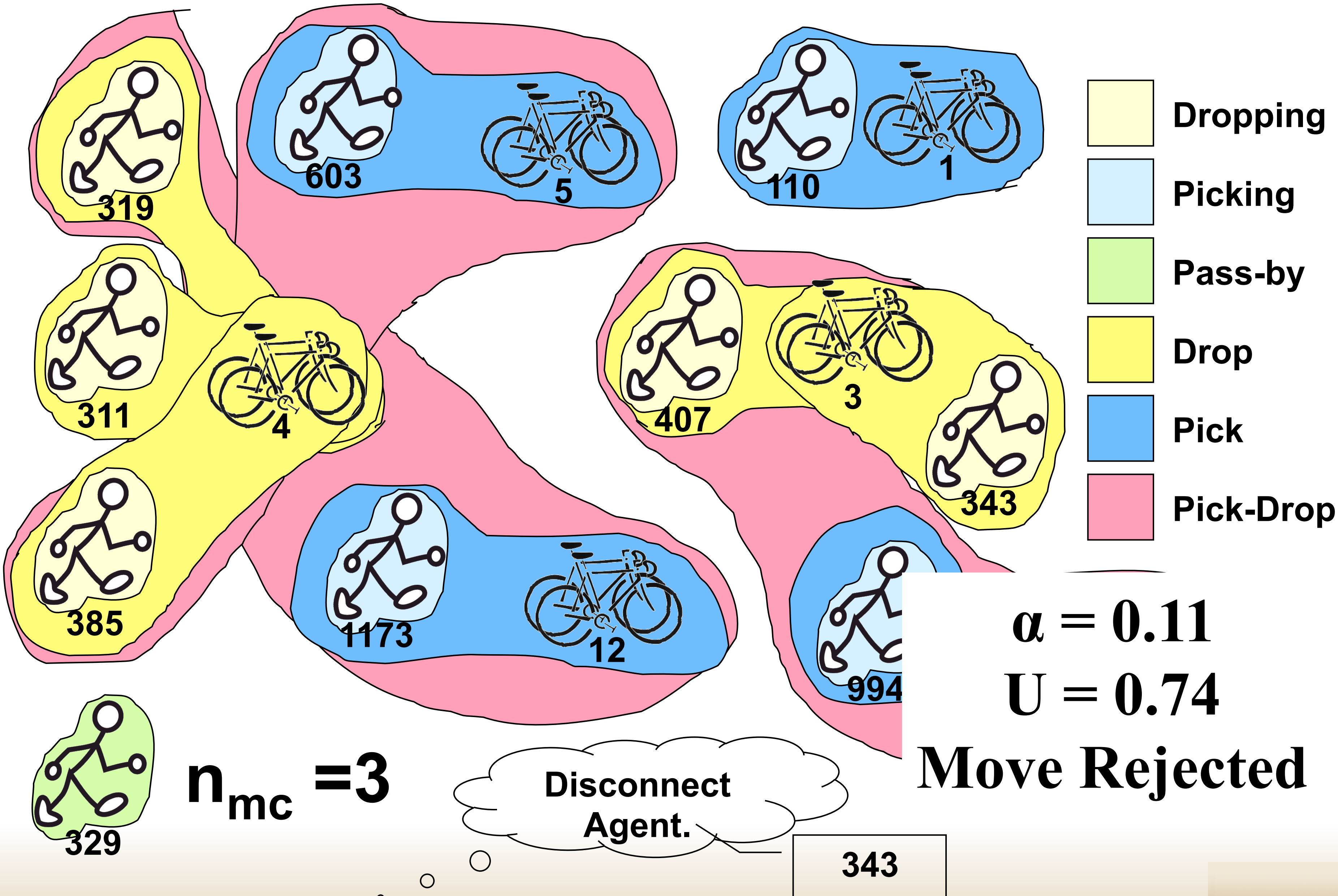
$\alpha = 1.0$

$U = 0.24$

Move Accepted

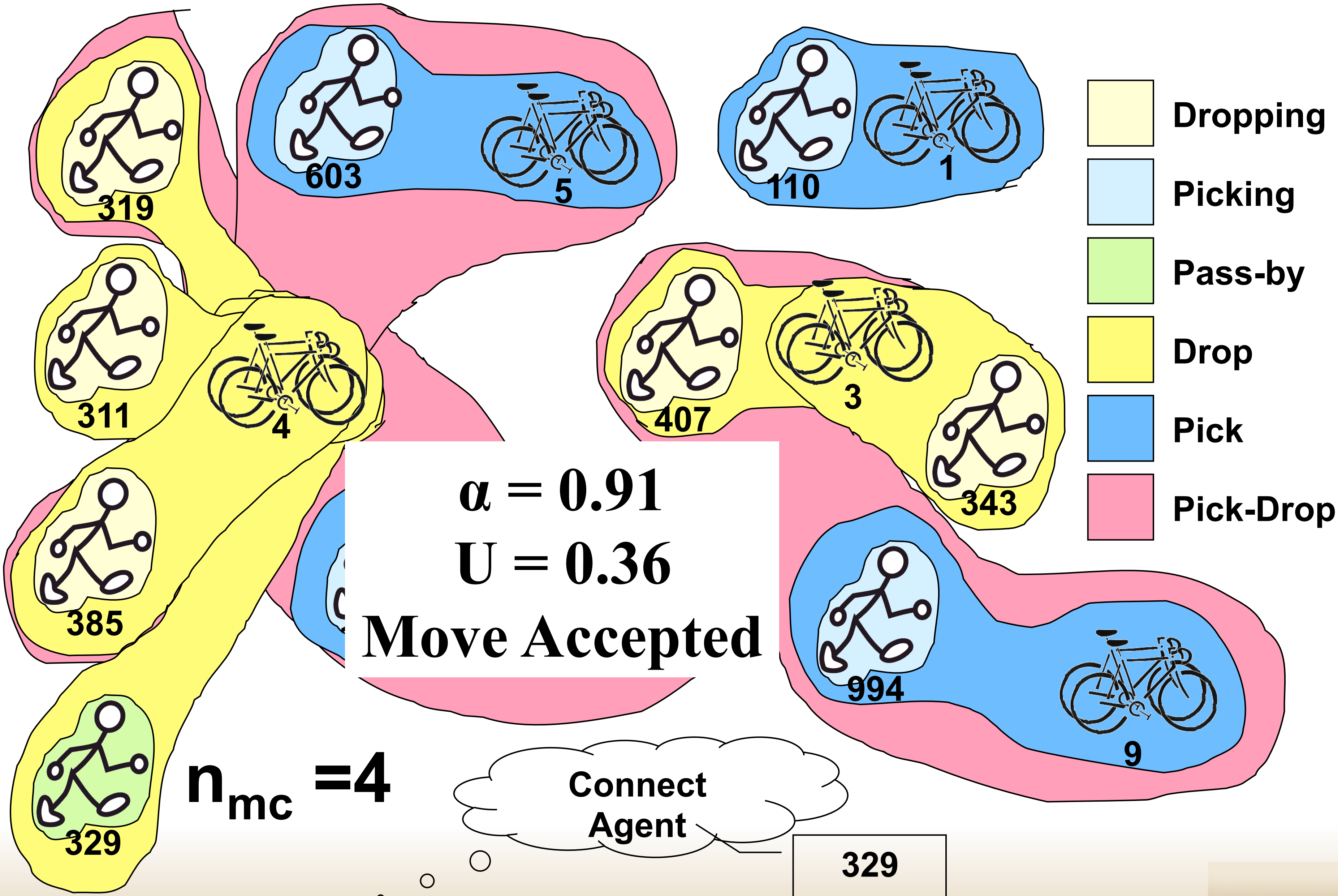
$n_{mc} = 2$

Examples



319
603
5
110
1
311
4
407
3
343
385
1173
12
994
329
Dropping
Picking
Pass-by
Drop
Pick
Pick-Drop
$n_{mc} = 3$
Disconnect Agent.
343
$\alpha = 0.11$
$U = 0.74$
Move Rejected

