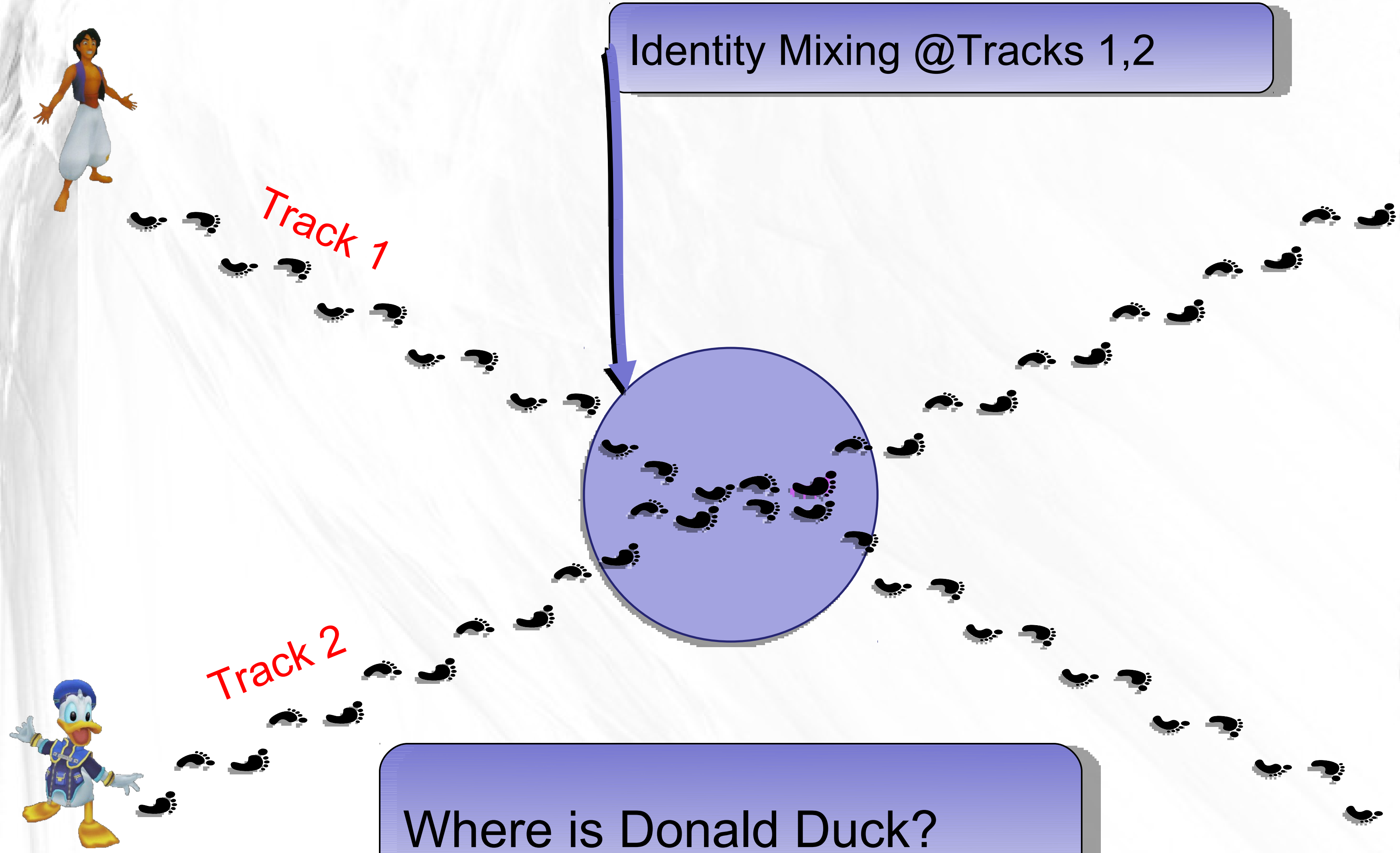
Identity Management Problem

— Reasoning and Inference over Permutations

Xiaoye Jiang
Stanford University

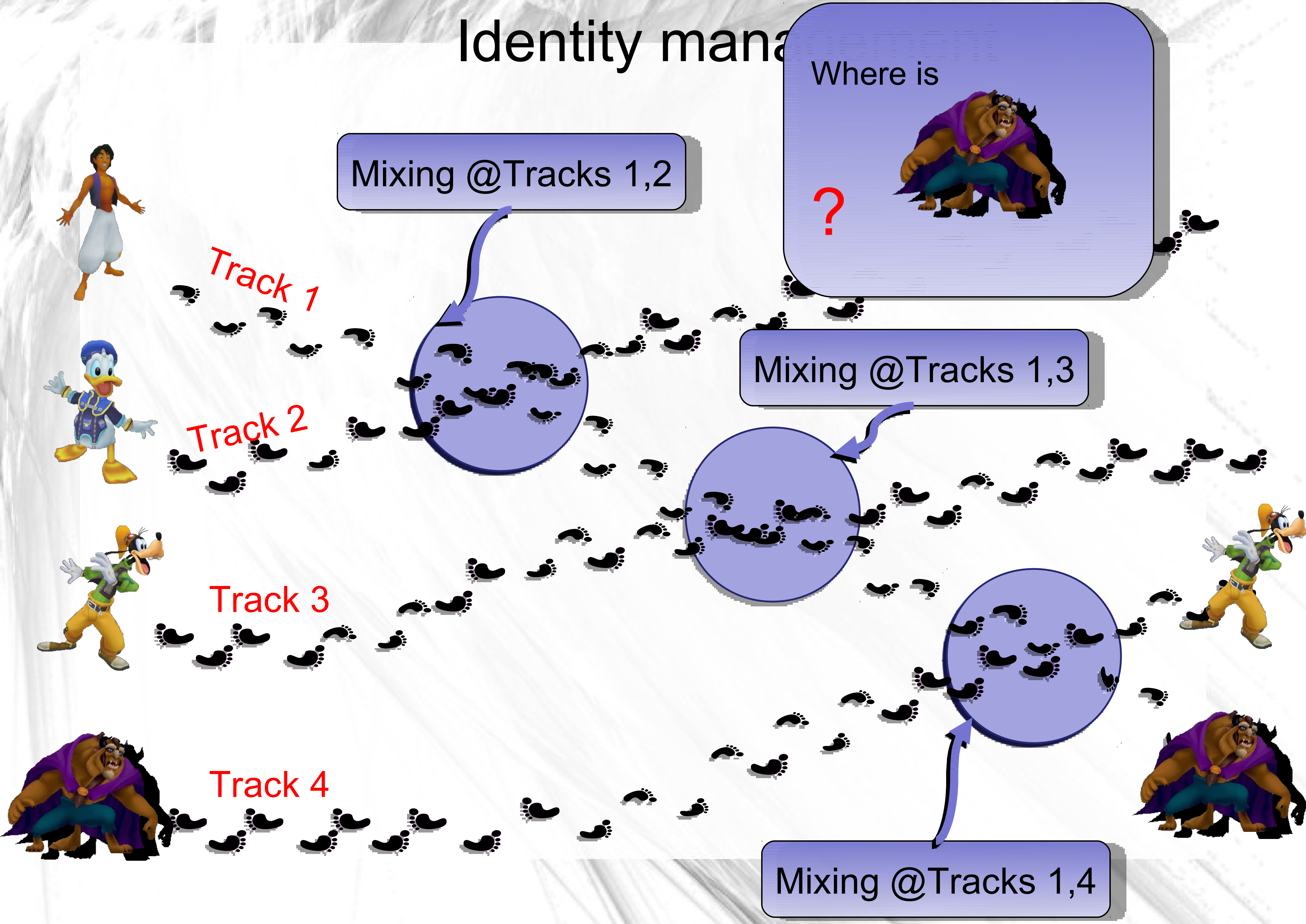
Joint work with J. Huang, C. Guestrin, L. Guibas

Identity management [Shin et al., '03]



Where is Donald Duck?

Identity management



Where is

?

Mixing @Tracks 1,2

Mixing @Tracks 1,3

Mixing @Tracks 1,4

Track 1

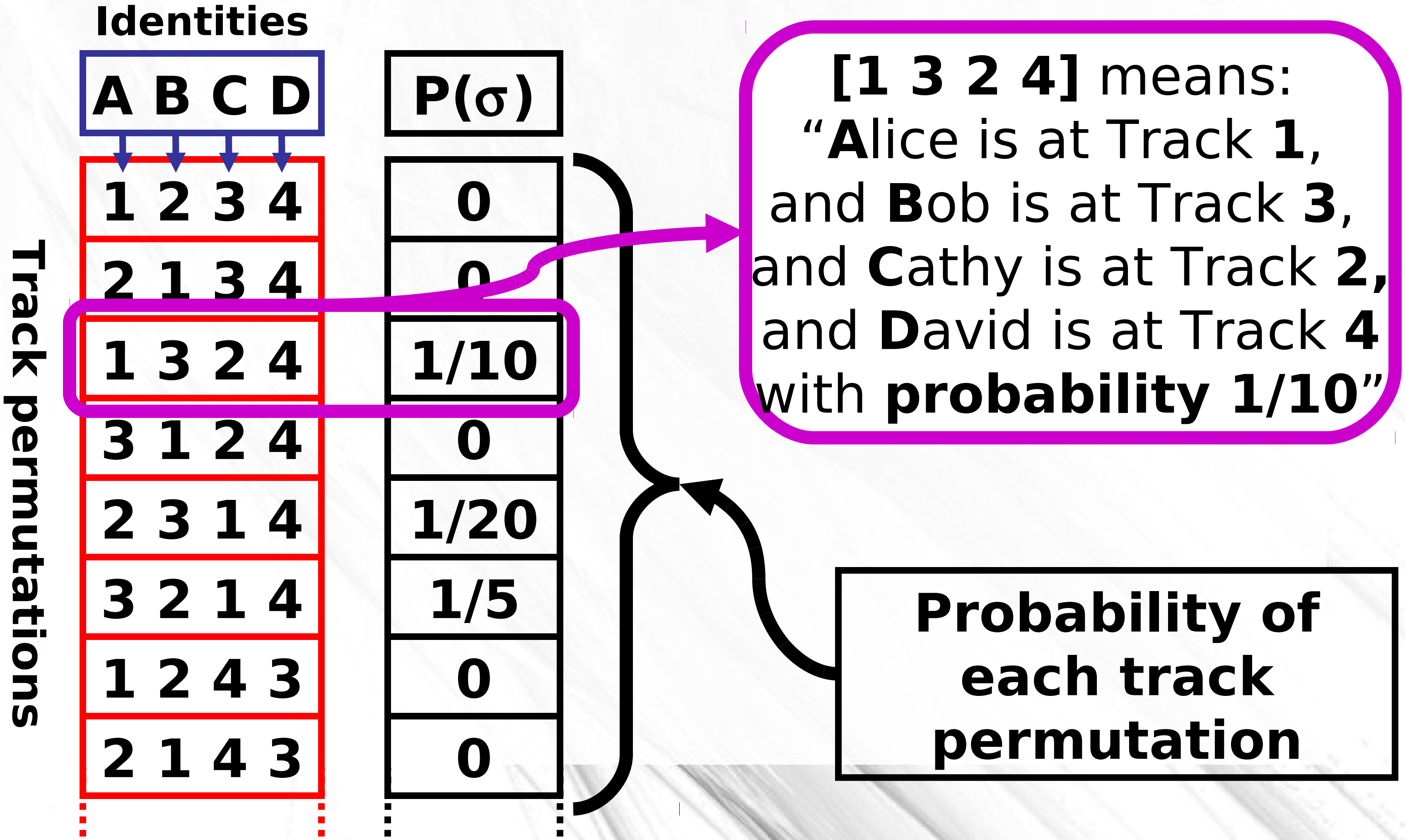
Track 2

Track 3

Track 4

Reasoning with Permutations

- We **model uncertainty** in identity management with **distributions over permutations**



Identities

A B C D	P(σ)
1 2 3 4	0
2 1 3 4	0
1 3 2 4	1/10
3 1 2 4	0
2 3 1 4	1/20
3 2 1 4	1/5
1 2 4 3	0
2 1 4 3	0

[1 3 2 4] means: "**A**lice is at Track **1**, and **B**ob is at Track **3**, and **C**athy is at Track **2**, and **D**avid is at Track **4** with **probability 1/10**"

Probability of each track permutation

How many permutations?

- There are **n!** permutations!

n	n!	Memory required to store n! doubles
9	362,880	3 megabytes
12	4.8x10^{8}	9.5 terabytes
15	1.31x10^{12}	1729 petabytes

My advisor won’t buy me this much memory!

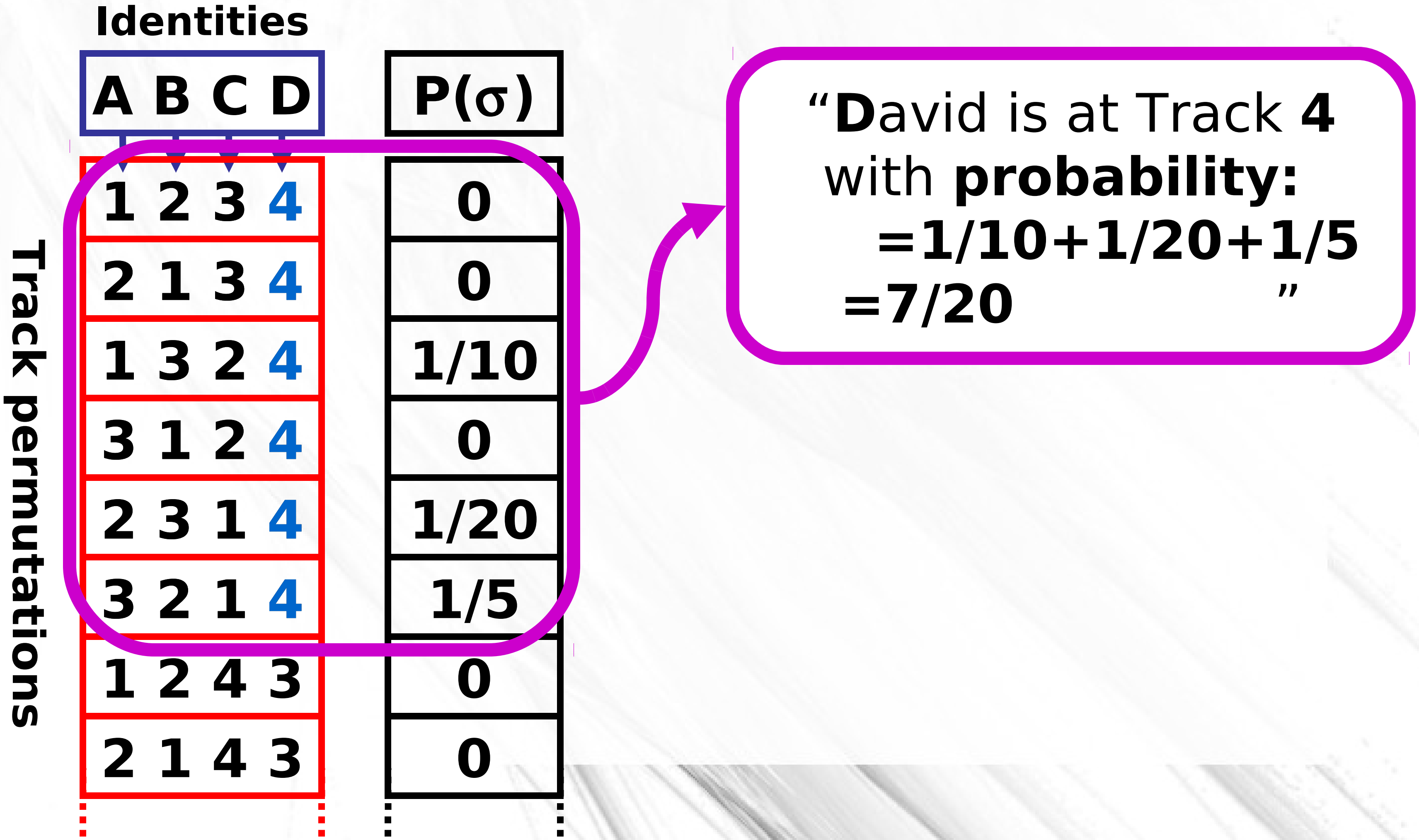
- Graphical models are not effective due to mutual exclusivity constraints (“**A**lice and **B**ob cannot both be at Track **1** simultaneously”)

