THE ROBOTICS INSTITUTE

Adaptive Fourier Domain Inference on the Symmetric Group

Jonathan Huang

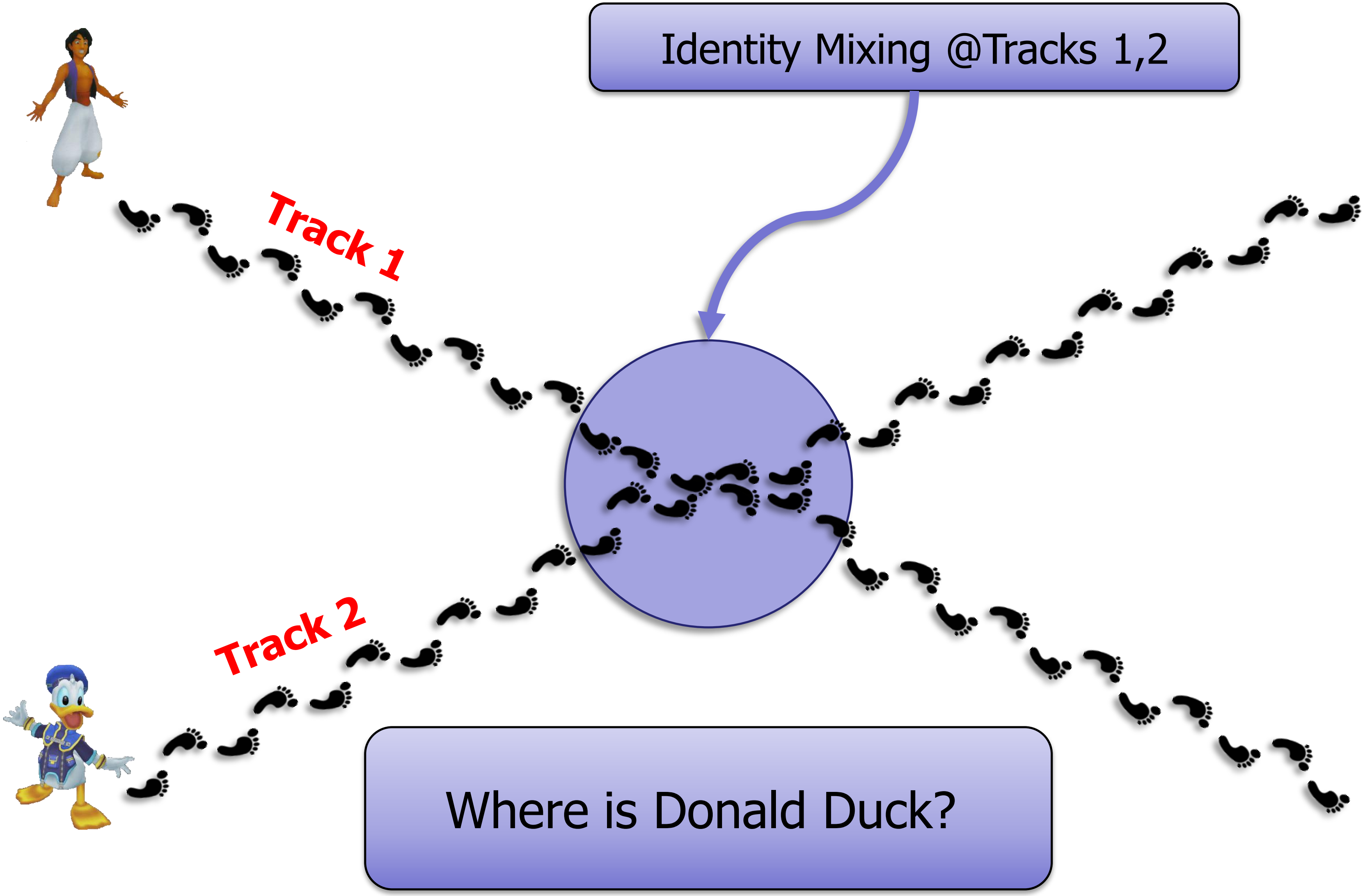
NIPS Algebraic Methods Workshop (AML ‘08)
12/11/08

Joint work with Carlos Guestrin,
Xiaoye Jiang, Leonidas Guibas

Select Lab

Carnegie Mellon

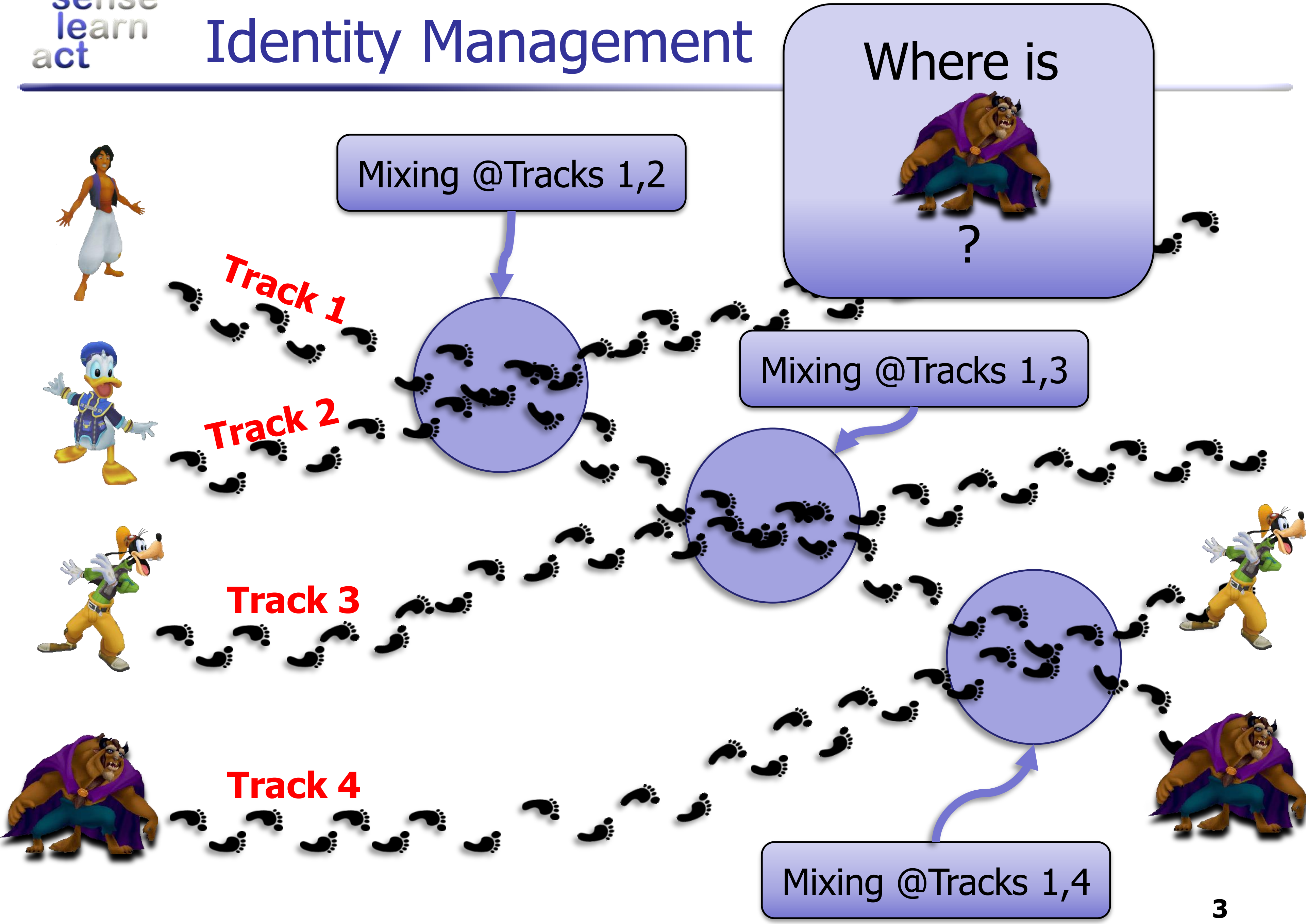
Identity Management [Shin et al., ‘03]