Examples



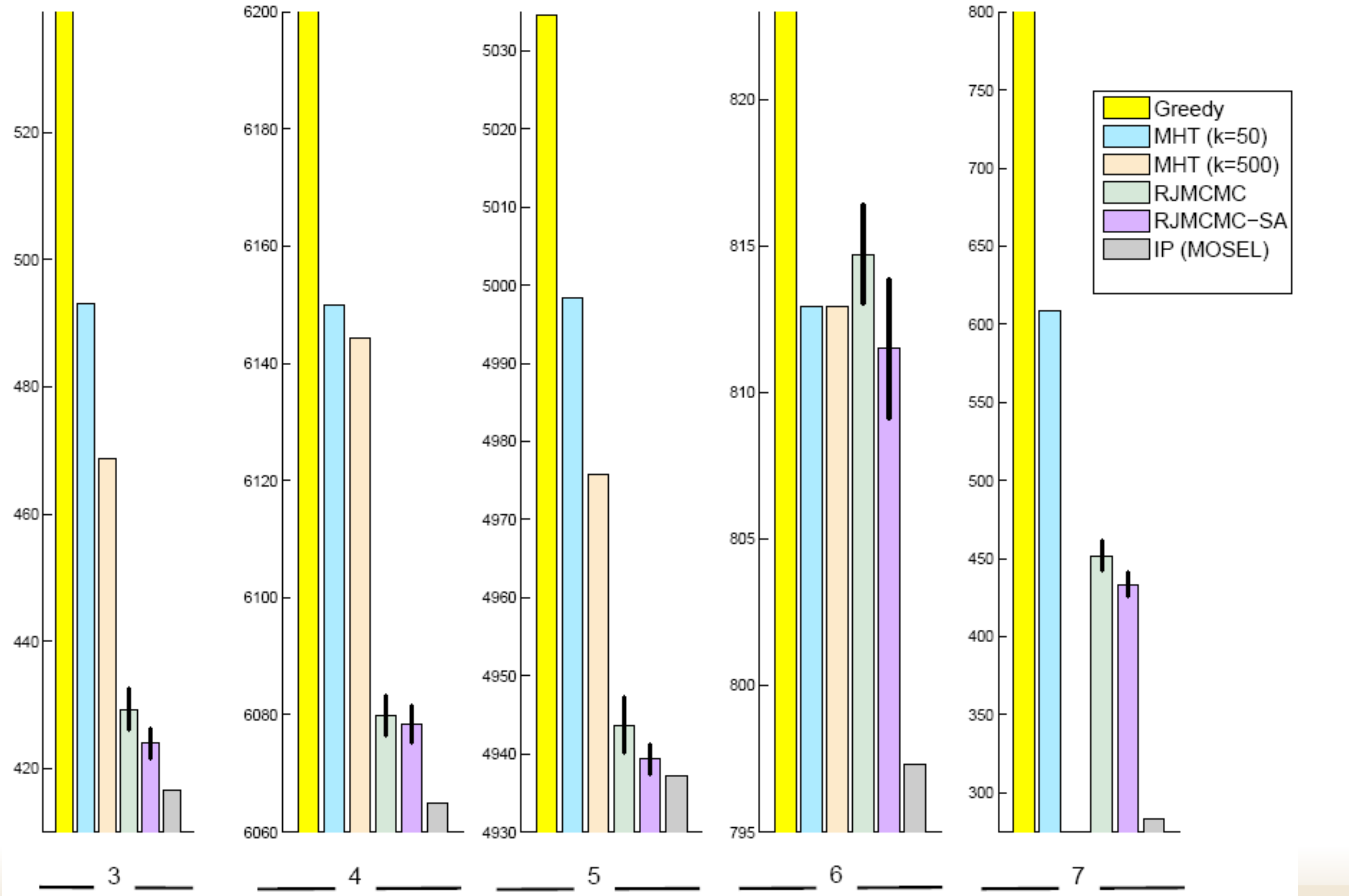
319
603
5
110
1
311
4
407
3
343
385
994
9
329
Dropping
Picking
Pass-by
Drop
Pick
Pick-Drop
$\alpha = 0.91$
$U = 0.36$
Move Accepted
$n_{mc} = 4$
Connect Agent
329