Objectives

- We would like to:
  - Find a **principled**, **compact representation** for **distributions over permutations** with **tuneable approximation quality**
  - Reformulate **Markov Model inference** operations with respect to our new representation:
    - Marginalization
    - Conditioning

1st order summaries

- An idea: For each (identity **j**, track **i**) pair, store **marginal probability** that **j** maps to **i**



Identities

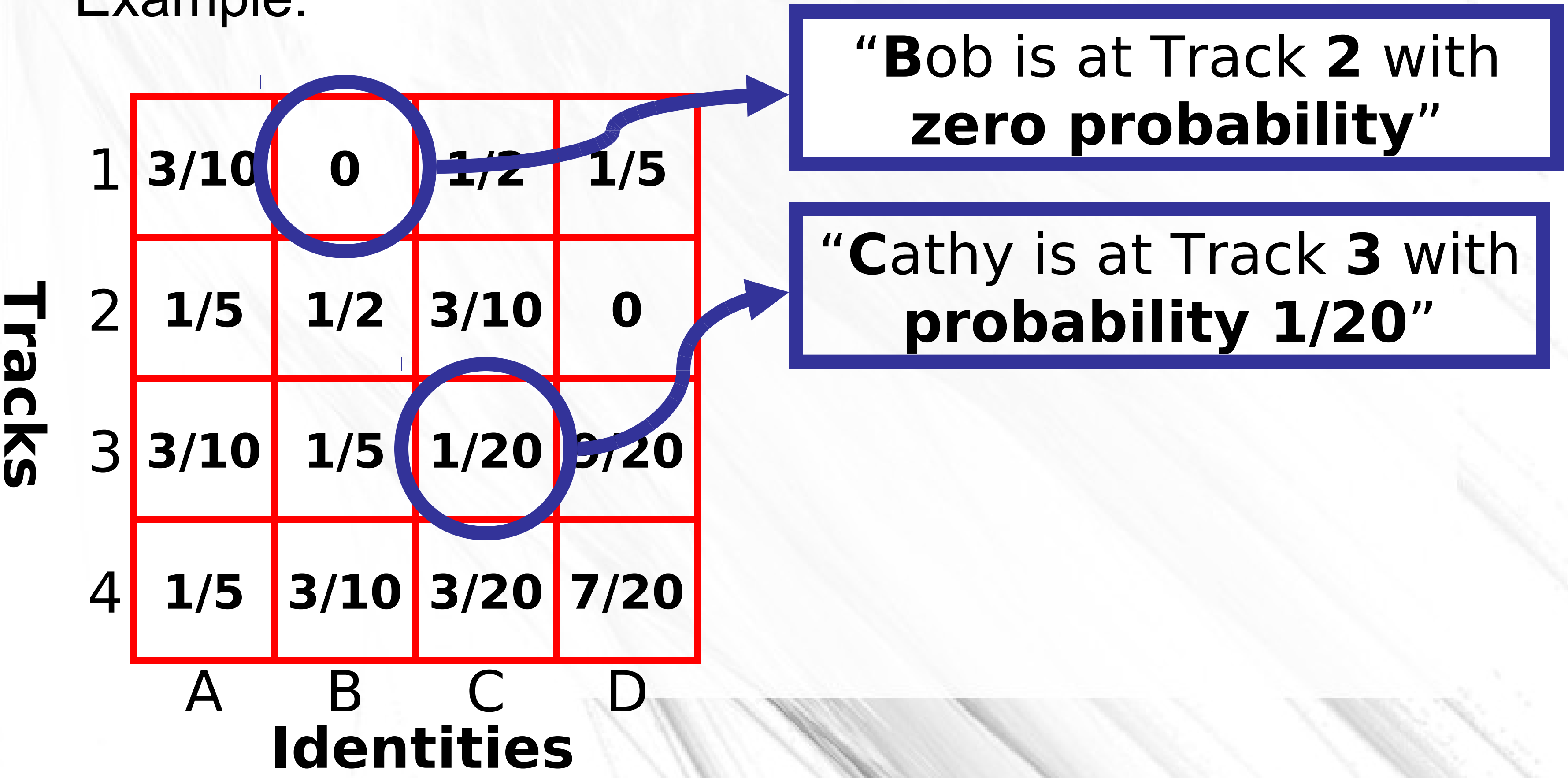
Track permutations

A	B	C	D	P(σ)
1	2	3	4	0
2	1	3	4	0
1	3	2	4	1/10
3	1	2	4	0
2	3	1	4	1/20
3	2	1	4	1/5
1	2	4	3	0
2	1	4	3	0

“**D**avid is at Track **4** with **probability:**
=1/10+1/20+1/5
=7/20 ”

1st order summaries

- We can **summarize a distribution** using a **matrix of 1st order marginals**
- Requires storing only **n^2** numbers!
- Example:



Tracks \ Identities	A	B	C	D
1	3/10	0	1/2	1/5
2	1/5	1/2	3/10	0
3	3/10	1/5	1/20	9/20
4	1/5	3/10	3/20	7/20

"**B**ob is at Track **2** with **zero probability**"

"**C**athy is at Track **3** with **probability 1/20**"

The problem with 1st order

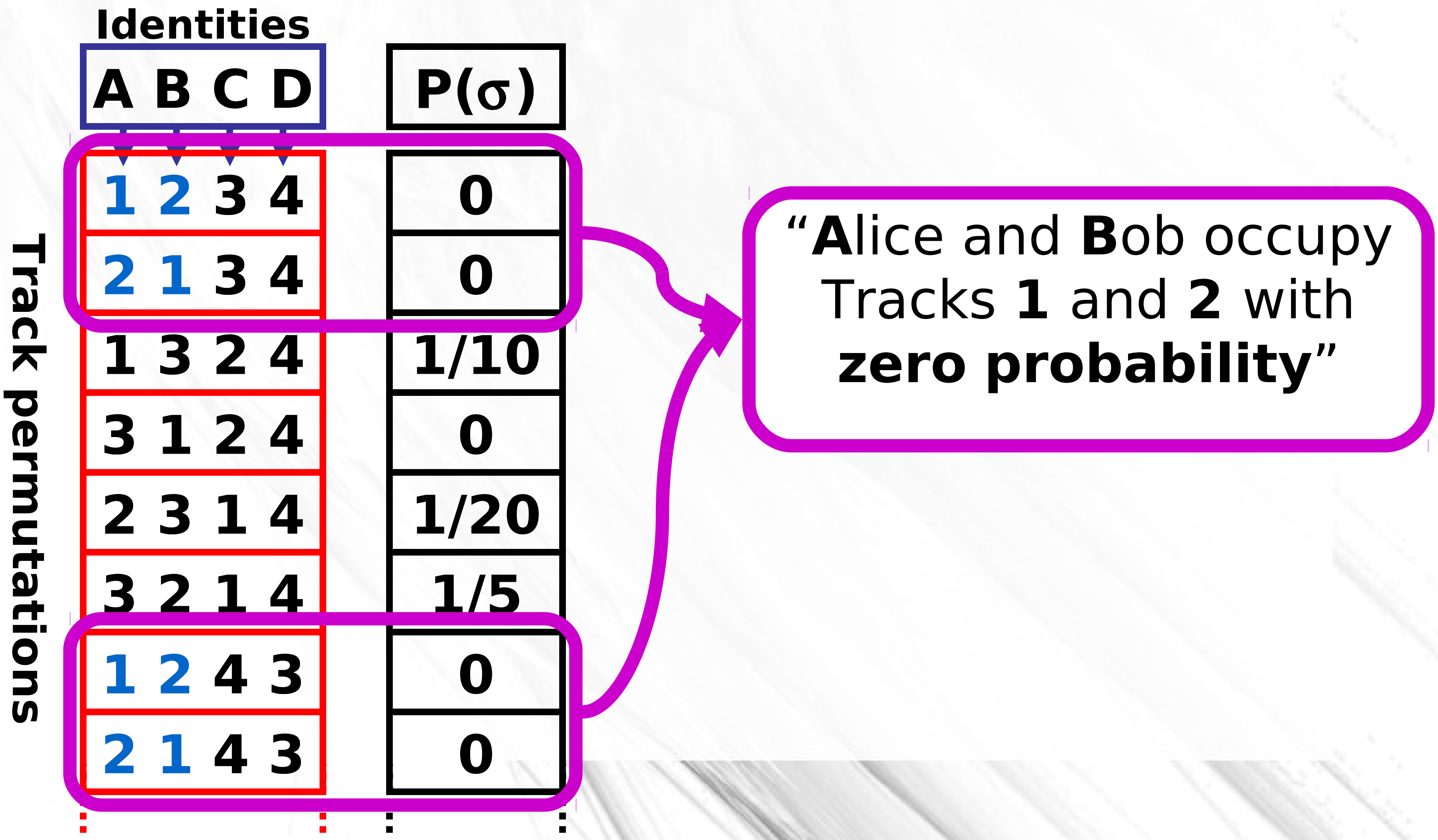
- What 1st order summaries **can** capture:
  - P(**A**lice is at Track **1**) = **3/5**
  - P(**B**ob is at Track **2**) = **1/2**
- N

1st order summaries **cannot capture higher order dependencies!**

  - P(**{A**lice,**B**ob**}** occupy Tracks **{1,2}**) = **0**

2nd order summaries

- Idea #2: store **marginal probabilities** that unordered pairs of identities **{k,l}** map to pairs of tracks **{i,j}**



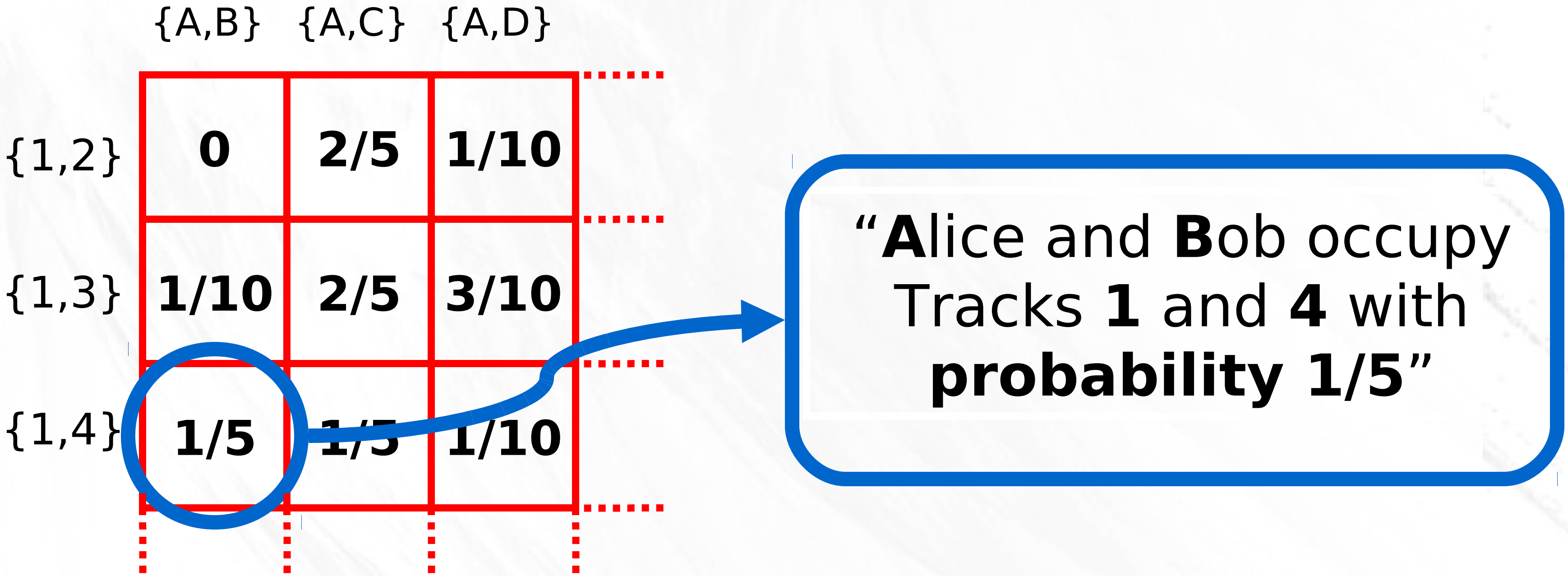
Identities

Track permutations

A	B	C	D	P(σ)
1	2	3	4	0
2	1	3	4	0
1	3	2	4	1/10
3	1	2	4	0
2	3	1	4	1/20
3	2	1	4	1/5
1	2	4	3	0
2	1	4	3	0

"**A**lice and **B**ob occupy Tracks **1** and **2** with **zero probability**"

2nd order summaries



	{A,B}	{A,C}	{A,D}
{1,2}	**0**	**2/5**	**1/10**
{1,3}	**1/10**	**2/5**	**3/10**
{1,4}	**1/5**	**1/5**	**1/10**

“**A**lice and **B**ob occupy Tracks **1** and **4** with **probability 1/5**”

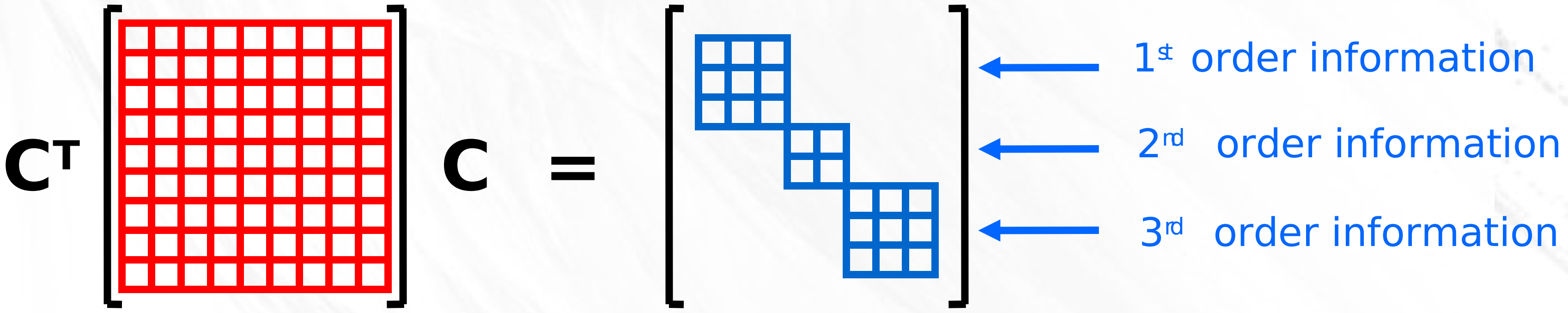
Et cetera…

- **And so forth…** We can define:
  - **3rd-order** marginals
  - **4th-order** marginals
  - **…**
  - **nth-order** marginals
    - (which recovers the original distribution but requires n! numbers)

- **Fundamental Trade-off**: we can capture higher-order dependencies at the cost of storing more numbers

Discarding redundancies

- Matrices of marginal probabilities carry redundant information
  - Example on 4 identities: the probability that {**A**lice,**B**ob} occupy Tracks {**1**,**2**} must be the same as the probability that {**C**athy,**D**avid} occupy Tracks {**3**,**4**}
- Can efficiently find a matrix C to “remove redundancies“:



1st order information

2rd order information

3rd order information

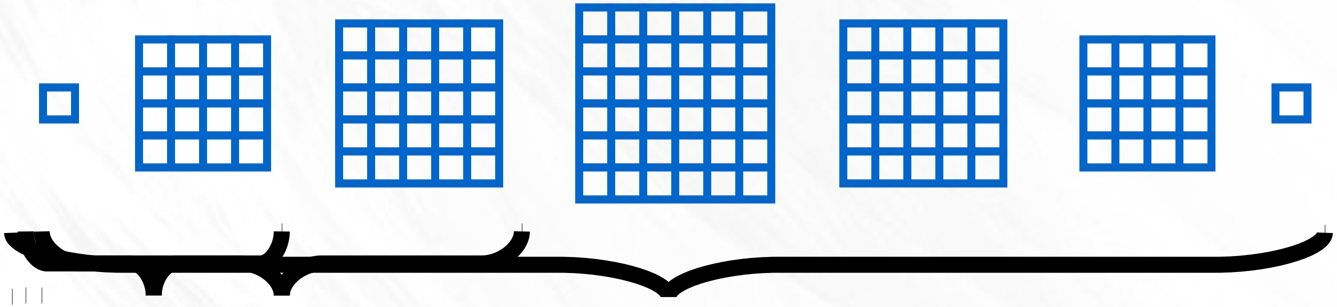
Matrix of high-order marginals

Block-diagonal sum of coefficients

- Instead of storing marginals, only store these blocks of coefficients (from which marginals can be reconstructed)

Completeness

- If we have enough coefficients (by removing the redundancies from nth order marginals), we can reconstruct the original distribution:



1st order

2nd order

nth order

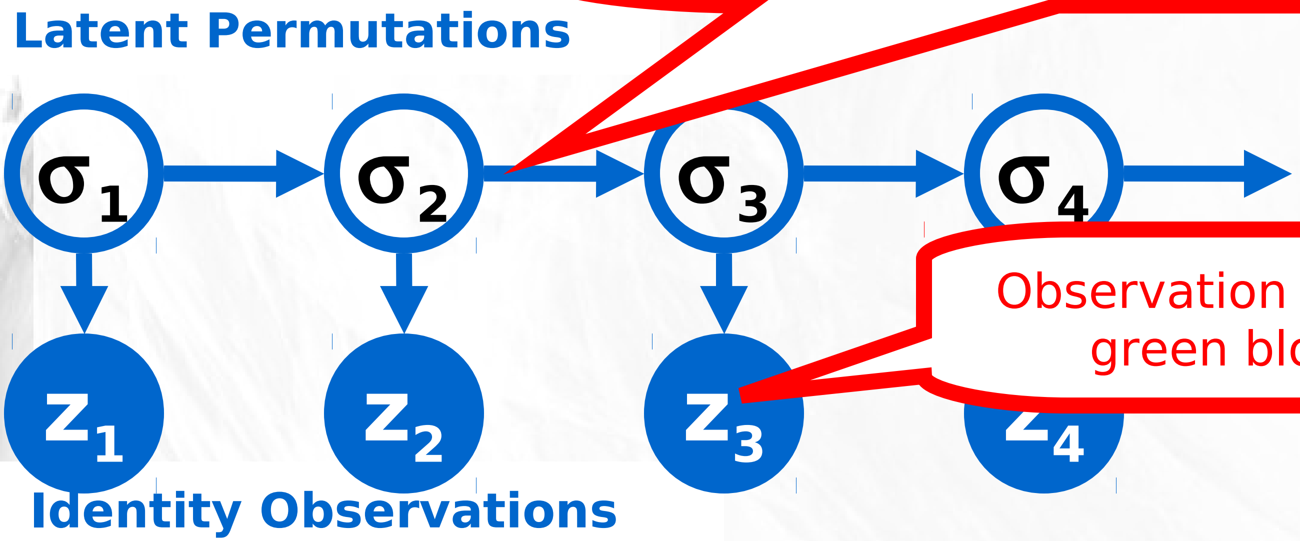
(Complete Basis for functions over permutations)

The Fourier interpretation

- The compact representations can be viewed as a **generalized Fourier basis** [Diaconis, ‘88]:
  - The familiar properties hold: **Linearity**, **Orthogonality**, **Completeness**, **Plancherel’s (Parseval’s) theorem**, **Convolution theorem**, ...

- **To do inference** using low dimensional **Fourier projections**, we need to cast all inference operations **in the Fourier domain** 🙂

Hidden



Mixing Model – “e.g. Tracks **2** and **3** swapped identities with probability ½”

Observation Model – “e.g. see green blob at track **3**”

- Problem statement: For each timestep, find posterior marginals conditioned on all past observations
- Need to formulate inference routines with respect to Fourier coefficients!

Hidden Markov model inference

- **Two basic inference operations** for Hidden Markov Models:
  - **Prediction/rollup:**

$$P_{t+1}(\sigma_{t+1}) = \sum_{\sigma_t} P(\sigma_{t+1}|\sigma_t)P_t(\sigma_t)$$

  - **Conditioning:**

$$P(\sigma|z) \propto P(z|\sigma)P(\sigma)$$

- How can we do these operations without enumerating all n! permutations?

Prediction/Rollup

- We assume that σ_{t+1} is generated by the rule:
  - Draw $\tau \sim$ **Q(τ)**
  - Set $\sigma_{t+1} = \tau \cdot \sigma_t$

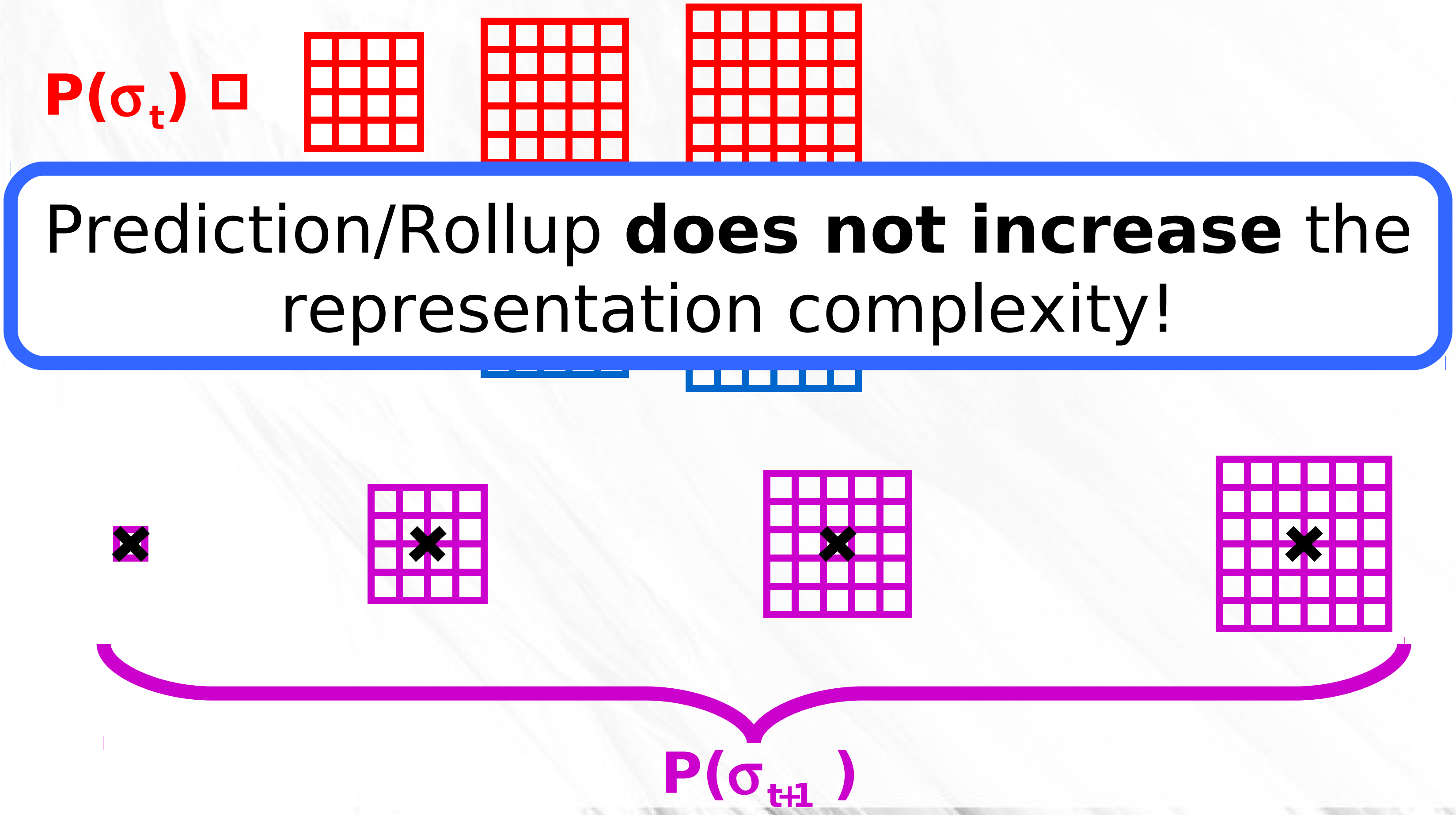
Mixing Model

- For example, **Q([2 1 3 4])=½** means that Tracks **1** and **2** swapped identities with probability ½.
- Prediction/Rollup can be written as a **convolution**:

$$P_{t+1}(\sigma_{t+1}) = \underbrace{\sum_{\sigma_t} P(\sigma_{t+1}|\sigma_t)P_t(\sigma_t)}_{\textbf{Convolution } (Q*P_t)!}$$

Fourier Domain Prediction/Rollup

- Convolutions are **pointwise products** in the Fourier domain:



Prediction/Rollup **does not increase** the representation complexity!

Conditioning

- **Bayes rule** is a **pointwise product** of the **likelihood function** and **prior distribution**:

$$\underbrace{P(\sigma|z)}_{\textbf{Posterior}} \propto \underbrace{P(z|\sigma)}_{\textbf{Likelihood}}\underbrace{P(\sigma)}_{\textbf{Prior}}$$

- Example likelihood function:
  - P(**z**=**green** | σ(**A**lice)=Track **1**) = 9/10
  - ("Prob. we see **green** at Track **1** given **A**lice is at Track **1** is **9/10**")

Kronecker Conditioning

Pointwise products correspond to **convolution in the Fourier domain** [Willsky, ‘78] (except with *Kronecker Products* in our case)

Our algorithm handles **any prior** and **any likelihood**, generalizing the previous FFT-based conditioning method [Kondor et al., ‘07]

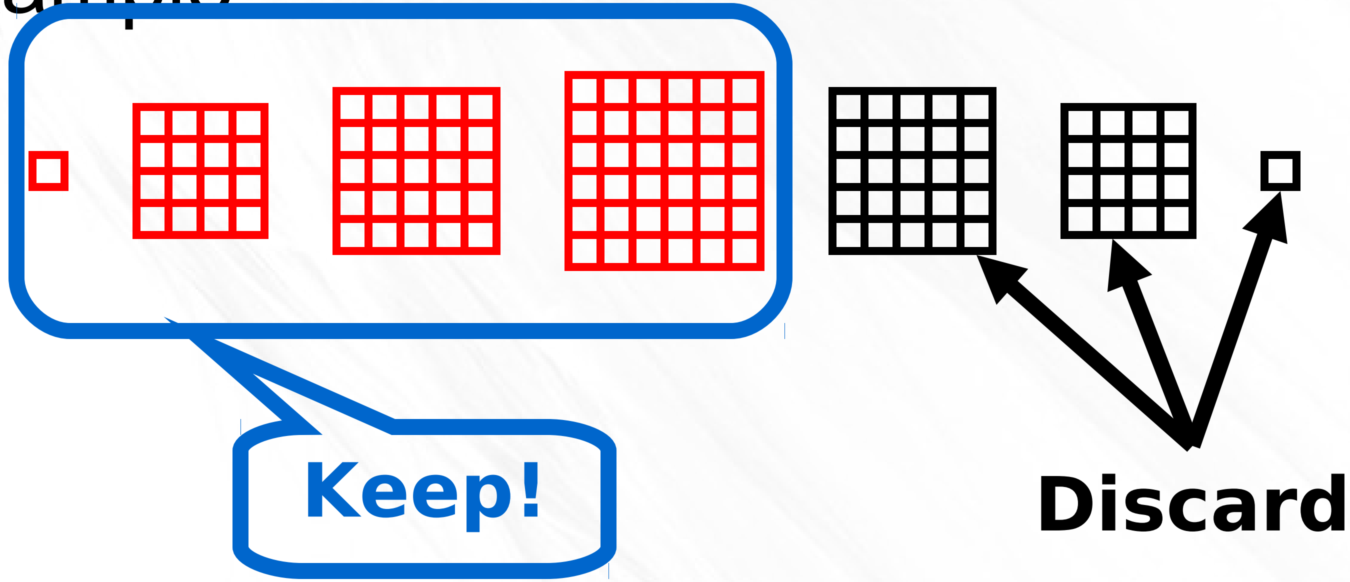
Conditioning

- Conditioning **increases the representation complexity!**
- Example: Suppose we start with **1st order marginals of**
  - P(**A**lice is at
  - P(**B**ob is at Track **1** or Track **2**) = .5
  - …
- Then we make a **1st order observation**:
  - "**C**athy is at Track **1** or Track **2** with probability 1"
- (This means that **A**lice and **B**ob cannot both be at Tracks **1** and **2**!)
  - P(**{Alice,Bob}** occupy Tracks **{1,2}**)=0

Need to store 2nd order probabilities after conditioning!

Bandlimiting

- After conditioning, we **discard “high-frequency” coefficients**
  - Equivalently, we **maintain low-order marginals**
- Example:



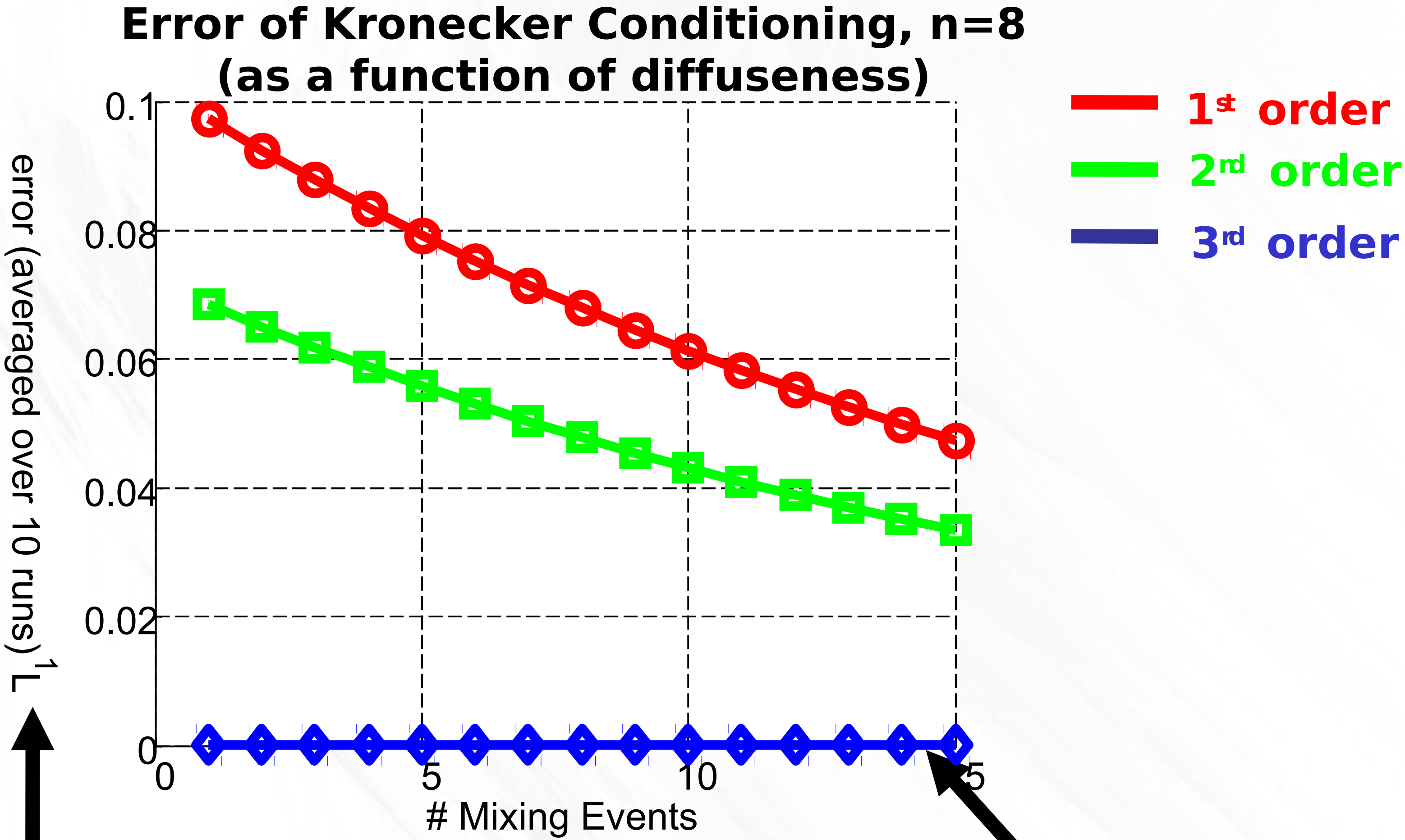
Error analysis

- Fourier domain Prediction/Rollup **is exact** ☺
- Kronecker Conditioning **introduces error** ☹
- But…
  - If **enough** coefficients are maintained, then Kronecker conditioning **is exact** at a subset of low-frequency terms! ☺

Theorem. If the Kronecker Conditioning Algorithm is called using **p^{th} order terms of the prior** and **q^{th} order terms of the likelihood**, then the **$(|p-q|)^{th}$ order marginals of the posterior** can be reconstructed **without error**.