Identity Mixing @Tracks 1,2

Where is Donald Duck?

Identity Management



Where is

?

Mixing @Tracks 1,2

Mixing @Tracks 1,3

Mixing @Tracks 1,4

Reasoning with Permutations

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

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

[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

Storage Complexity

- There are **n!** permutations!

n	n!	Memory required to store n! doubles
9	362,880	3 megabytes
12	4.8×10^8	9.5 terabyes
15	1.31×10^{12}	1729 petabytes (!!)

x 1,800,000

- Graphical models not effective due to mutual exclusivity constraints ("**A**lice and **B**ob cannot both be at Track **1** simultaneously")
  - One such constraint for each pair of identities

1st order summaries

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

Identities

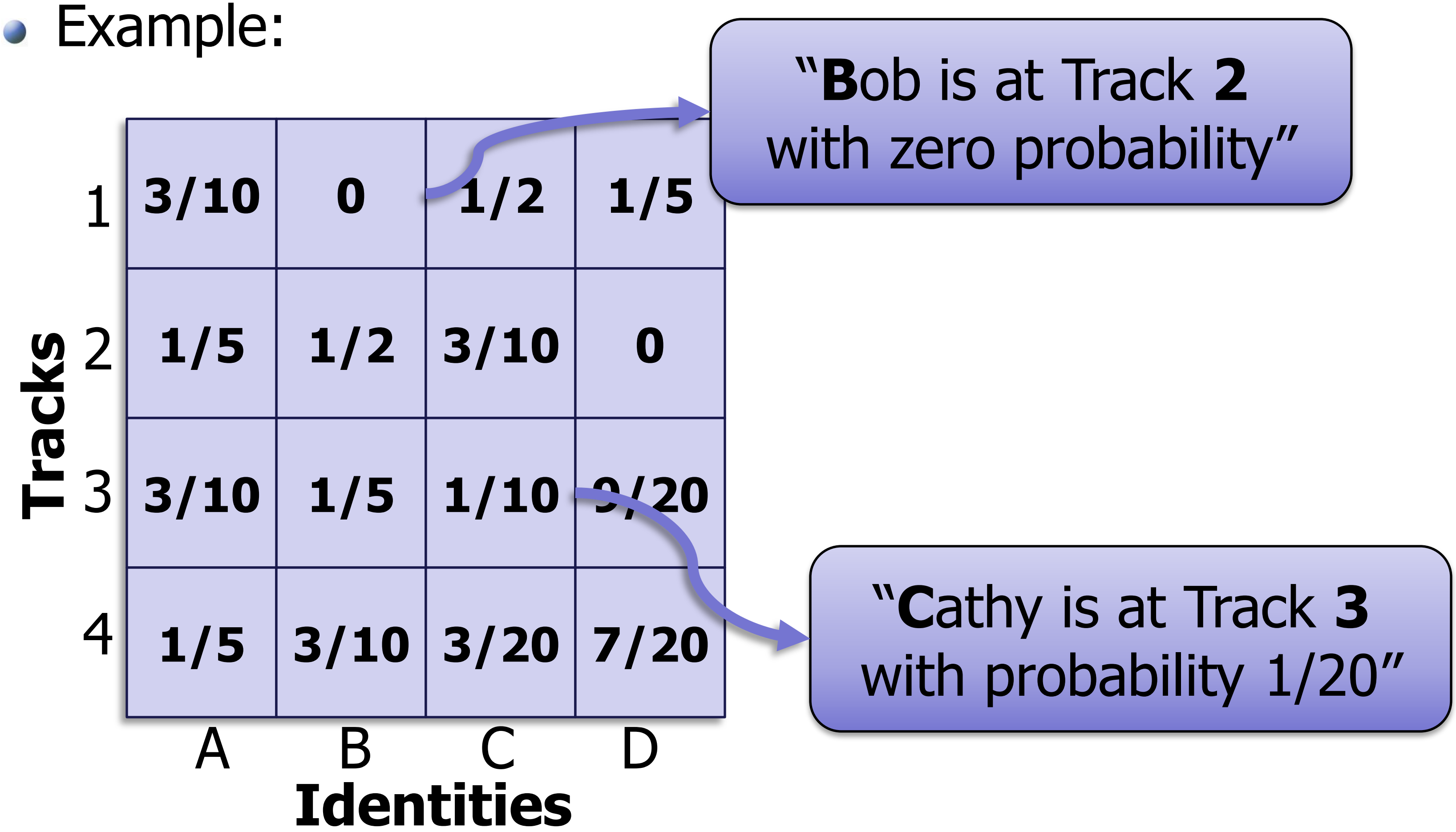
Track Permutations

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

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



Tracks

	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**	**1/10**	**9/20**
4	**1/5**	**3/10**	**3/20**	**7/20**

Identities

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

2^{nd} order summaries

- Idea #2: store marginal probabilities that **ordered 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

"**C**athy is Track **3**
and
David is in Track **4**
with zero probability"

2nd order summaries

- Can also 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**

"**B**ob is at Track **1**
and
Alice is at Track **3**
with probability 1/12"

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

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)
  - By the way, the **0th-order** marginal is the normalization constant (which equals 1)