Dataset

Site 1
3 days (37 hours)
476 people
453 bicycle clusters
82 drop-picks

Site 2
2 days (30 hours)
309 people
2053 bicycle clusters
36 drop-picks

Results



Results

-log(p)

	Greedy	MHT			RJMCMC		RJMCMC-SA		IP	
		k=50	k=100	k=500	μ	σ	μ	σ	MATLAB	XPRESS-MP
1	102.25	58.78	58.78	57.86	57.90	0.11	57.86	0.00	57.86	57.86
2	23.54	4.64	4.64	4.64	4.64	0.00	4.64	0.00	4.64	4.64
3	609.66	493.18	468.80	468.80	429.30	3.23	423.98	2.36	416.64	416.64
4	6272.69	6149.95	6144.98	6144.30	6079.88	3.43	6078.40	3.23	6065.0	6065.00
5	5034.46	4998.39	4982.86	4975.82	4943.71	3.59	4939.33	1.87	4937.1	4937.08
6	860.37	812.96	812.96	812.96	814.71	1.69	811.50	2.36	797.29	797.29
7	934.36	608.92	607.39	-	451.92	9.29	433.50	7.76	-	283.51

accuracy

	Local	Global									
		Greedy	MHT			RJMCMC		RJMCMC-SA		IP	
			k=50	k=100	k=500	μ	σ	μ	σ	MATLAB	XPRESS-MP
1	74.13	72.41	91.38	91.38	91.38	88.36	1.09	87.46	1.79	91.38	91.38
2	85.19	85.19	100.00	100.00	100.00	100.00	0.00	100.00	0.00	100.00	100.00
3	64.06	58.59	84.38	84.38	84.38	87.68	0.89	83.36	1.65	88.28*	87.5*
4	74.60	73.81	74.60	75.40	75.40	83.93	1.09	83.15	1.31	81.75*	83.33*
5	86.13	89.05	82.48	84.67	88.32	91.90	0.79	92.65*	0.90	94.16	94.16
6	65.18	66.07	60.71	60.71	60.71	68.53	1.68	70.98	1.04	73.21	73.21
7	46.18	45.69	44.67	45.69	-	47.28	1.18	47.61	0.88	-	46.70

Results

We actually caught thieves!!