Kronecker Conditioning experiments

Better



Measured at 1st order marginals

(Keeping 3rd order marginals is enough to ensure zero error for 1st order marginals)

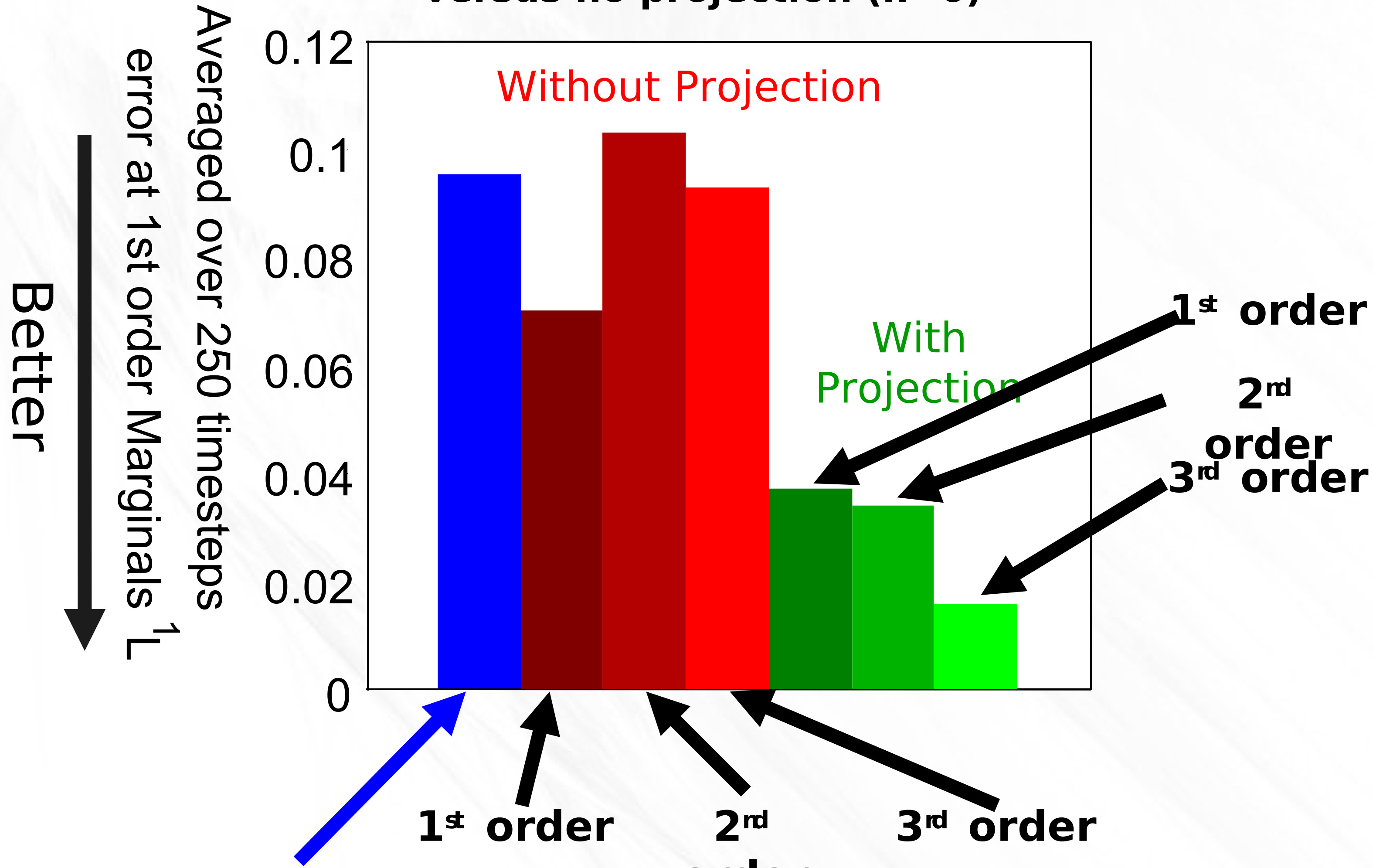
Dealing with negative numbers

- Consecutive Conditioning steps **can propagate errors to all frequency levels**
- Errors can sometimes cause our marginal probabilities to be **negative!** ☹

- **Our Solution: Project** to relaxed **Marginal Polytope** (space of Fourier coefficients corresponding to **nonnegative marginal probabilities**) ☺
  - Projection can be formulated as an **efficient Quadratic Program** in the **Fourier domain**

Simulated data drawn from HMM

Projection to the Marginal polytope versus no projection (n=6)



1st order

2nd order

3rd order

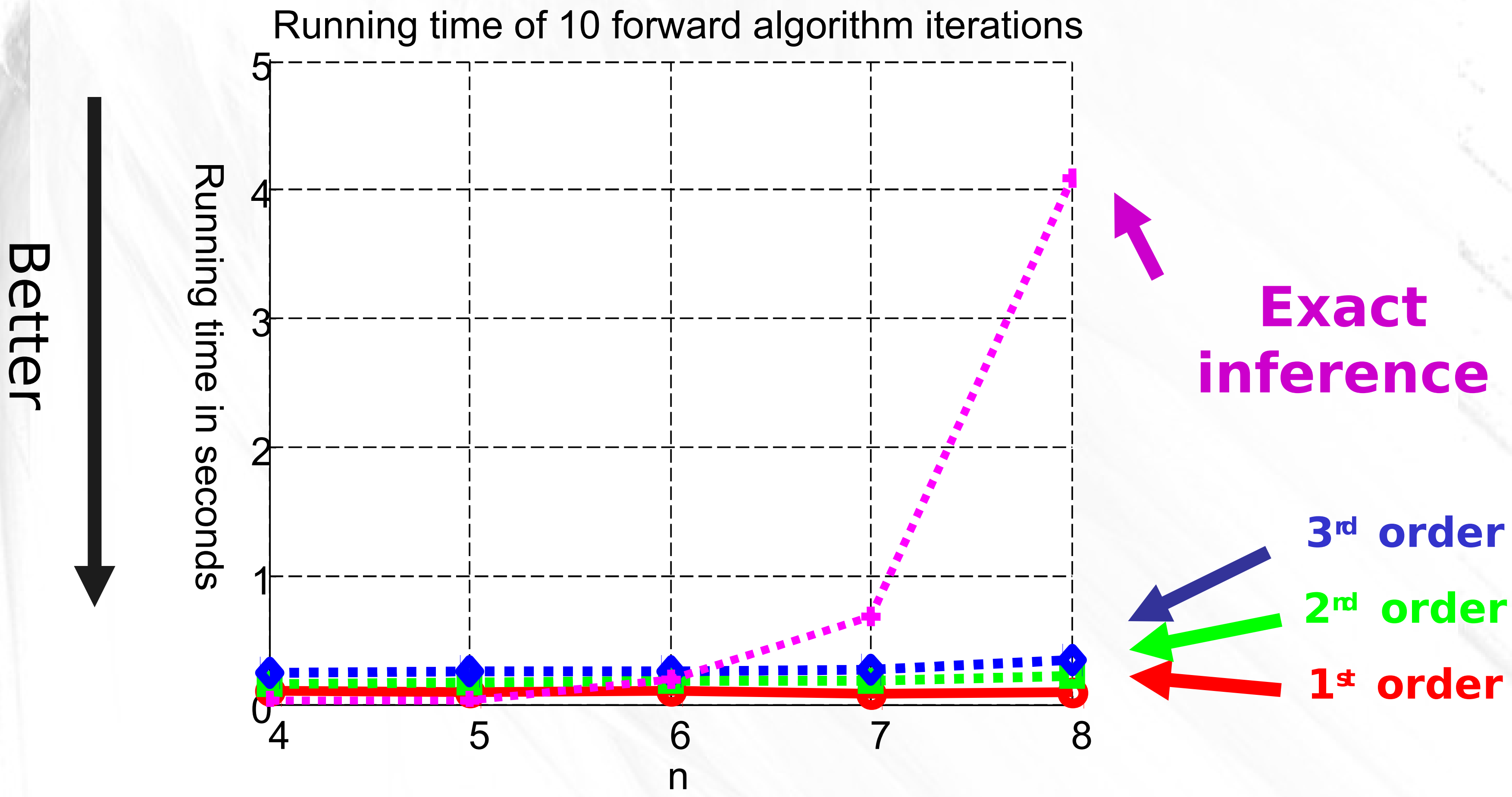
1st order

2nd order

3rd order

Approximation by a uniform distribution

Running Time comparison



Running time of 10 forward algorithm iterations

Better

Exact inference

3rd order

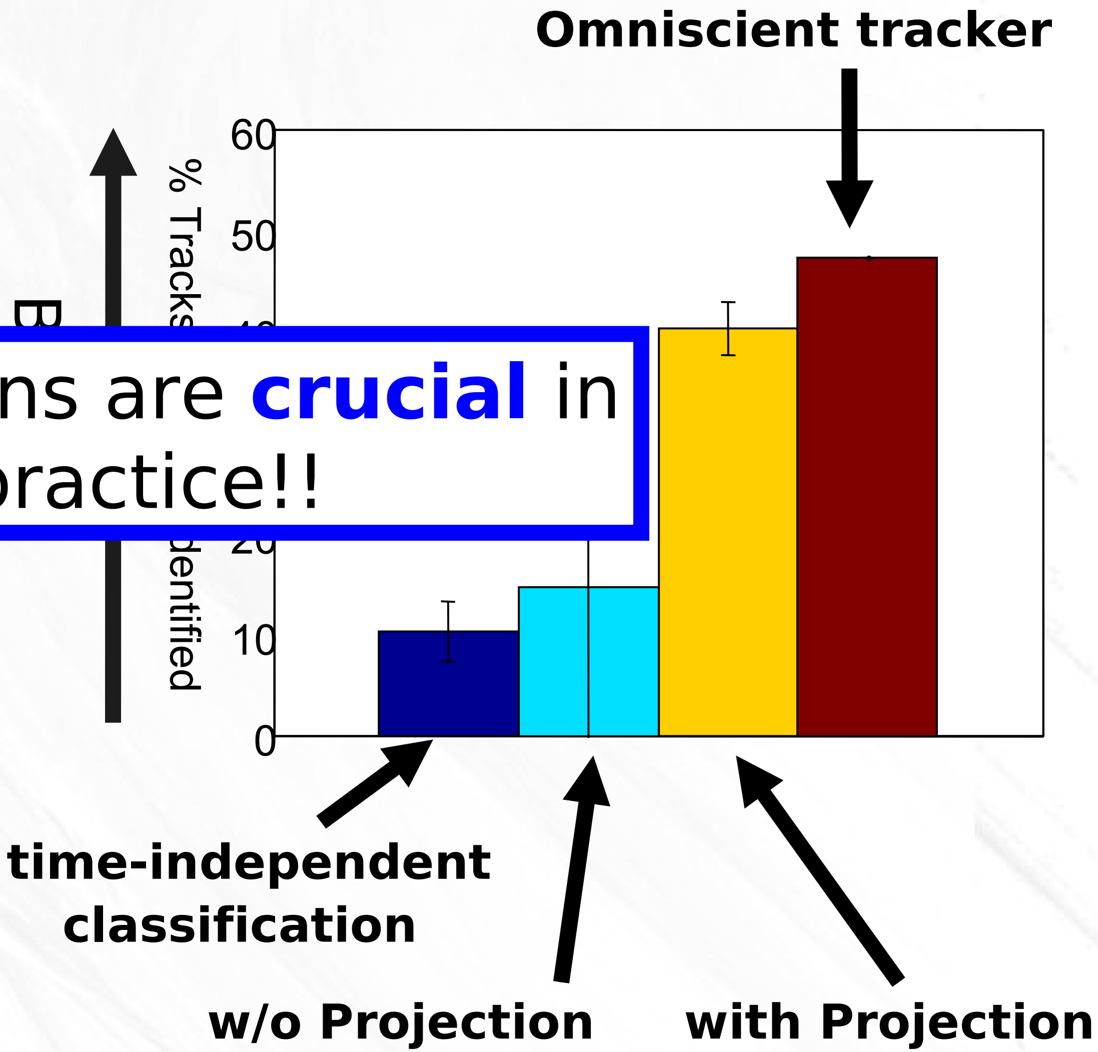
2nd order

1st order

Tracking with a camera network

- **Camera Network** data:
  - 8 cameras, multiview, occlusion effects
  - 11 individuals in lab
  - Identity observations obtained from color histograms
  - Mixing eve when peo to each ot

Projections are **crucial** in practice!!