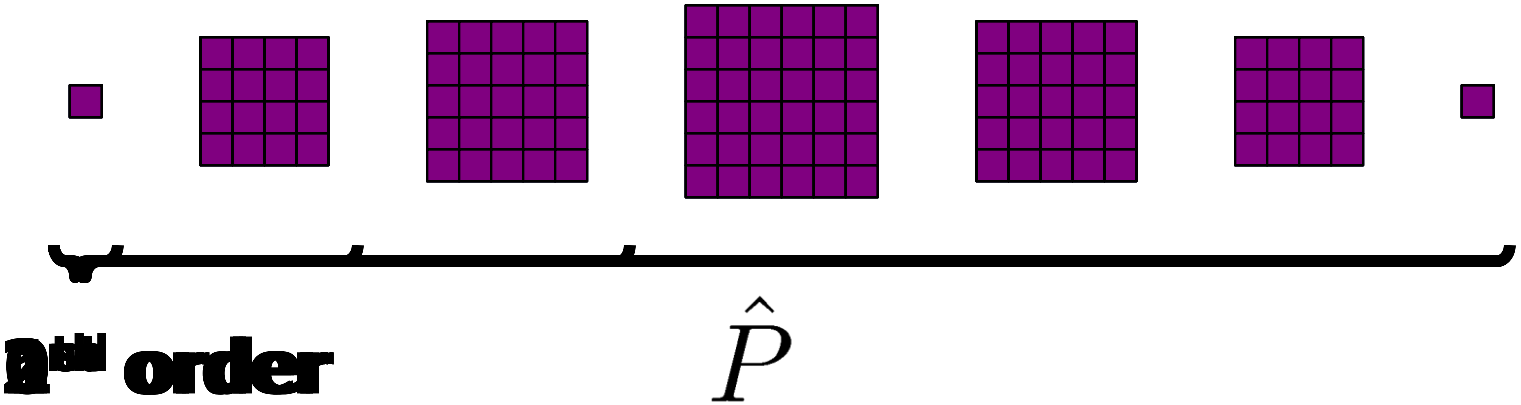
- **Fundamental Trade-off**: can capture higher-order dependencies at the cost of storing more numbers

The Fourier interpretation

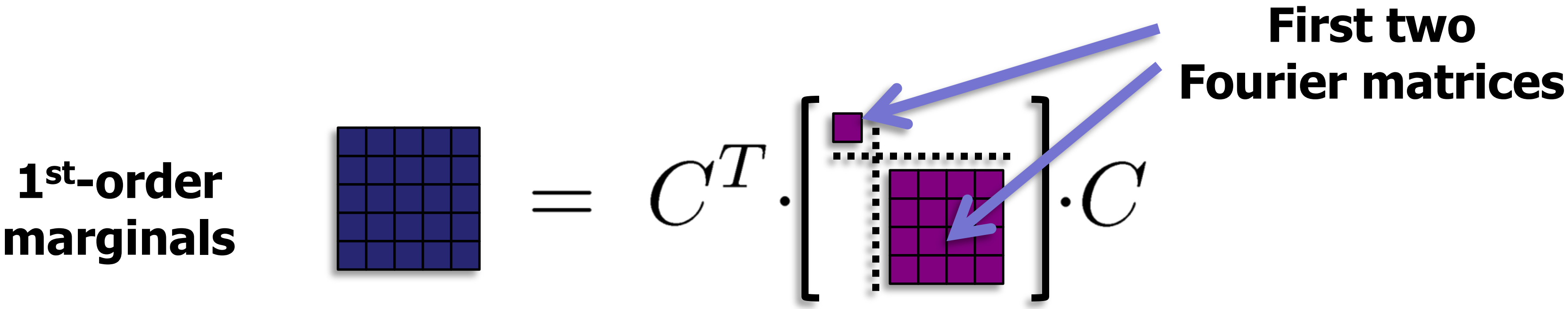
- Marginal summaries are connected to Fourier analysis!
  - Used for multi-object tracking [Kondor et al, '07]
- Simple marginals are **"low-frequency"**: intuitively,
  - **1st order marginals** are the *lowest frequency* responses (except for DC component)
  - **2nd order marginals** contain higher frequencies than 1st order marginals
  - **3rd order marginals** contain still higher frequency information
- Note that higher-order marginals can contain lower-order information

Fourier coefficient matrices

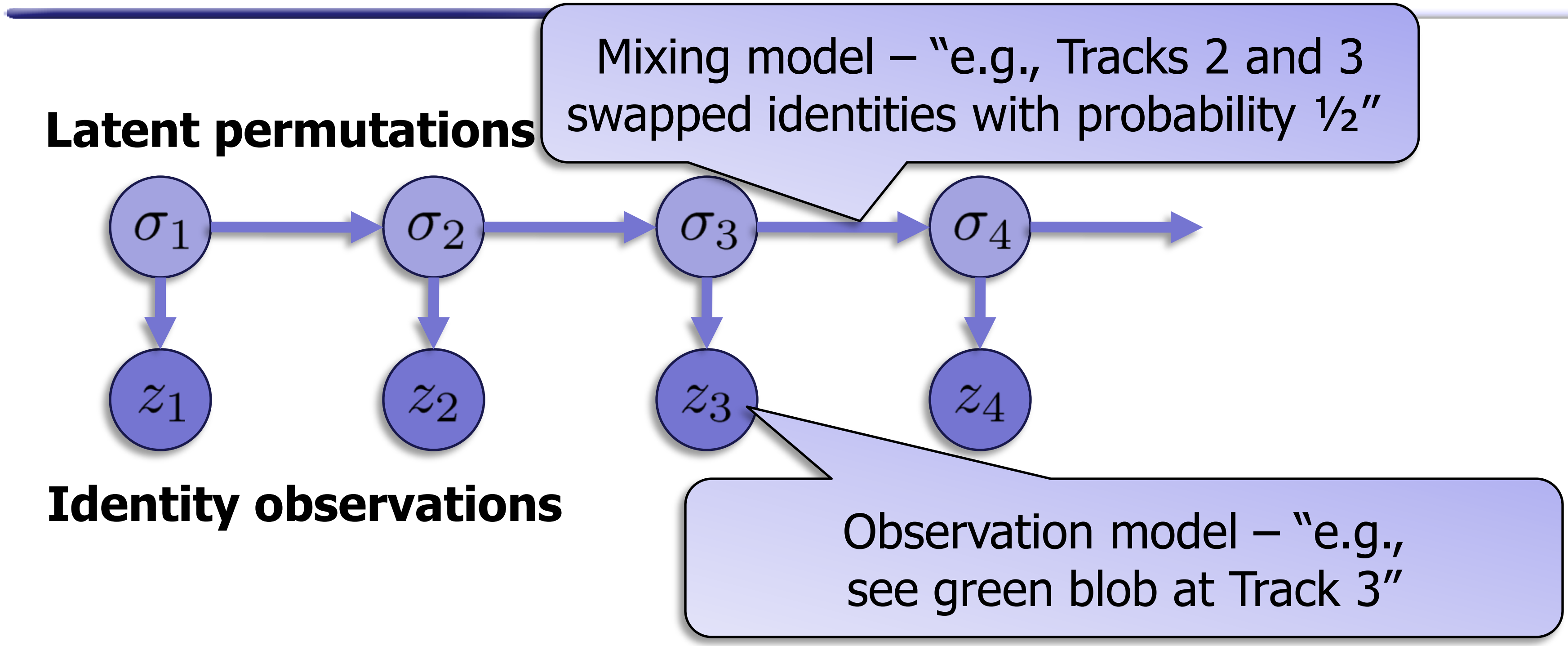
- Fourier coefficients on permutations are given as a collection of square matrices ordered by “frequency”:



- Marginals are constructed by conjugating Fourier coefficient matrices by a (pre-computed) constant matrix:



Hidden Markov Model Inference



Latent permutations

Mixing model – "e.g., Tracks 2 and 3 swapped identities with probability ½"

Identity observations

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 rewrite all inference operations completely in the Fourier domain**

Hidden Markov model inference

- **Two basic inference operations** for HMMs:

(Prediction/Rollup)

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

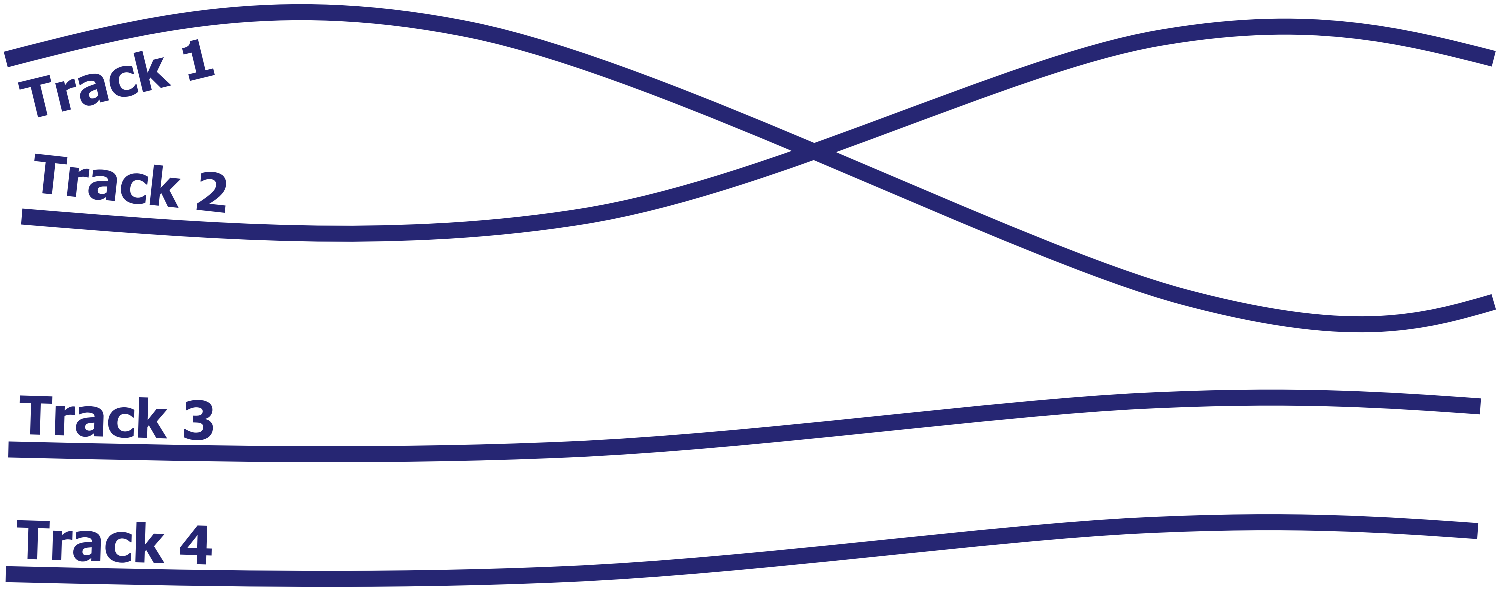
(Conditioning)

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

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

Random walk transition model

- We assume that σ_{t+1} is generated by the rule:
  - Draw $\tau \sim$ **Q(τ)** ← **Mixing Model**
  - Set $\sigma_{t+1} = \tau \cdot \sigma_t$
- For example, **Q([2 1 3 4])=½** means that Tracks **1** and **2** swapped identities with probability ½



Prediction/Rollup

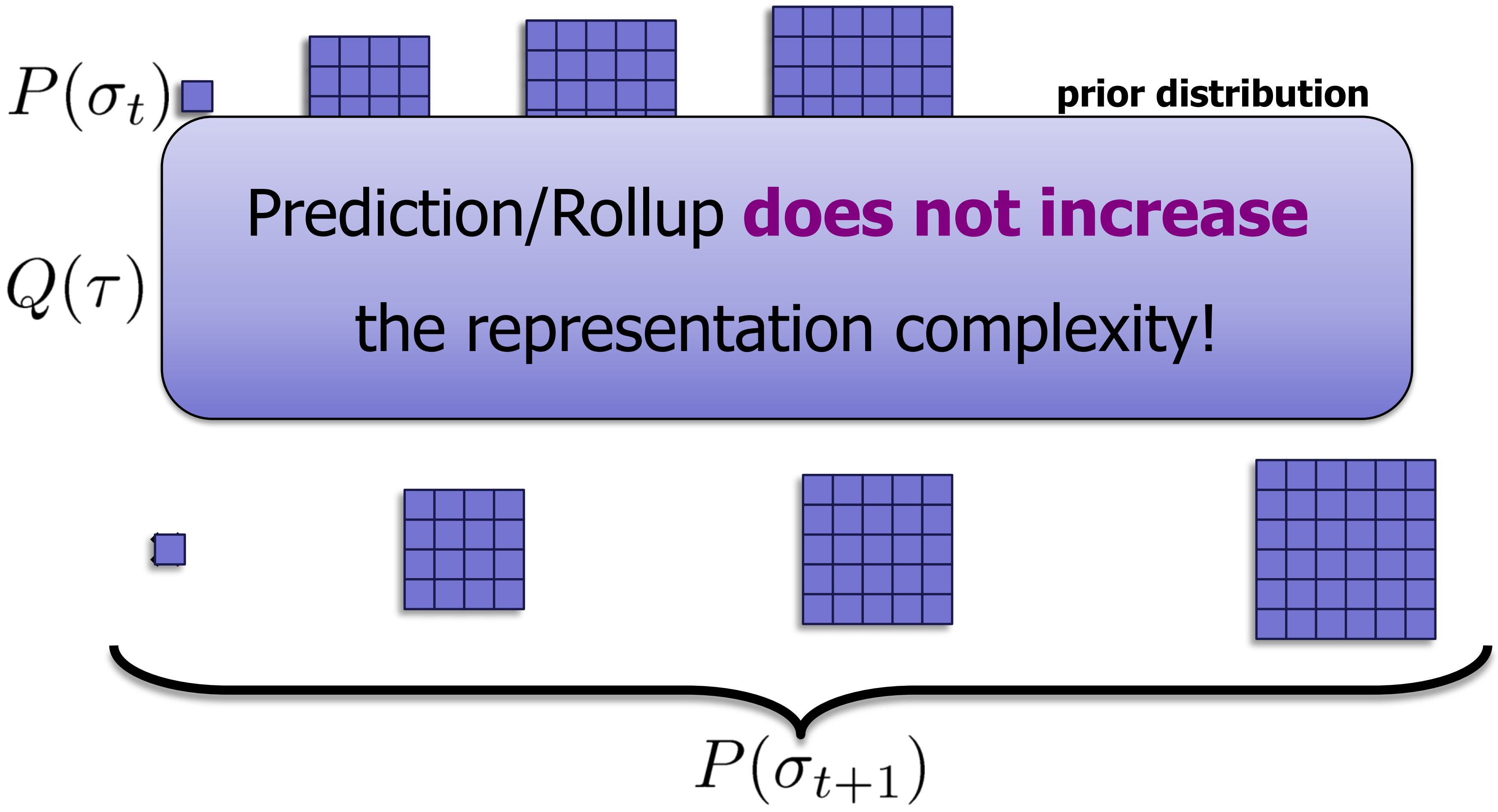
- Inputs:
    - Prior distribution **$P(\sigma_t)$**
    - Mixing Model **$Q(\tau)$**