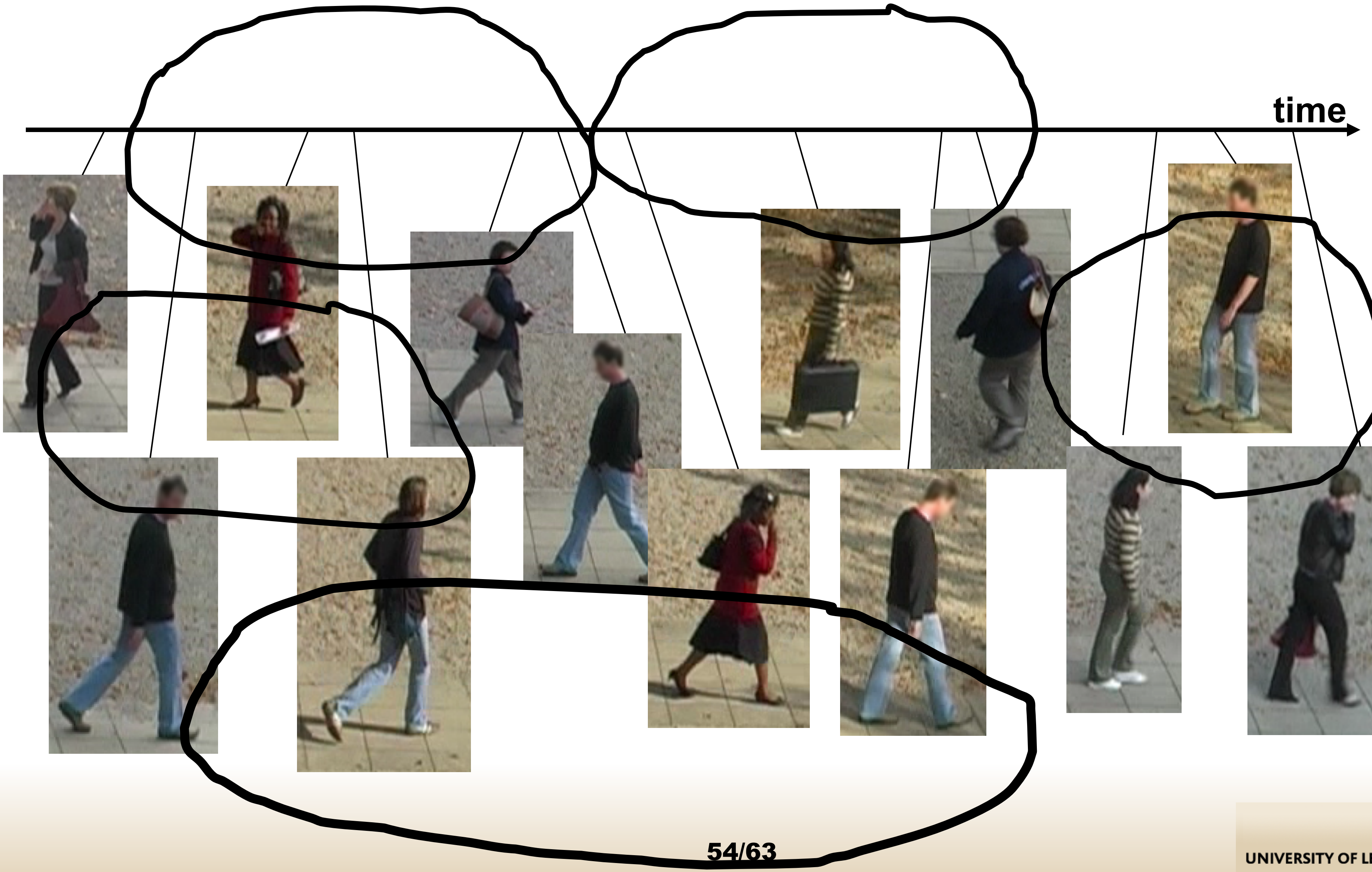
Recorded time: 11 hours and 30 minutes
Warning time: 13 minutes