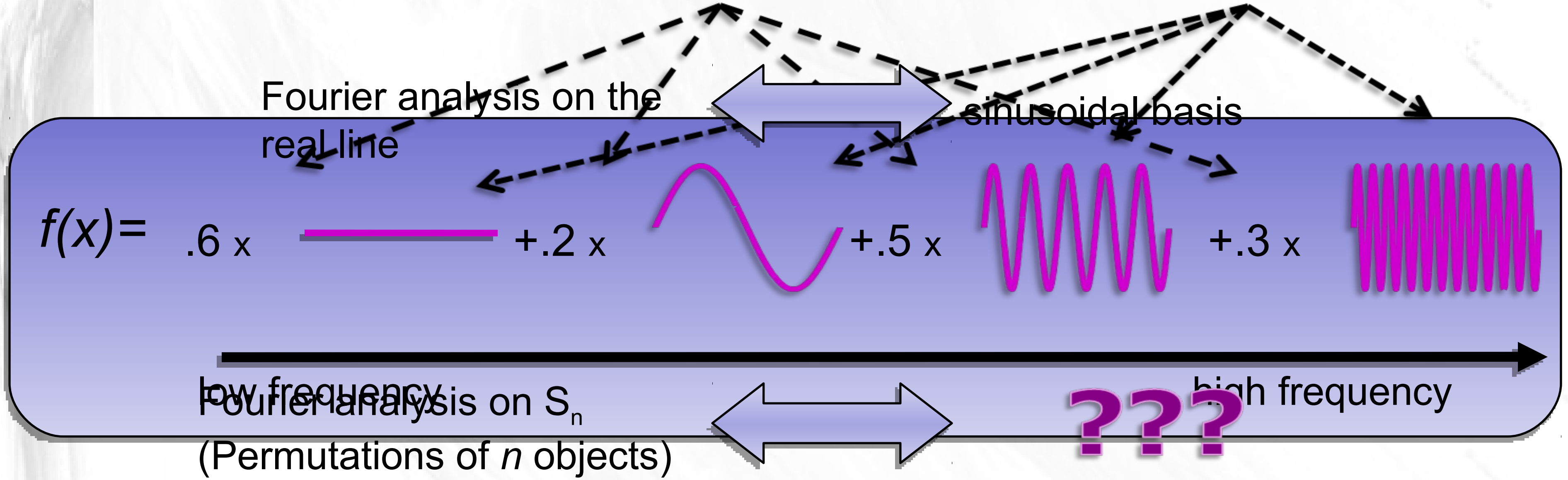


Summary of Fourier Approach

- Presented an intuitive, principled representation for distributions on permutations with
  - Fourier-analytic interpretations, and
  - Tuneable approximation quality
- Formulated general and efficient inference operations directly in the Fourier Domain
- Analyzed sources of error which can be introduced by bandlimiting and showed how to combat them by projecting to the marginal polytope
- Evaluated approach on real camera network application and simulated data

Fourier theoretic approaches

- Approximate distributions over permutations with **low frequency basis functions** [Kondor2007, Huang2007]



Fourier coefficients

Fourier basis functions

Fourier analysis on the real line

sinusoidal basis

f(x)= .6 x +.2 x +.5 x +.3 x

low frequency

high frequency

Fourier analysis on S_n
(Permutations of *n* objects)

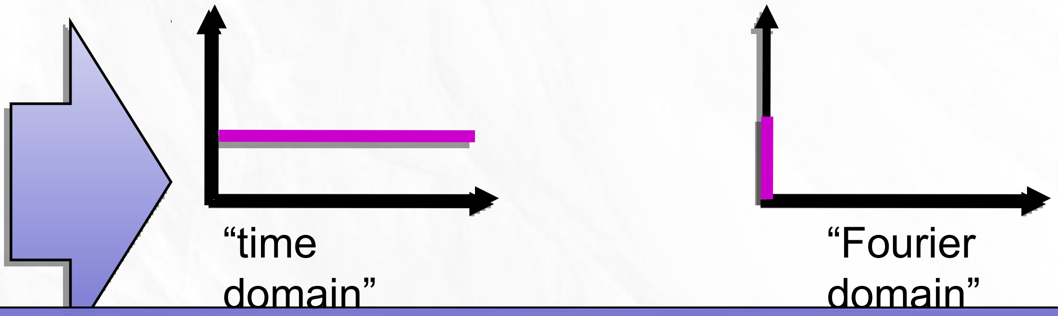
???

Uncertainty principle on a line

Signal *f*

Power spectrum

Uniform distribution

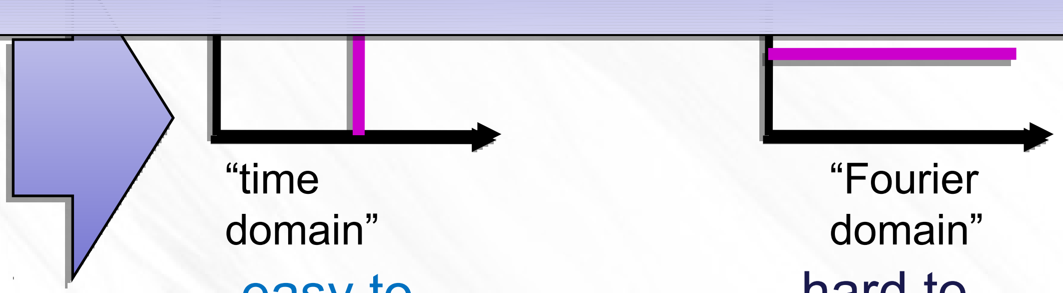


“time domain”

“Fourier domain”

Uncertainty Principle: a signal *f* cannot be sparsely represented in both the time *and* Fourier domains

Peaked distribution



“time domain”

easy to represent

“Fourier domain”

hard to represent

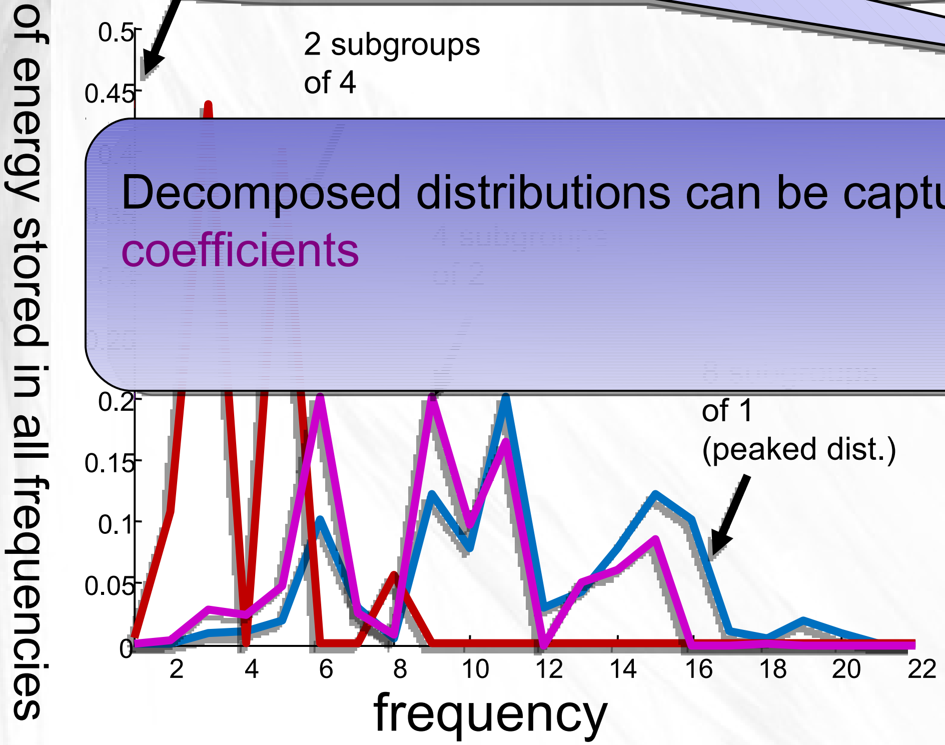
Uncertainty principle on permutations

P

confusion between tracks 1,2

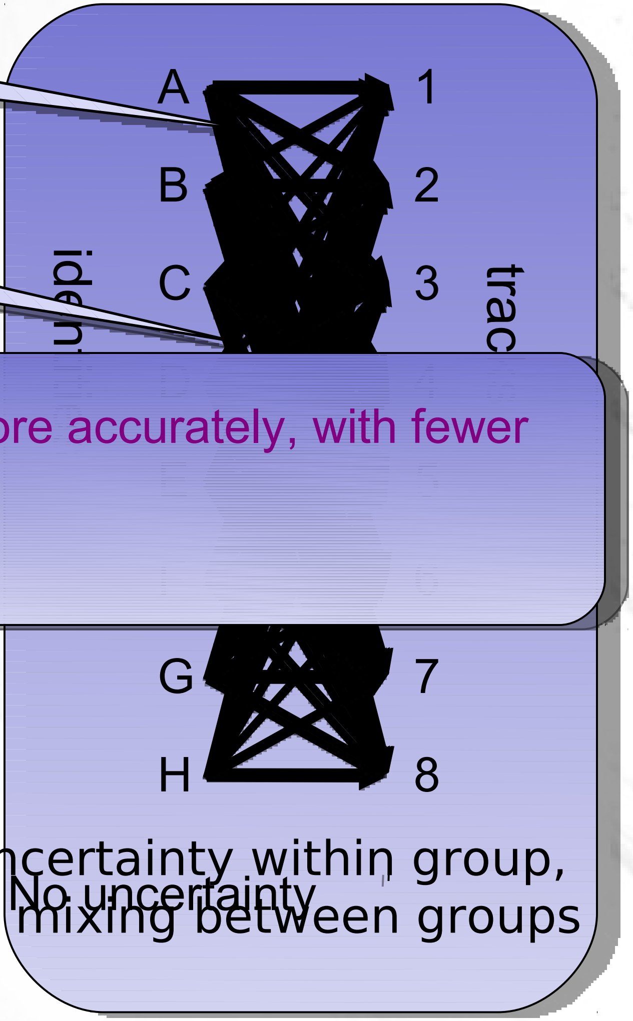
Uniform on 1

confusion between tracks 3,4



harder to represent

Decomposed distributions can be captured more accurately, with fewer coefficients



Uncertainty within group, no mixing between groups

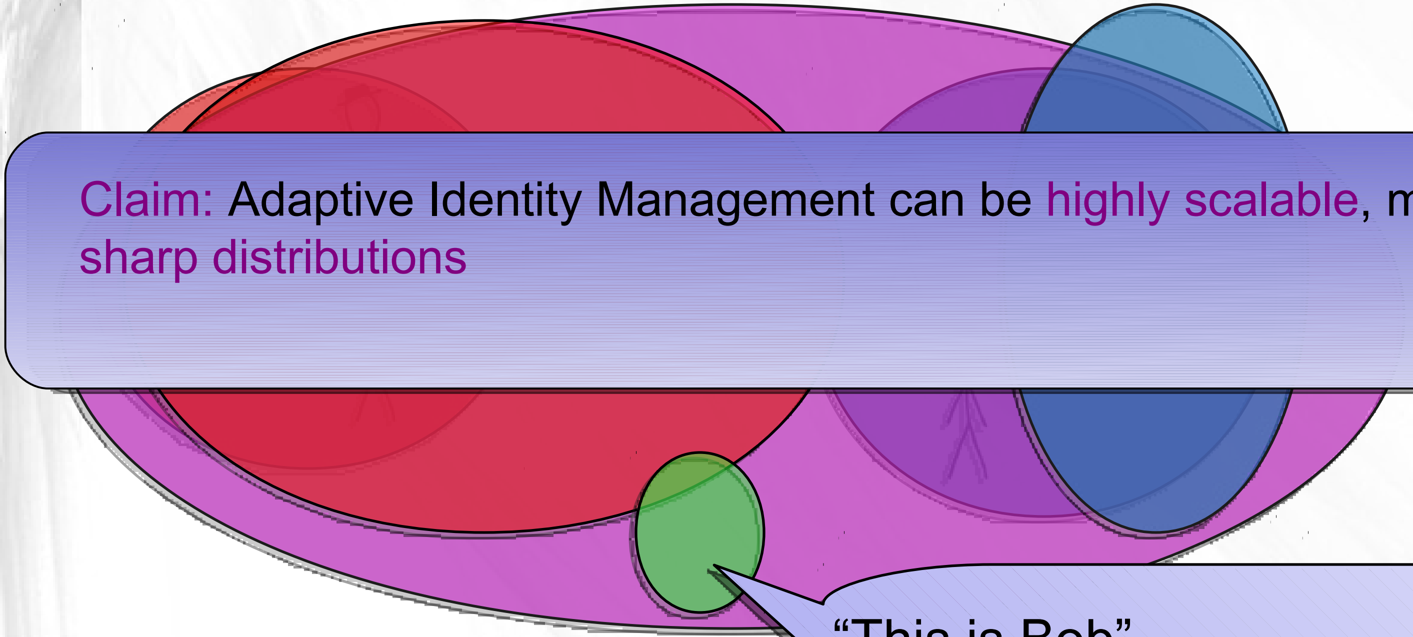
No uncertainty

Keep full joint distribution over S_8

8 independent distributions over S_1

Adaptive decompositions

- Our approach: **adaptively factor problem** into subgroups allowing for higher order representations for smaller subgroups



Claim: Adaptive Identity Management can be highly scalable, more accurate for sharp distributions

"This is Bob"

(and Bob was originally in the Blue group)

Contributions

- Characterization of constraints on Fourier coefficients on permutations implied by probabilistic independence

- Two algorithms: for factoring a distribution (Split) and combining independent factors in the Fourier domain (Join)

- Adaptive algorithm for scalable identity management (handles *up to n~100 tracks*)

First-order independence condition

- **Independence**

$$P(\sigma(i) = k_1 \,\text{and}\, \sigma(j) = k_2)$$

$$\parallel$$

$$\boxed{P(\sigma(i) = k_1) \cdot P(\sigma(j) = k_2)}$$

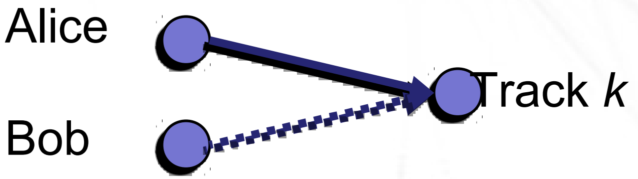
Product of first-order marginals

- **Mutual Exclusivity**
  - “Alice and Bob not **both** at Track

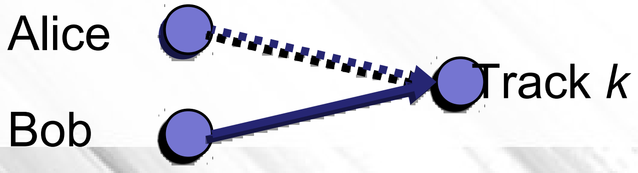
$$P(\sigma(i) = k \,\text{and}\, \sigma(j) = k)$$

$$\boxed{= 0}$$

$$\boxed{P(\sigma(i) = k)P(\sigma(j) = k) = 0}$$

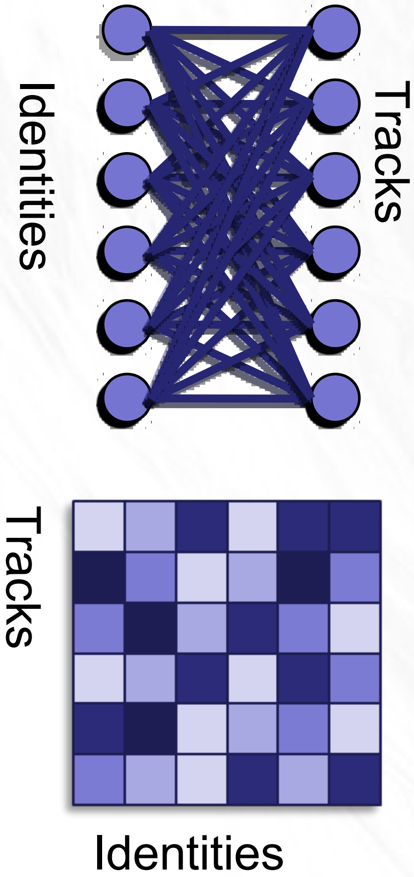


or



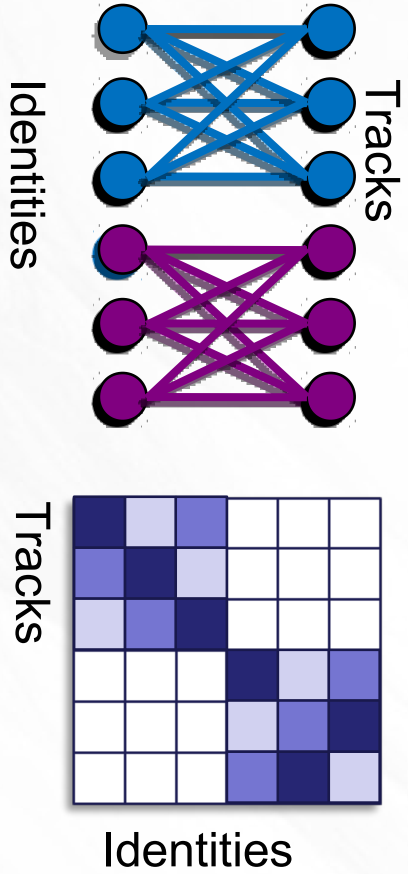
First-order independence condition

Not independent



vs.