- Prediction/Rollup can be written as a **convolution**:

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

Fourier Domain Prediction/Rollup

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



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

Hidden Markov model inference

- **Two basic inference operations** for HMMs:

(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?

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**")

Conditioning

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

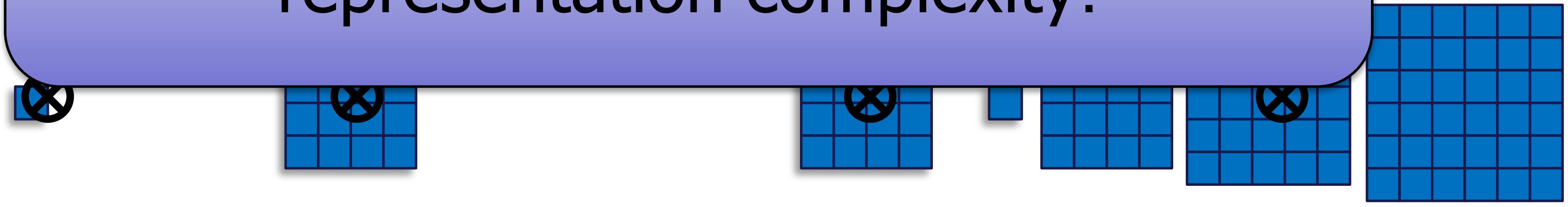
Need to store 2nd-order probabilities after conditioning!

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 **can increase**

representation complexity!



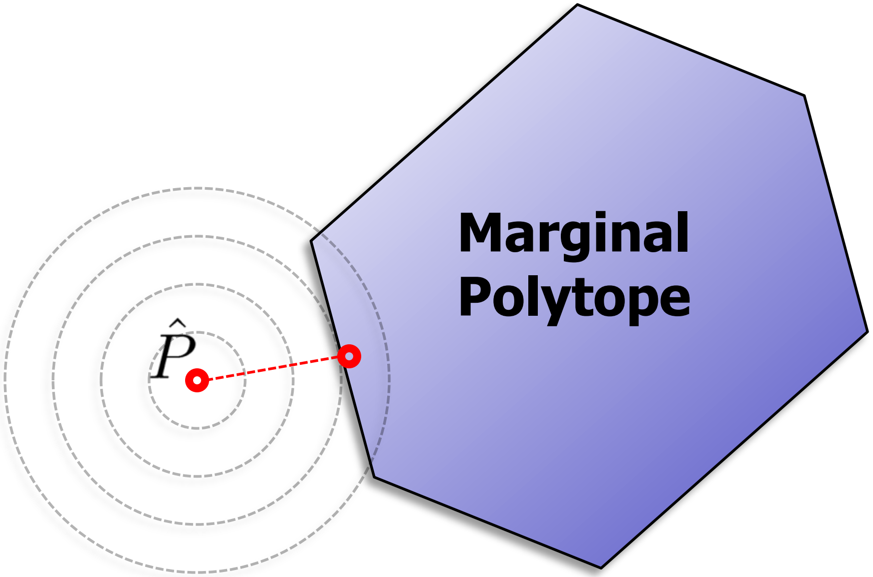
posterior

$P(\sigma_t|z)$

Dealing with bandlimiting errors

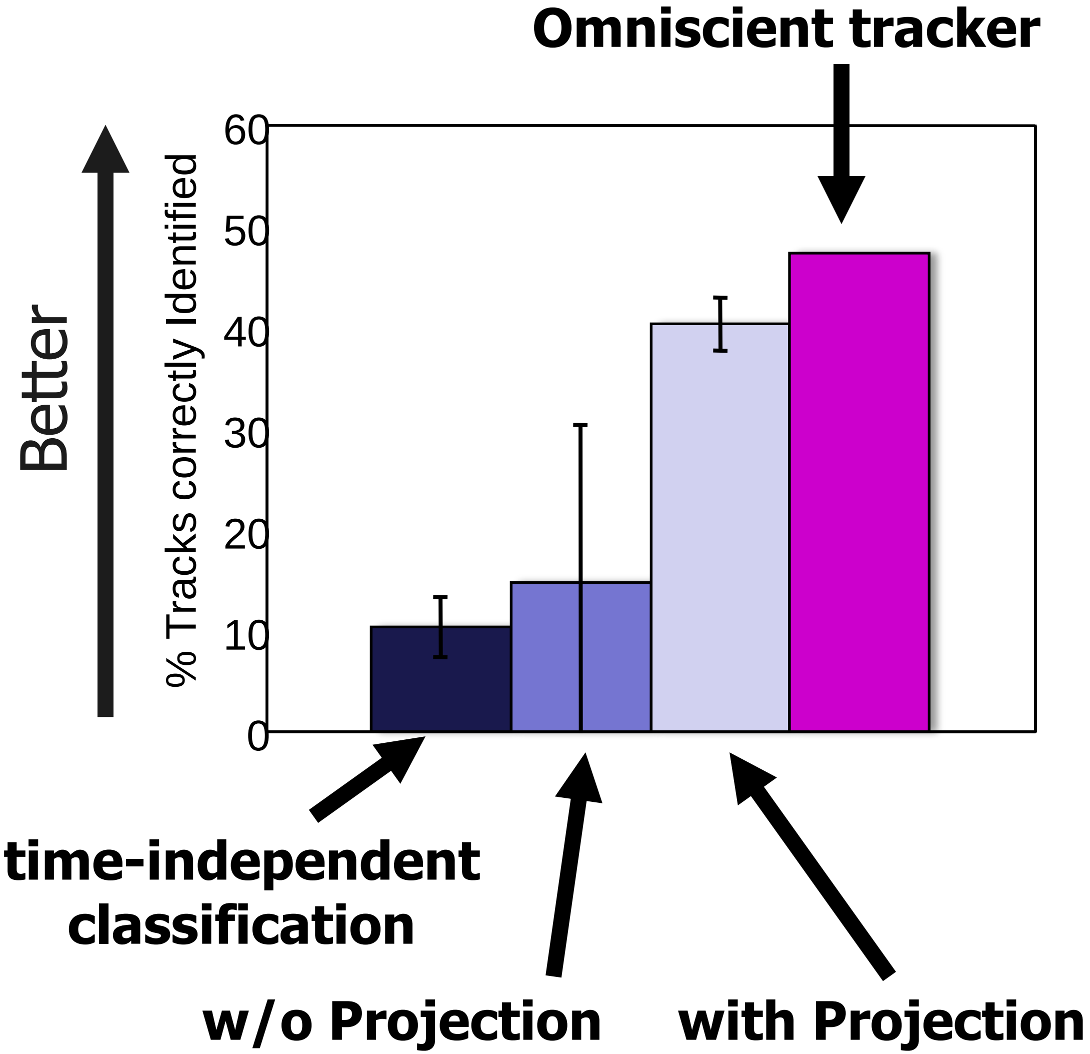
- Consecutive conditioning steps can propagate errors,
  - (sometimes causing approximate marginals to be negative!)