Case II: The Enter-Exit Problem

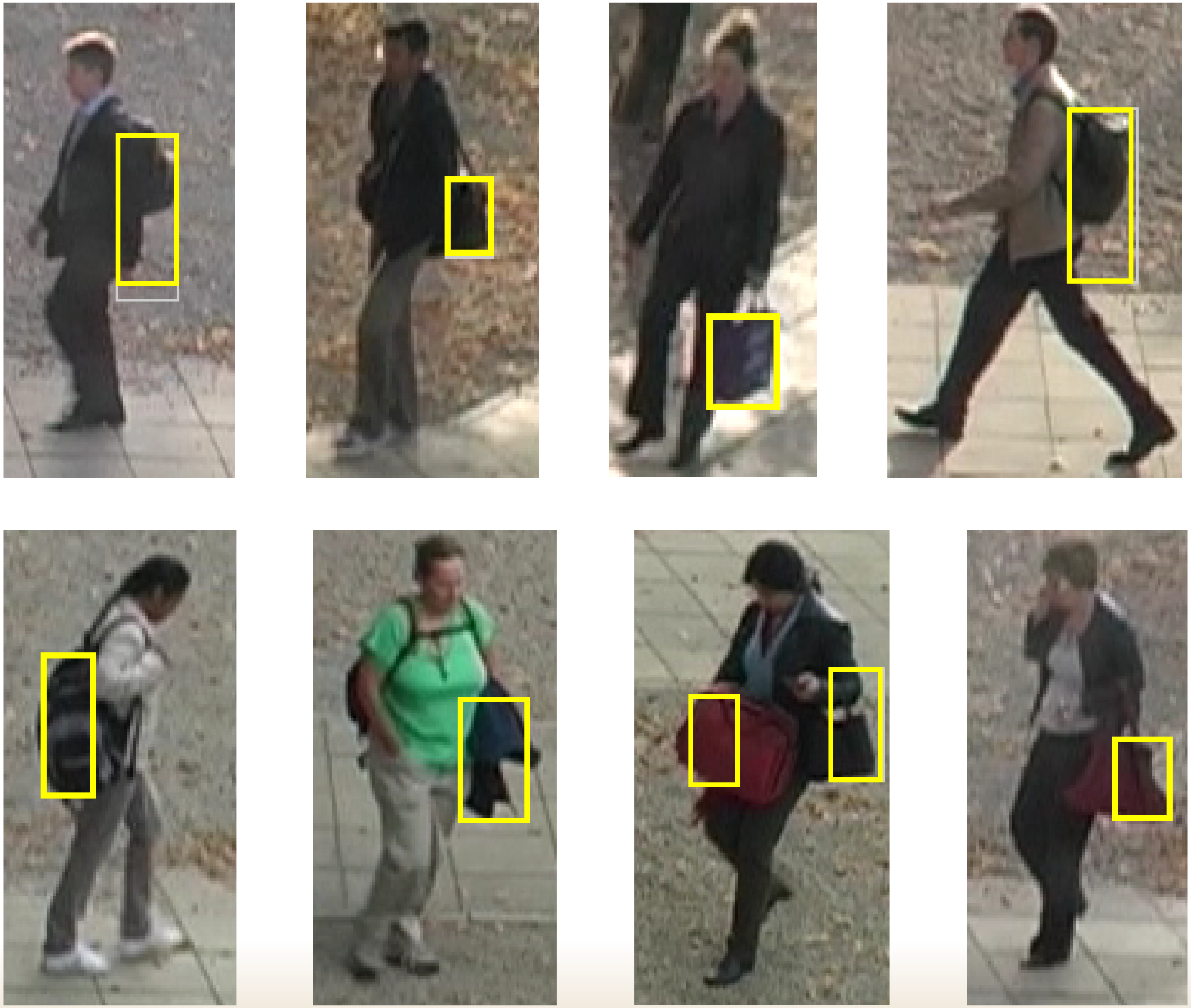
Global Explanation

- Global explanation

Global Explanation

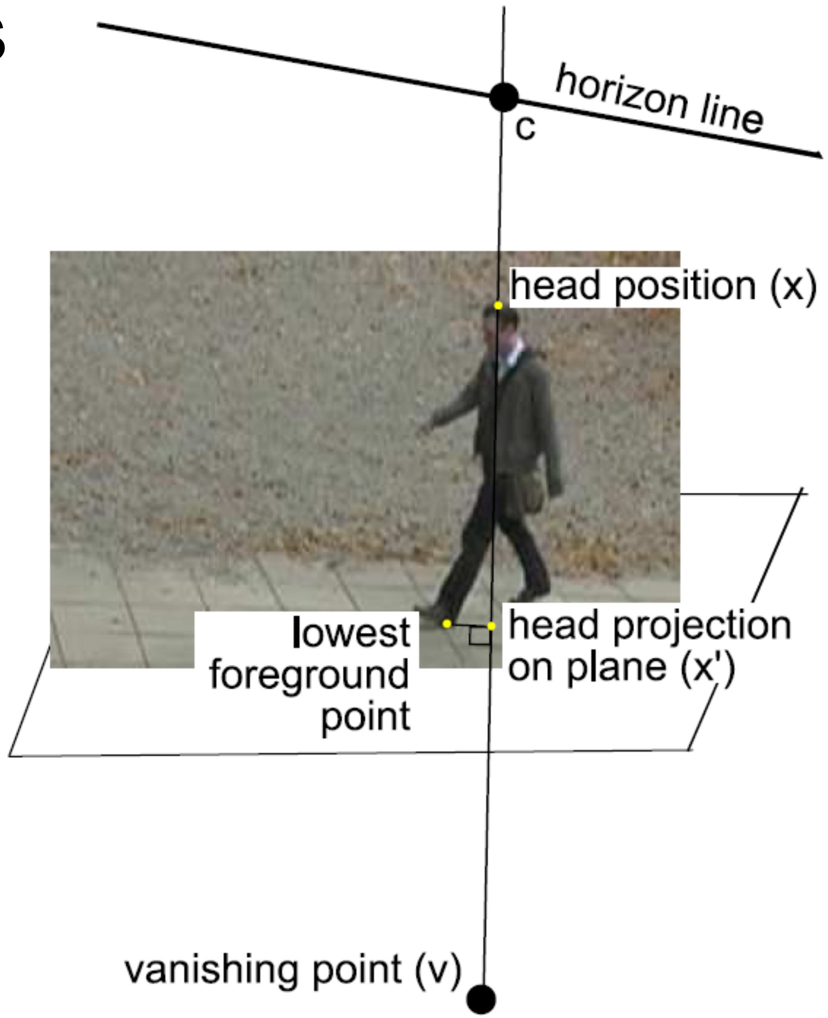


Using Baggage Detector [ECCV 08]



Selected Features

1. Matching Heights



Selected Features

2. Clothing Colour

Selected Features

3. Baggage Colour

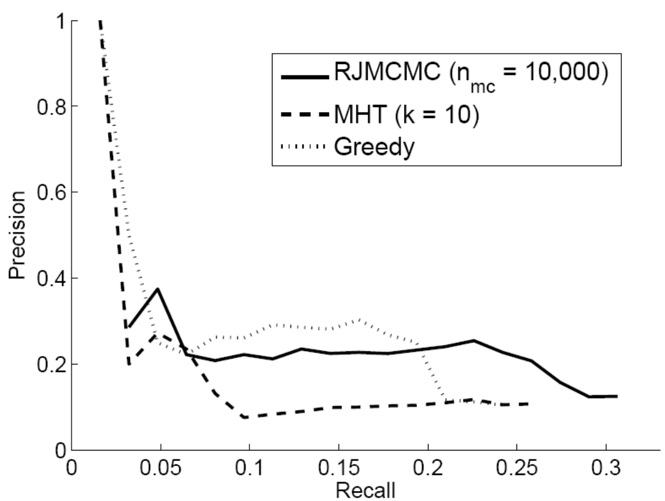
Selected Features

4. Baggage Relative Height

Experiment

- 12 hours
- 326 people
- 429 candidate bags
- 62 ground truth pairs

Results



	Local	Global		
		Greedy	MHT	RJMCMC
Paired	13	14	16	19
Unpaired	49	48	46	43
Incorrect Pairs	173	133	135	142

Conclusion

- Defining activity using AMG
  - Hierarchies of events
  - Multisets
  - Intra-activity constraints → synthetic attributes
  - Inter-activity constraints → inherited attributes
- Finding the best parse tree → Recognition
  - Building BN
  - Searching for MAP
- Two case studies

Thank you ☺

Damen, Dima and Hogg, David (Sep 2009). Attribute Multiset Grammars for Global Explanations of Activities. British Machine Vision Conference (BMVC).

Damen, Dima and Hogg, David (June 2009). Recognizing Linked Events: Searching the Space of Feasible Explanations. Computer Vision and Pattern Recognition (CVPR).

Damen, Dima and Hogg, David (Oct 2008). Detecting Carried Objects from Short Video Sequences. European Computer Vision Conference (ECCV).