Independent

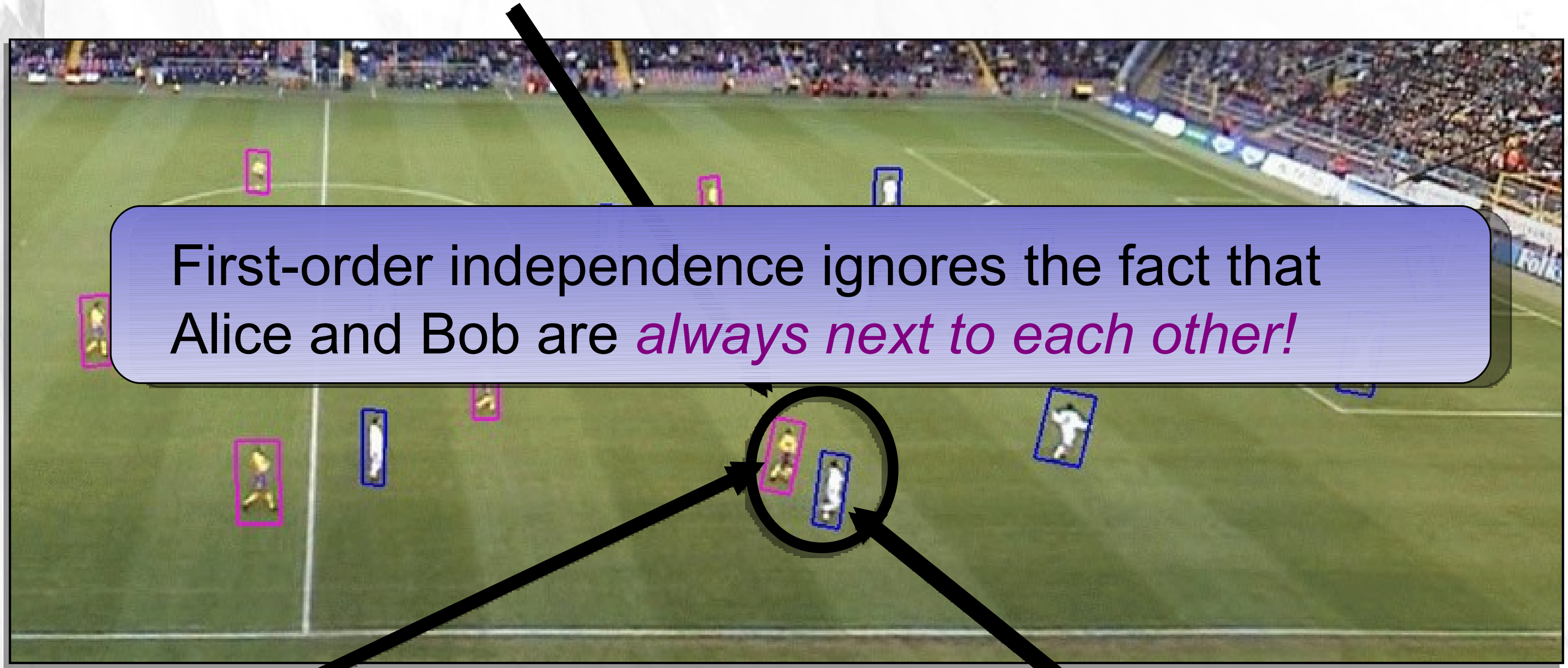


Can verify condition using first-order marginals

First-order independence

- First-order condition is insufficient:

“Alice guards Bob”



First-order independence ignores the fact that Alice and Bob are *always next to each other!*

“Alice is in red team”

“Bob is in blue team”

image from [sullivan06]

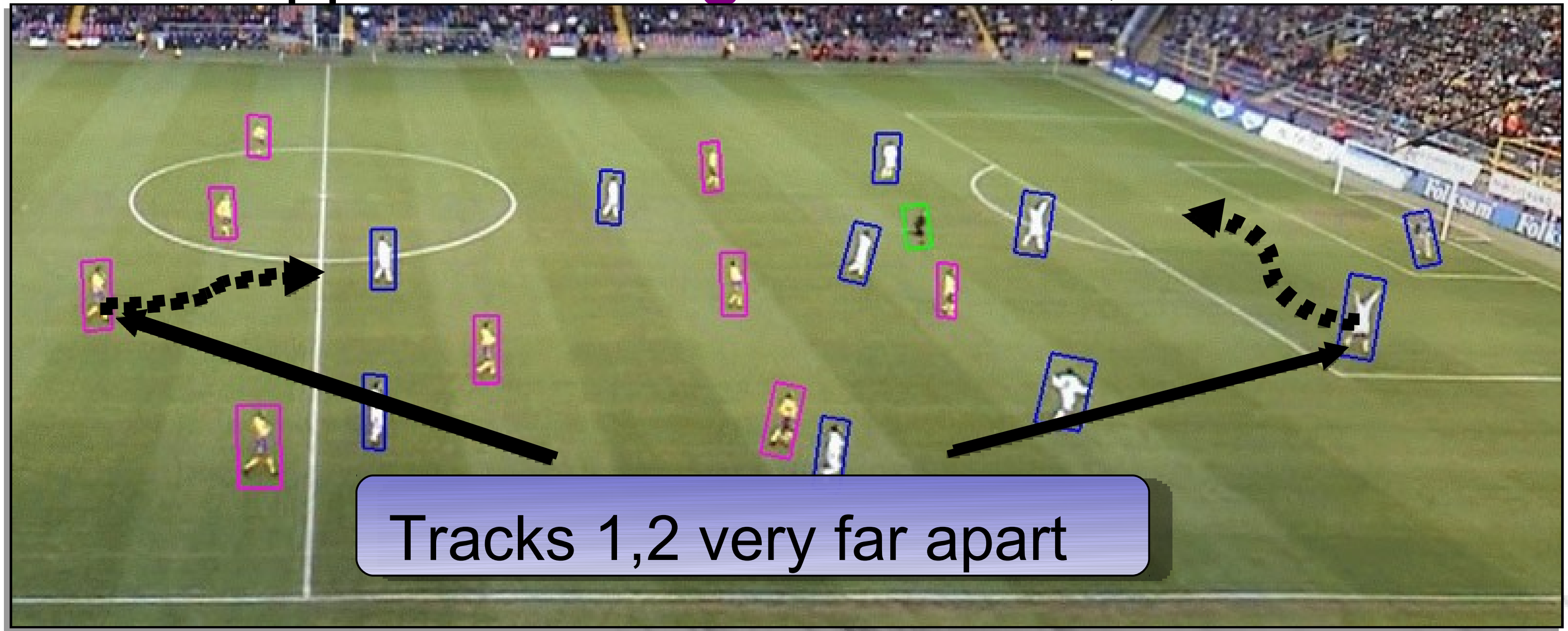
The problem with first-order

- First-order marginals look like:

Can write as *second-order* marginal:

P({Alice,Bob} occupy Tracks {1,2}) = 0

- Now suppose **Alice guards Bob**, and…



Tracks 1,2 very far apart

Second-order summaries

- Store summaries for ordered pairs:

Identities

Tracks

	(A,B)	(B,A)	(A,C)	(C,A)
(1,2)	1/6	1/12	1/8	1/12
(2,1)	1/12	1/6	1/12	1/8
(1,3)	1/12	1/12	1/8	1/24
(3,1)	1/12	1/12	1/24	1/8

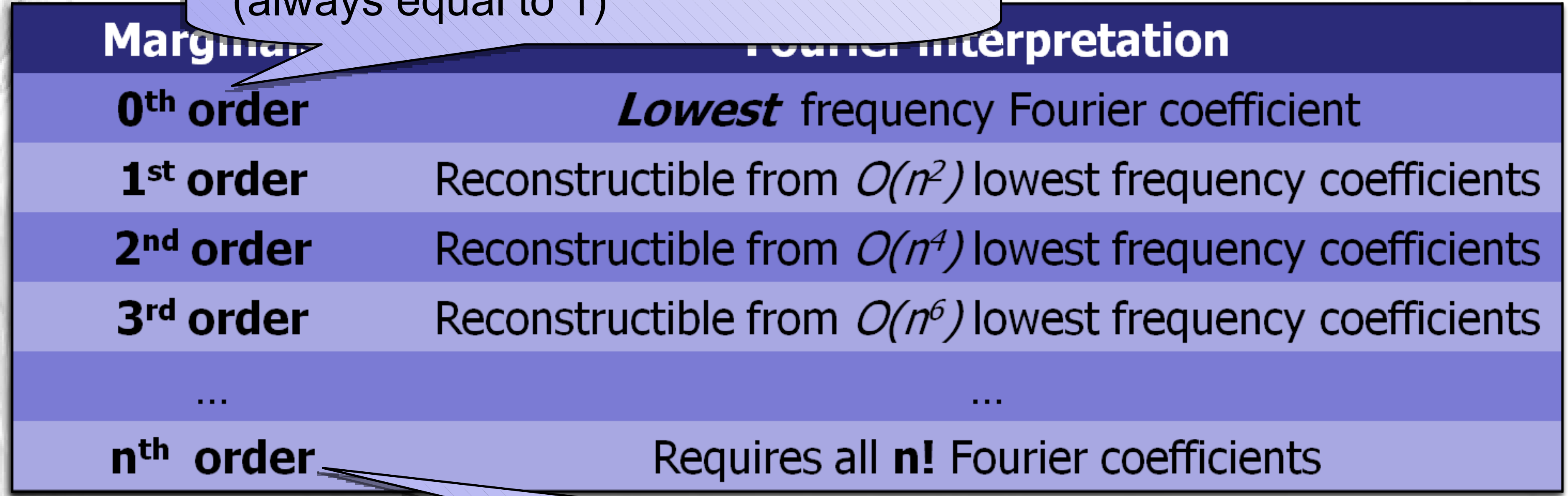
“Bob is at Track 1
and
Alice is at Track 3
with probability 1/12”

store marginal probability that identities (k,l) map to tracks (i,j)

- 2nd order summary requires $O(n^4)$ storage

Higher orders and connections to Fourier

Sum over entire distribution (always equal to 1)



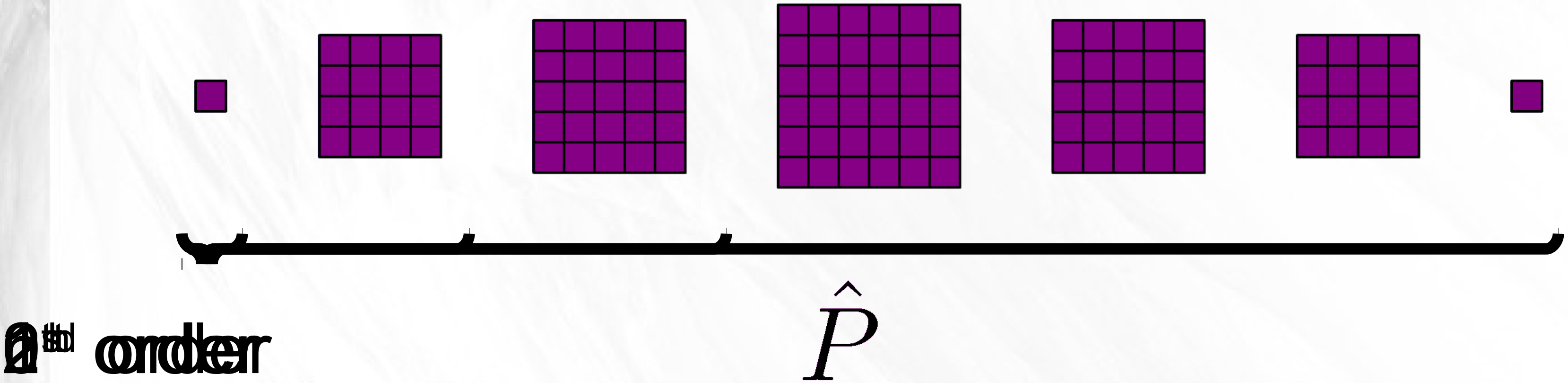
Marginal	Fourier interpretation
0th order	***Lowest*** frequency Fourier coefficient
1st order	Reconstructible from $O(n^2)$ lowest frequency coefficients
2nd order	Reconstructible from $O(n^4)$ lowest frequency coefficients
3rd order	Reconstructible from $O(n^6)$ lowest frequency coefficients
...	...
nth order	Requires all **n!** Fourier coefficients

Recovers original distribution, requires storing n! numbers

- **Trade-of** toring more numbers
- **Remark**: high-order marginals contain low-order information

Fourier coefficient matrices

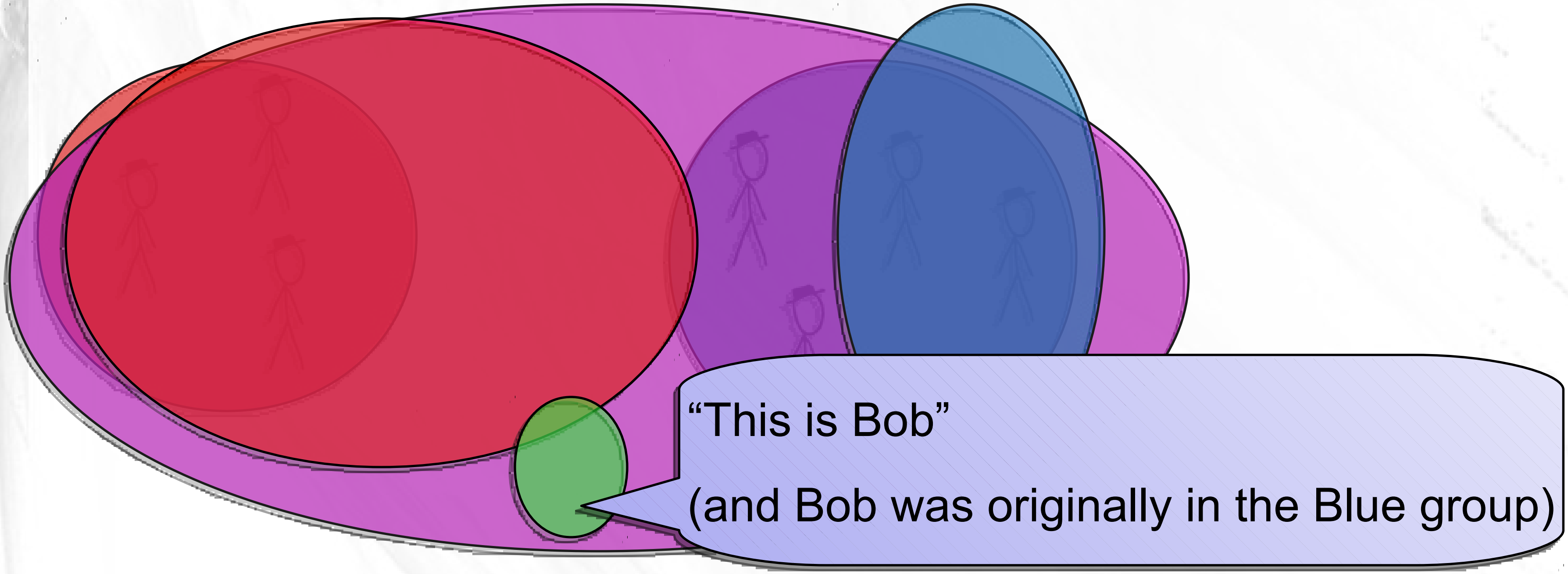
- Fourier coefficients on permutations are a collection of square matrices ordered by "frequency":



- **Bandlimiting** - keep a truncated set of coefficients
- **Fourier domain inference** – prediction/conditioning in the Fourier domain
  - [Kondor et al,AISTATS07]
  - [Huang et al,NIPS07]

Back to independence

- Need to consider two operations



- Groups **join** when tracks from two groups mix
- Groups **split** when an observation allows us to reason over smaller groups independently

Problems

- If the joint distribution *h* factors as a product of distributions *f* and *g:*

$$h(\sigma) = f(\sigma) \cdot g(\sigma)$$

Distribution over tracks {1,…,p}

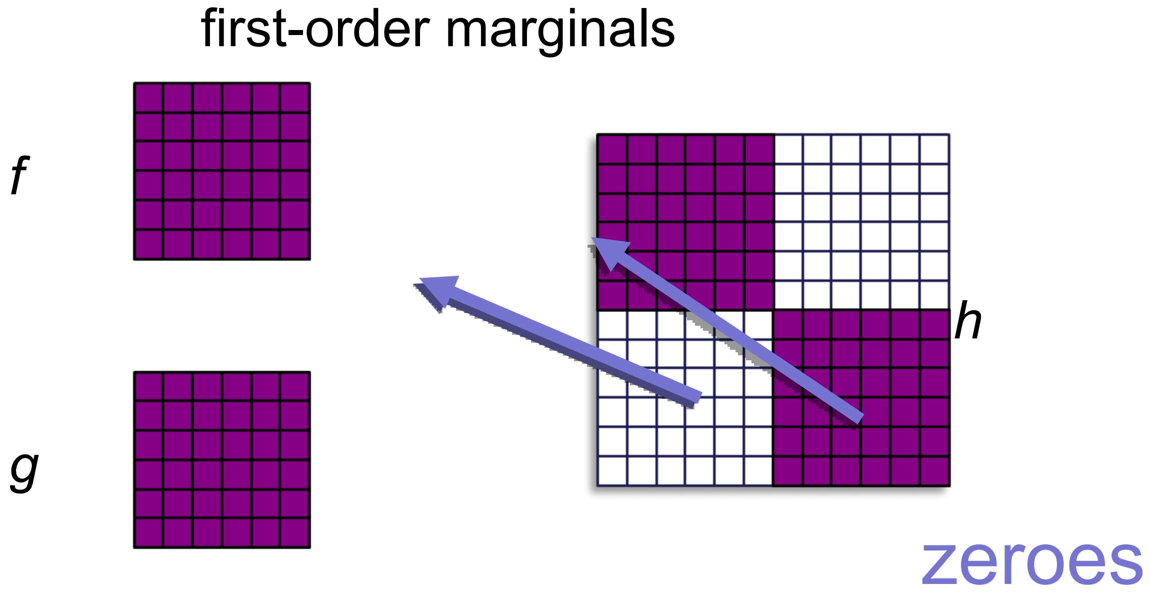
Distribution over tracks {p+1,…,n}

(Join problem) Find Fourier coefficients of the joint *h* given Fourier coefficients of factors *f* and *g*?