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



Tracking with a camera network

- **Camera Network** data:
  - **8** cameras, multi-view, occlusion effects
  - **11** individuals in lab
  - Identity observations obtained from color histograms
  - Mixing events declared when people walk close to each other



Omniscient tracker

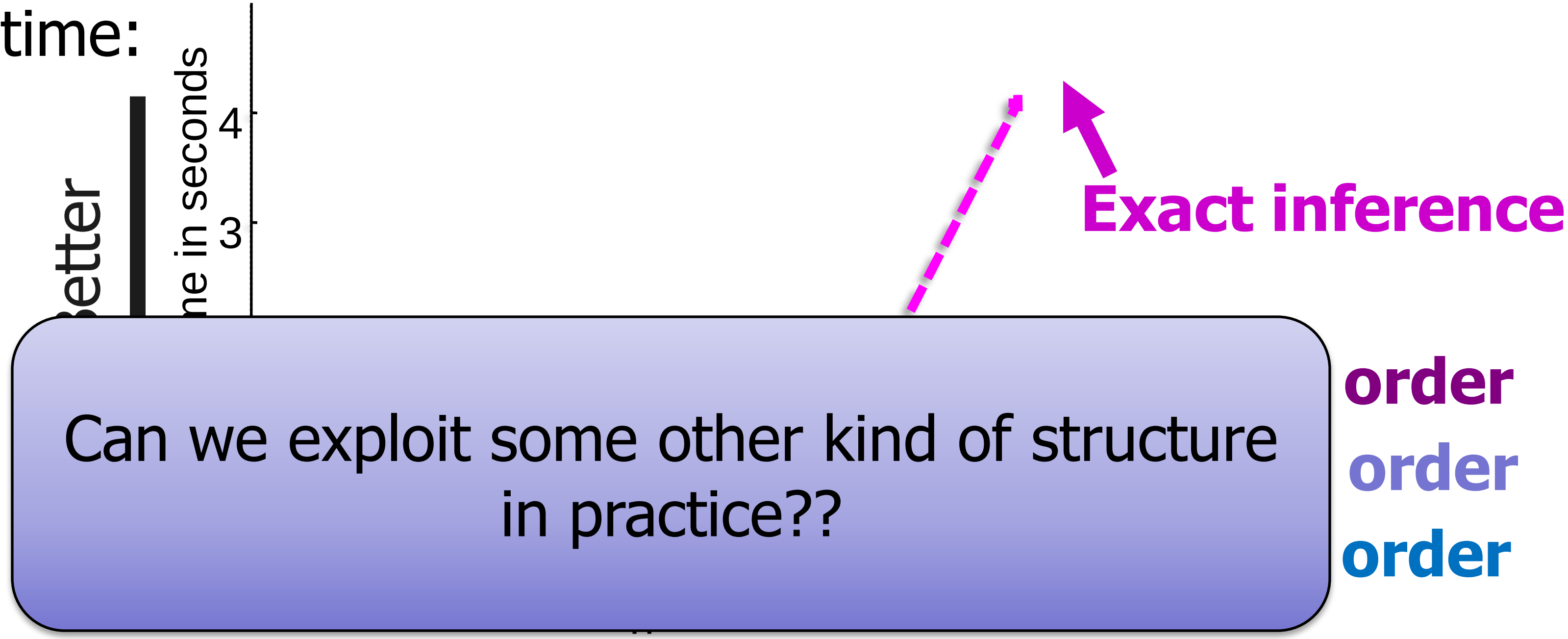
time-independent classification

w/o Projection

with Projection

Scaling

- For fixed representation depth, Fourier domain inference is polytime:



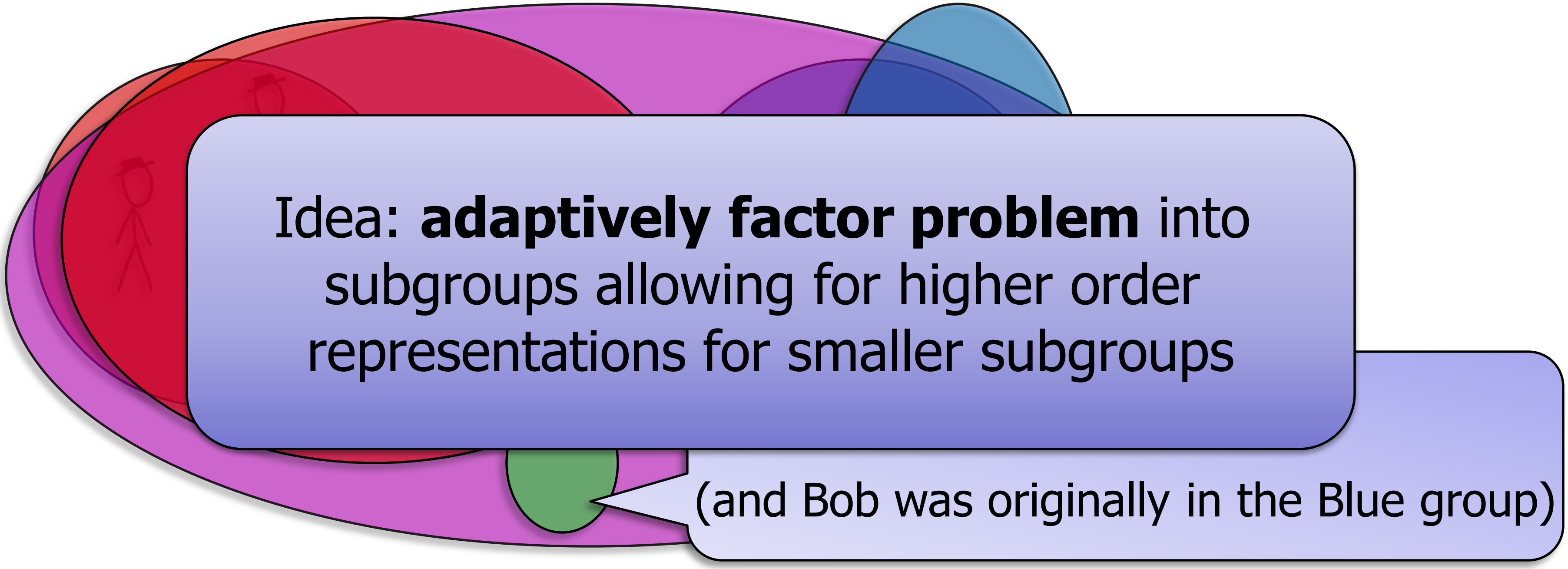
Can we exploit some other kind of structure in practice??

- But complexity can still be bad…

Representation Depth	# of Fourier coefficients
1st order	$O(n^2)$
2nd order	$O(n^4)$
3rd order	$O(n^6)$
4th order	$O(n^8)$

Adaptive Identity Management

- In practice, it is often sufficient to reason over smaller subgroups of people **independently**



Idea: **adaptively factor problem** into subgroups allowing for higher order representations for smaller subgroups

(and Bob was originally in the Blue group)

- 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}

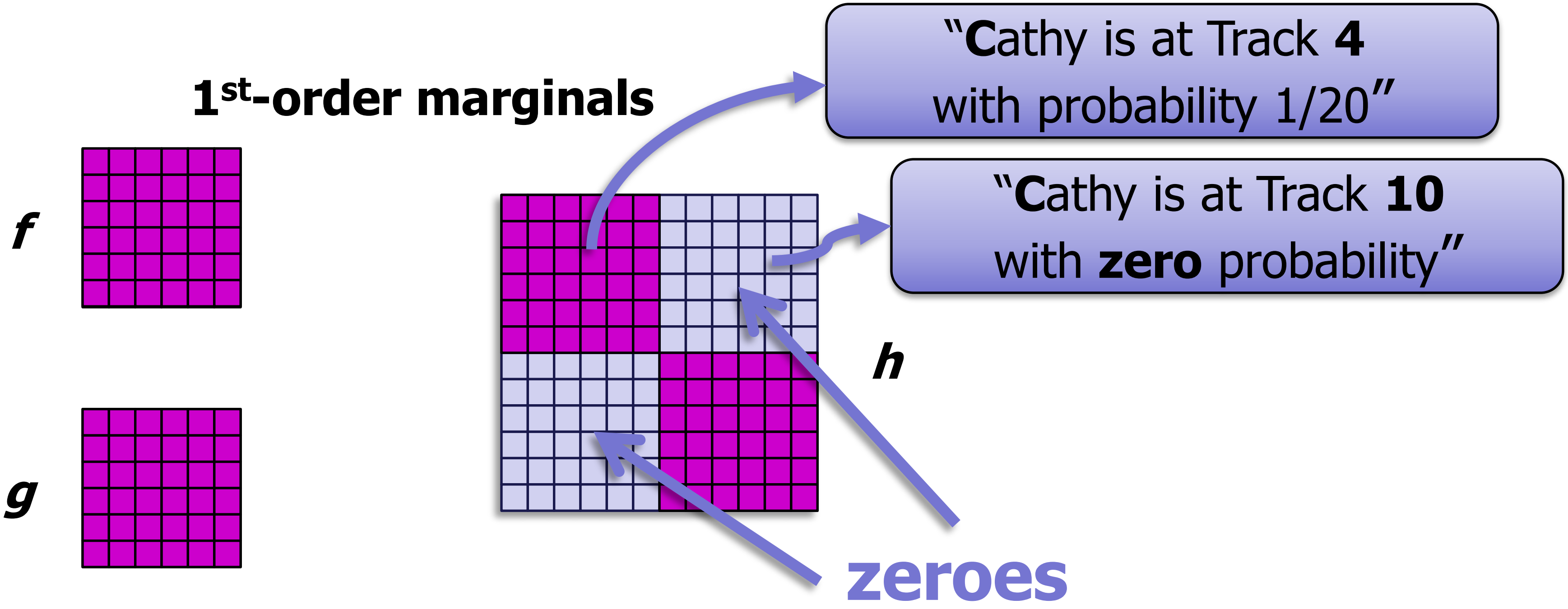
Distribution over tracks {p+1,…,n}

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

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

First-order Independence

- Let f be a distribution on permutations of {1,…,p}, and g be a distribution on permutations of {p+1,…,n}
- **Join problem for 1st-order marginals:**
  - Given 1st-order marginals of *f* and *g*, what does the matrix of 1st-order marginals of *h* look like?



1st-order marginals

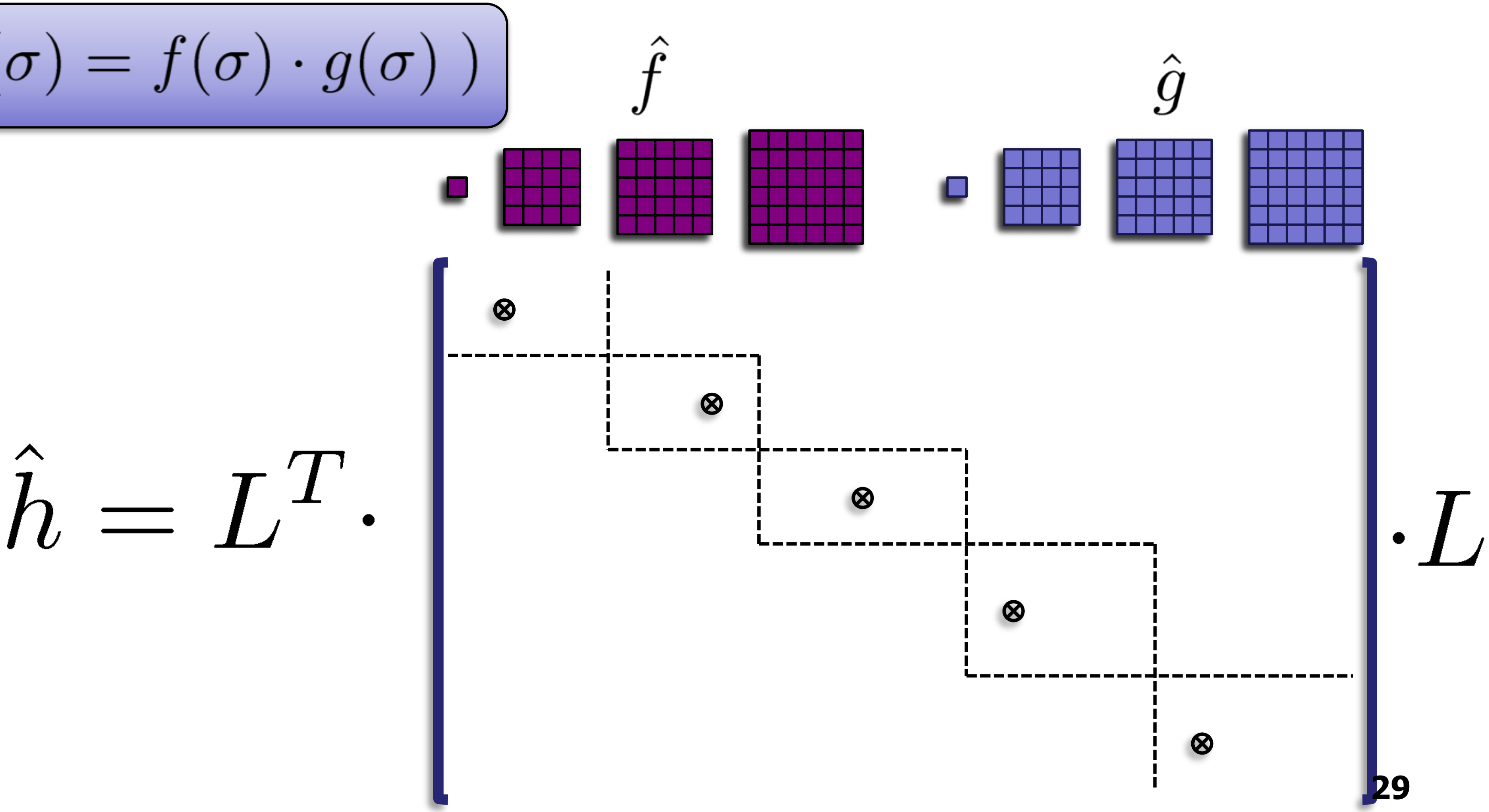
"**C**athy is at Track **4** with probability 1/20"