(Split problem) Find Fourier coefficients of factors *f* and *g* given Fourier coefficients of the joint *h*?

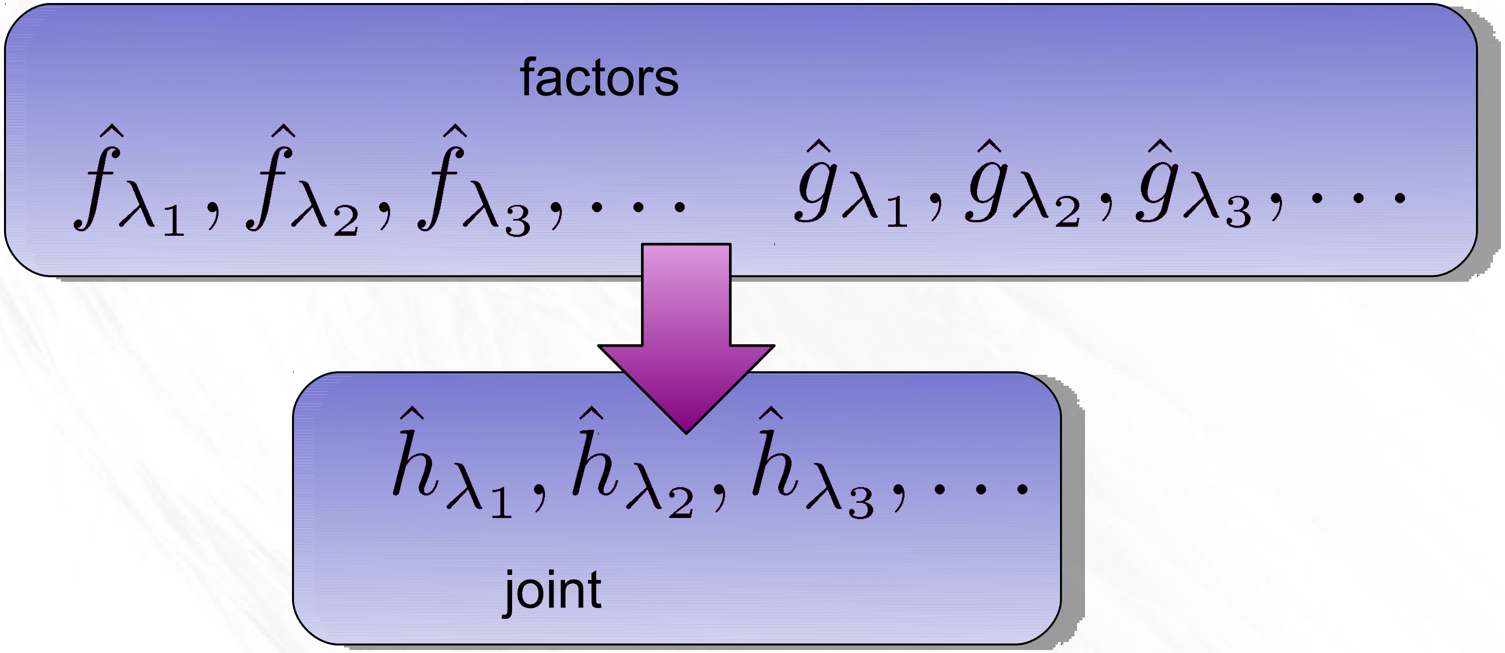
First-order join

- Given first-order marginals of *f* and *g*, what does the matrix of first-order marginals of *h* look like?



Higher-order joining

- Given Fourier coefficients of the factors *f* and *g* at each frequency level:



factors

$$\hat{f}_{\lambda_1}, \hat{f}_{\lambda_2}, \hat{f}_{\lambda_3}, \ldots \qquad \hat{g}_{\lambda_1}, \hat{g}_{\lambda_2}, \hat{g}_{\lambda_3}, \ldots$$

$$\hat{h}_{\lambda_1}, \hat{h}_{\lambda_2}, \hat{h}_{\lambda_3}, \ldots$$

joint

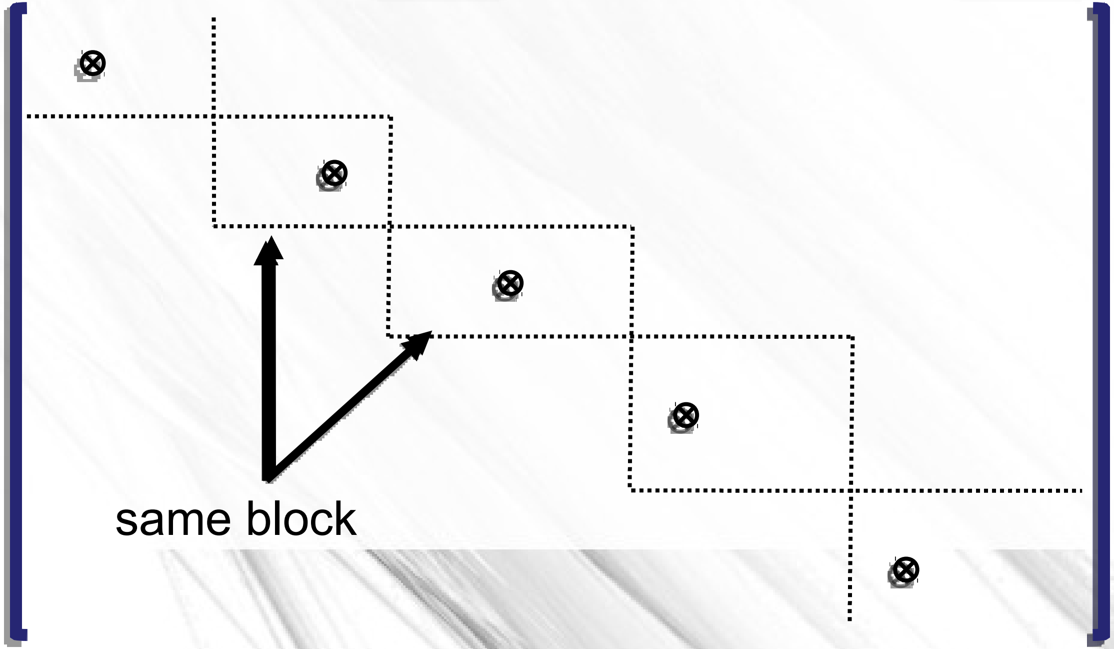
- Compute Fourier coefficients of the joint distribution *h* at each frequency level

Higher-order joining

- Joining for higher-order coefficients gives similar block-diagonal structure
  - Also get *Kronecker product structure* for each block

Blocks appear multiple times (multiplicities related to *Littlewood-Richardson coefficients*)

$$\hat{h}_\lambda = L_\lambda^T \cdot$$



$$\cdot L_\lambda$$

Problems

- If the joint distribution *h* factors as a product of distributions *f* and *g:*

$$h(\sigma) = f(\sigma) \cdot g(\sigma)$$

Distribution over tracks {1,…,p}

Distribution over tracks {p+1,…,n}

(Join problem) Find Fourier coefficients of the joint *h* given Fourier coefficients of factors *f* and *g*?

(Split problem) Find Fourier coefficients of factors *f* and *g* given Fourier coefficients of the joint *h*?

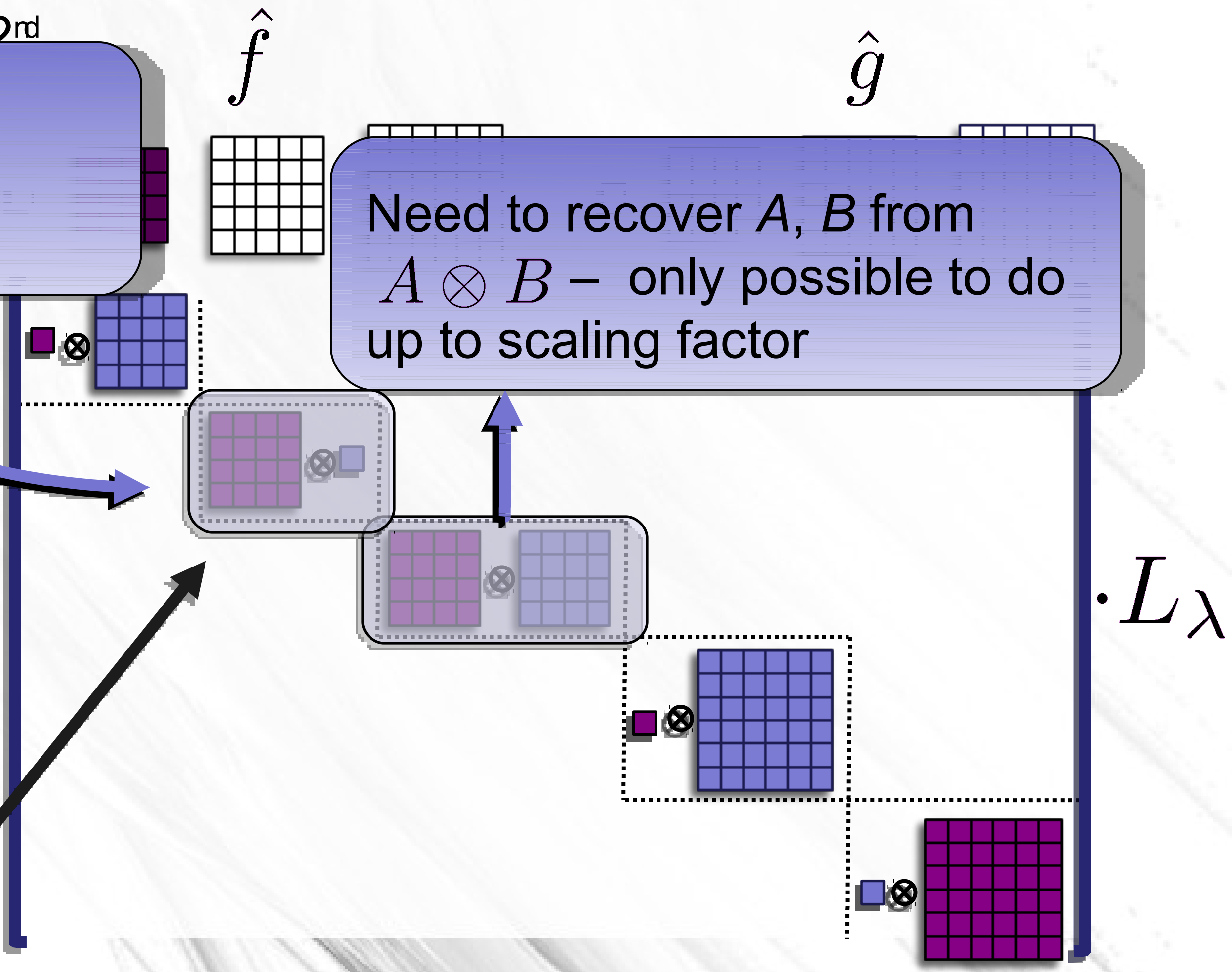
Splitting

- Want to “invert” the Join process:

Consider recovering 2nd Fourier block

Our approach: search for blocks of the form:

$\hat{f}_\lambda \otimes 1 \qquad 1 \otimes \hat{g}_\lambda$



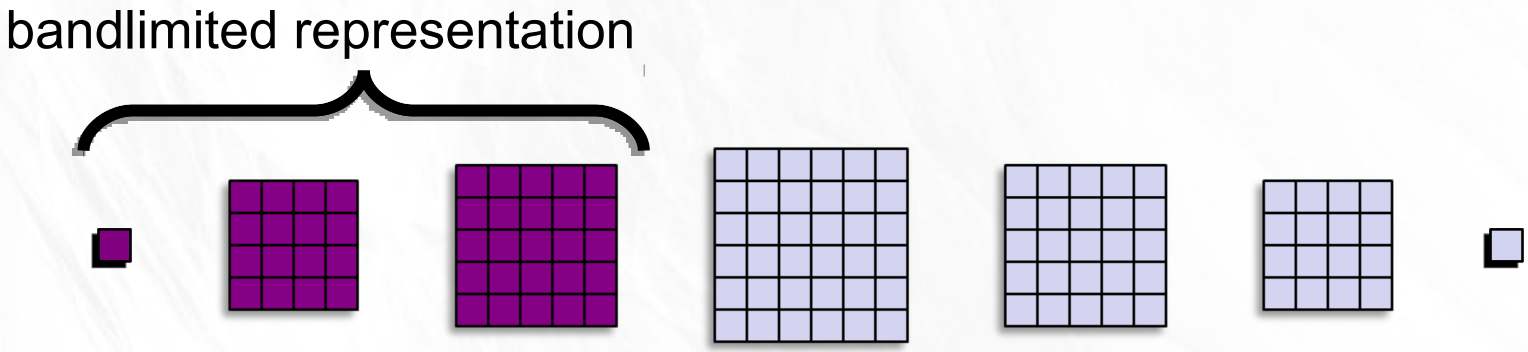
$$\hat{h}_\lambda = L_\lambda^T \cdot$$

Need to recover *A*, *B* from $A \otimes B$ – only possible to do up to scaling factor

Theorem: these blocks always exist! (and are efficient to find)

Marginal preservation

- **Problem**: In practice, never have entire set of Fourier coefficients!



- **Marginal preservation guarantee**:

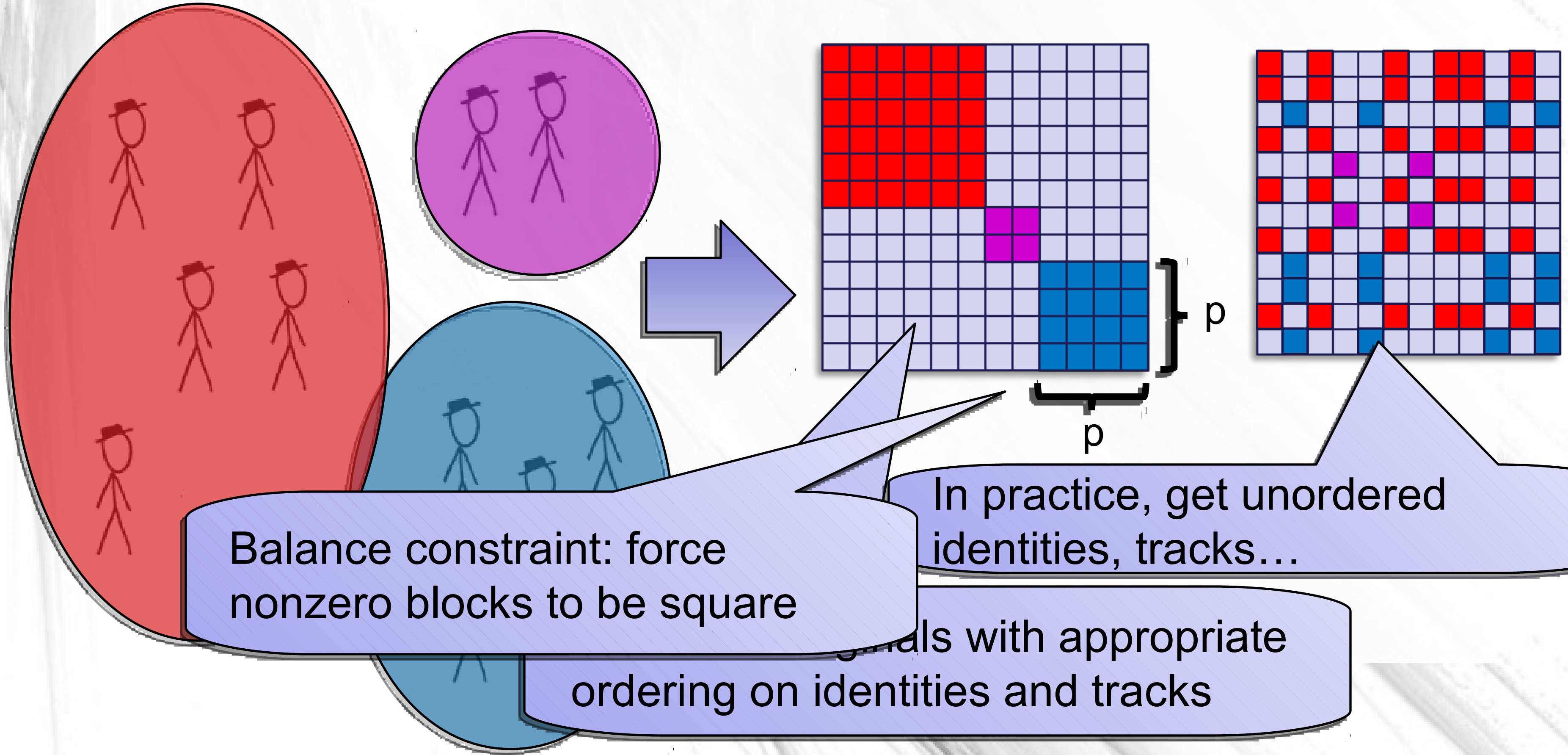
Theorem: *Given m^{th}-order marginals for independent factors, we* exactly *recover m^{th}-order marginals for the joint.*

  - Conversely, get a similar guarantee for splitting
  - (Usually get some higher order information too)

Detecting independence

- To adaptively split large distributions, need to detect

Can use (bi)clustering on matrix of marginals to discover an appropriate ordering!



In practice, get unordered identities, tracks…

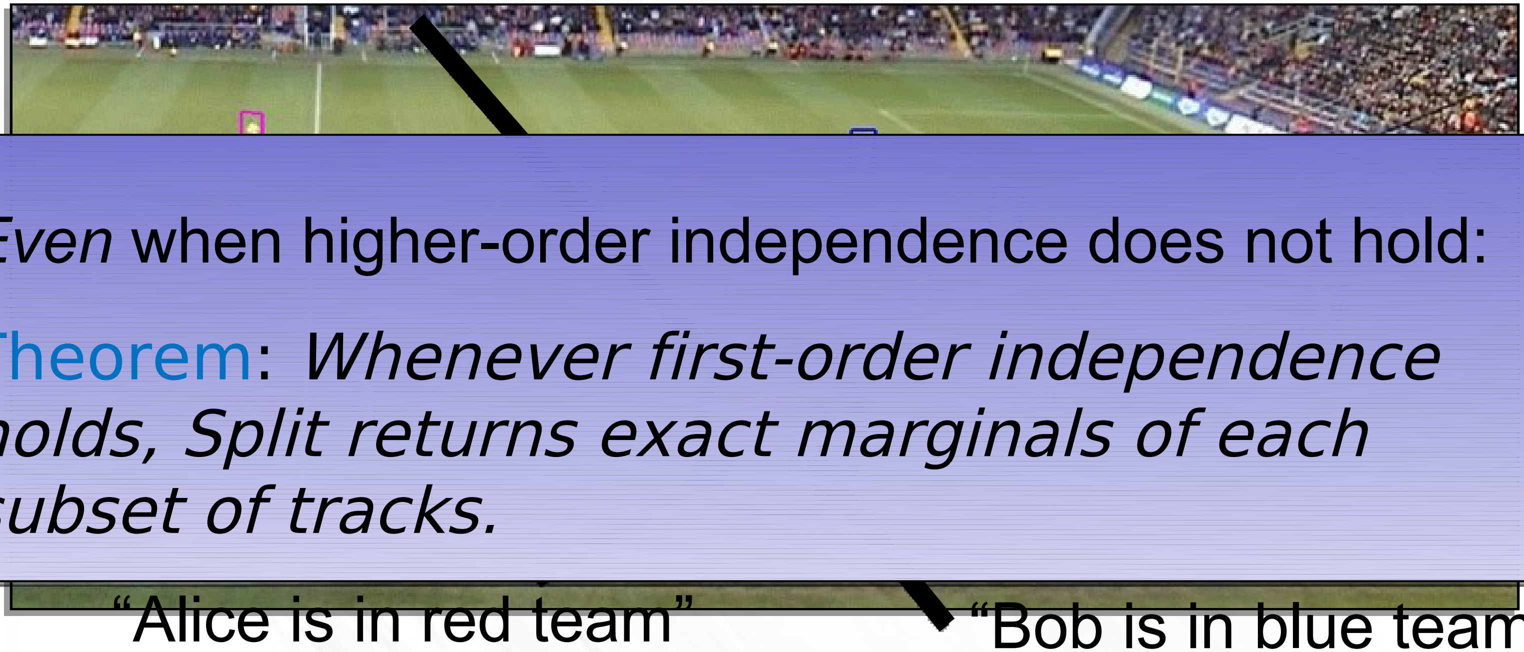
Balance constraint: force nonzero blocks to be square

als with appropriate ordering on identities and tracks

First-order independence

- First-order condition is insufficient:

“Alice guards Bob”



Even when higher-order independence does not hold:

Theorem: *Whenever first-order independence holds, Split returns exact marginals of each subset of tracks.*

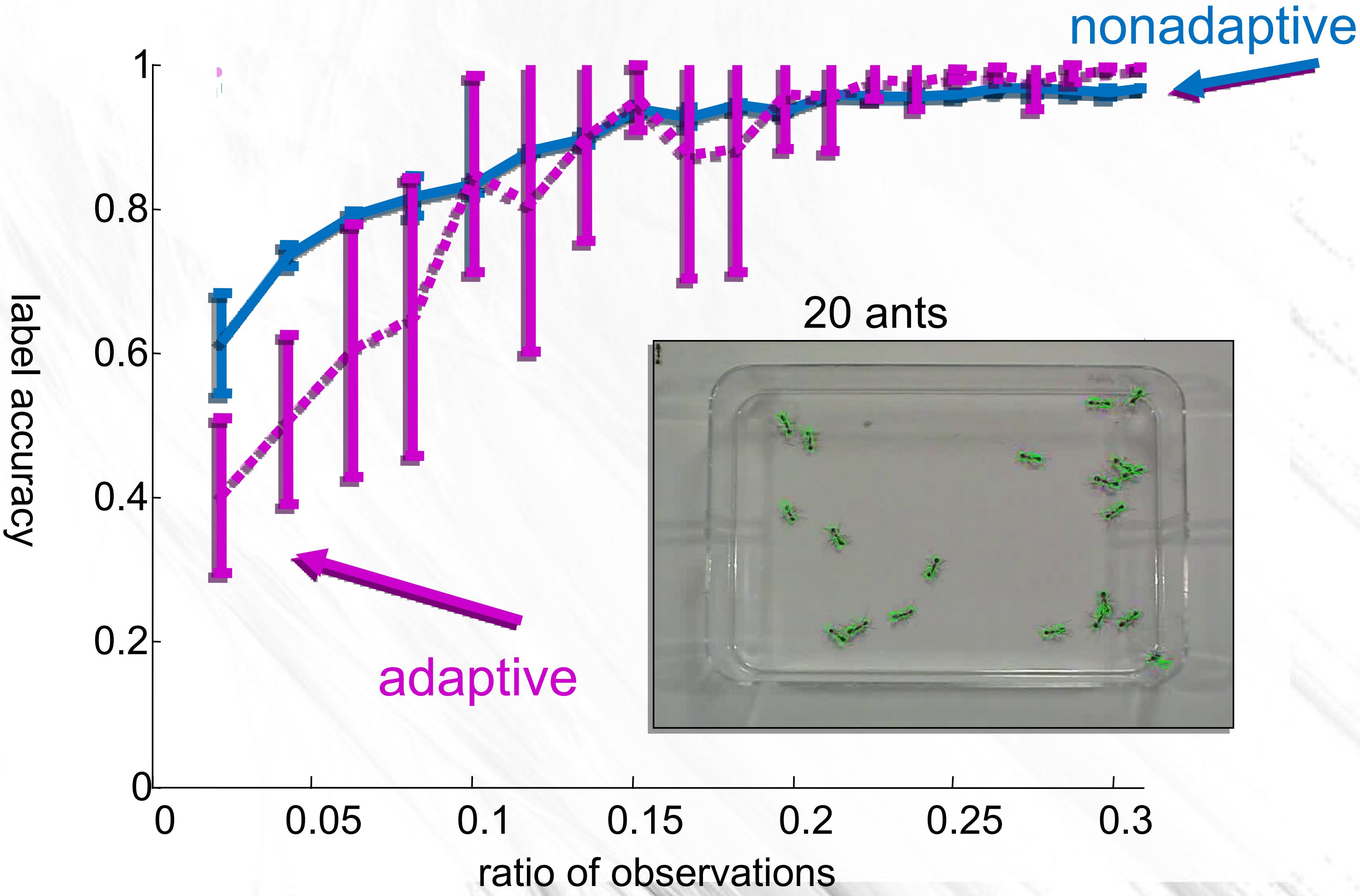
“Alice is in red team”

“Bob is in blue team”

Can check for higher order independence after detecting at first-order

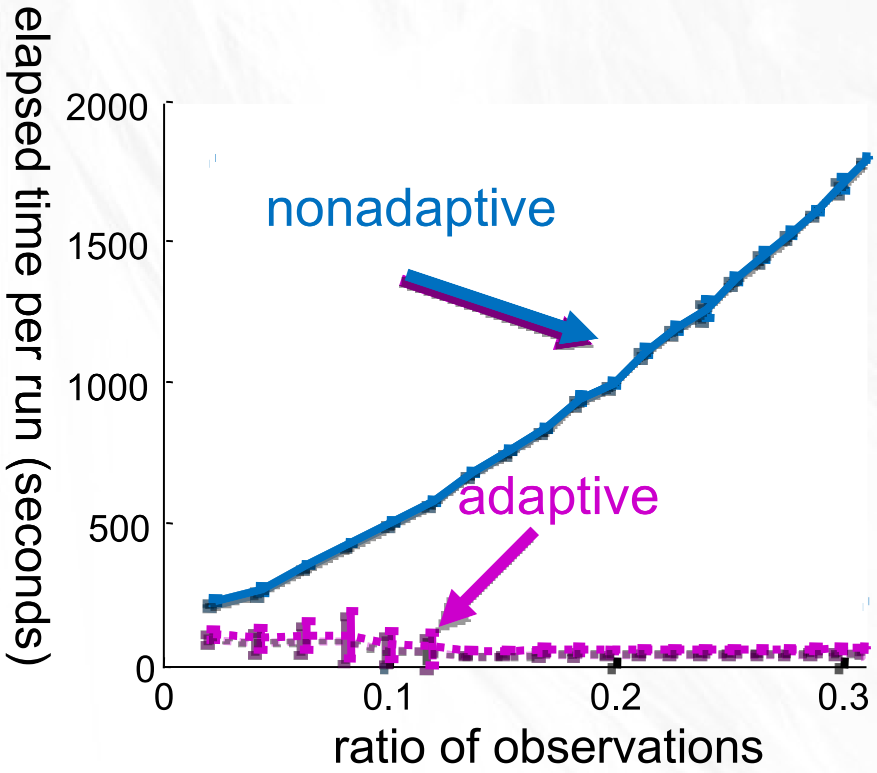
- What if we call *Split* when ***only the first-order condition*** is satisfied?

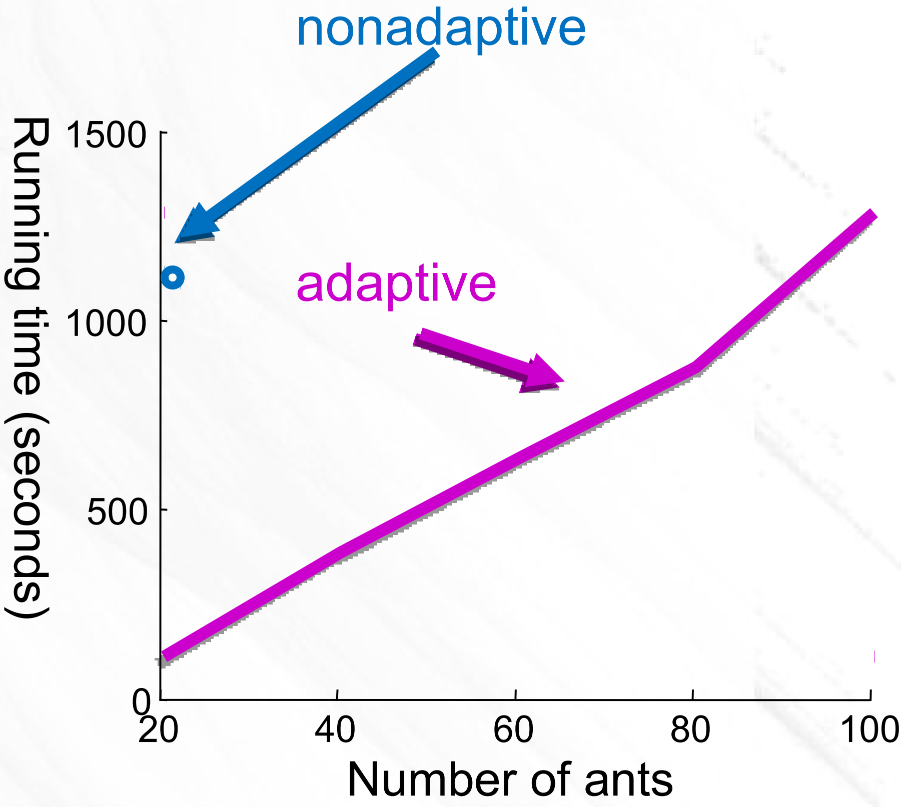
Experiments - accuracy



dataset from [Khan et al. 2006]

Experiments – running time





Final Conclusions

Scalable and adaptive identity management algorithm to track up to n=100 objects

Two new algorithms

marginalization,
conditioning,
join, split

Completely Fourier-theoretic characterization of probabilistic independence

Thank you !