"**C**athy is at Track **10** with **zero** probability"

Joining

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

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



Joining

- Coefficients of the joint related to coefficients of the factors by:

$$\hat{h}_\lambda = L^T \cdot \bigoplus_{\mu,\nu} \bigoplus_{\ell=1}^{c^\lambda_{\mu,\nu}} \left(\hat{f}_\mu \otimes \hat{g}_\nu\right) \cdot L$$

- **Block multiplicities** equivalent to *Littlewood-Richardson* coefficients
  - #P-hard to compute in general, but (very) tractable for low-order decompositions

- **Complexity**: same as prediction/rollup step for the joint distribution (with known block multiplicities)

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}

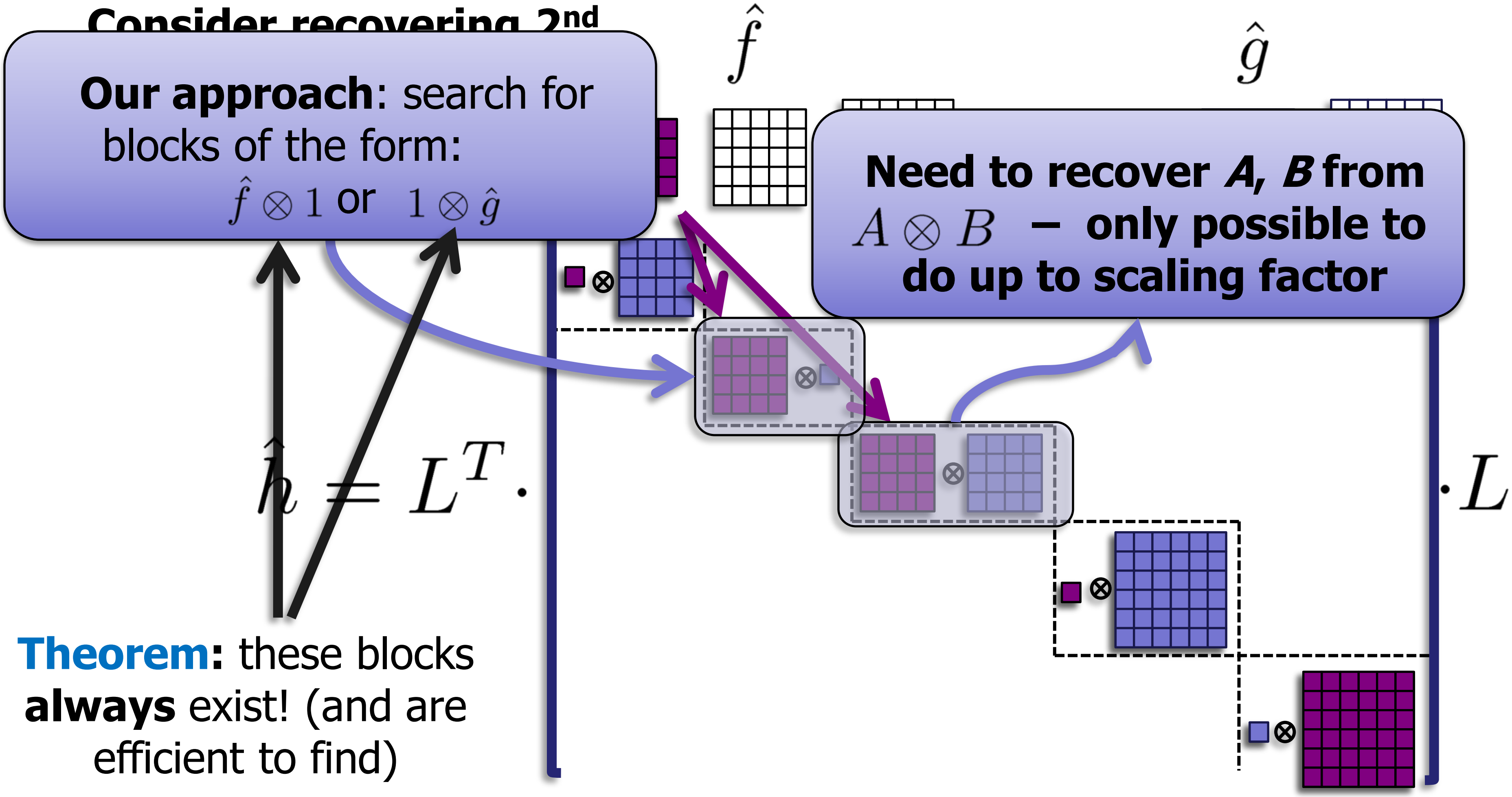
Distribution over tracks {p+1,…,n}

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

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

Splitting

- We would like to "invert" the Join process:



Consider recovering 2nd

Our approach: search for blocks of the form: $\hat{f} \otimes 1$ or $1 \otimes \hat{g}$

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

- Now we know how to **join/split** given the Fourier transform of the input distribution
- **Problem**: In practice, never have entire set of Fourier coefficients!
- **Marginal preservation guarantee**:

Theorem: *Given m*th*-order marginals for independent factors, then we* **exactly** *recover m*th*-order marginals for the joint distribution.*

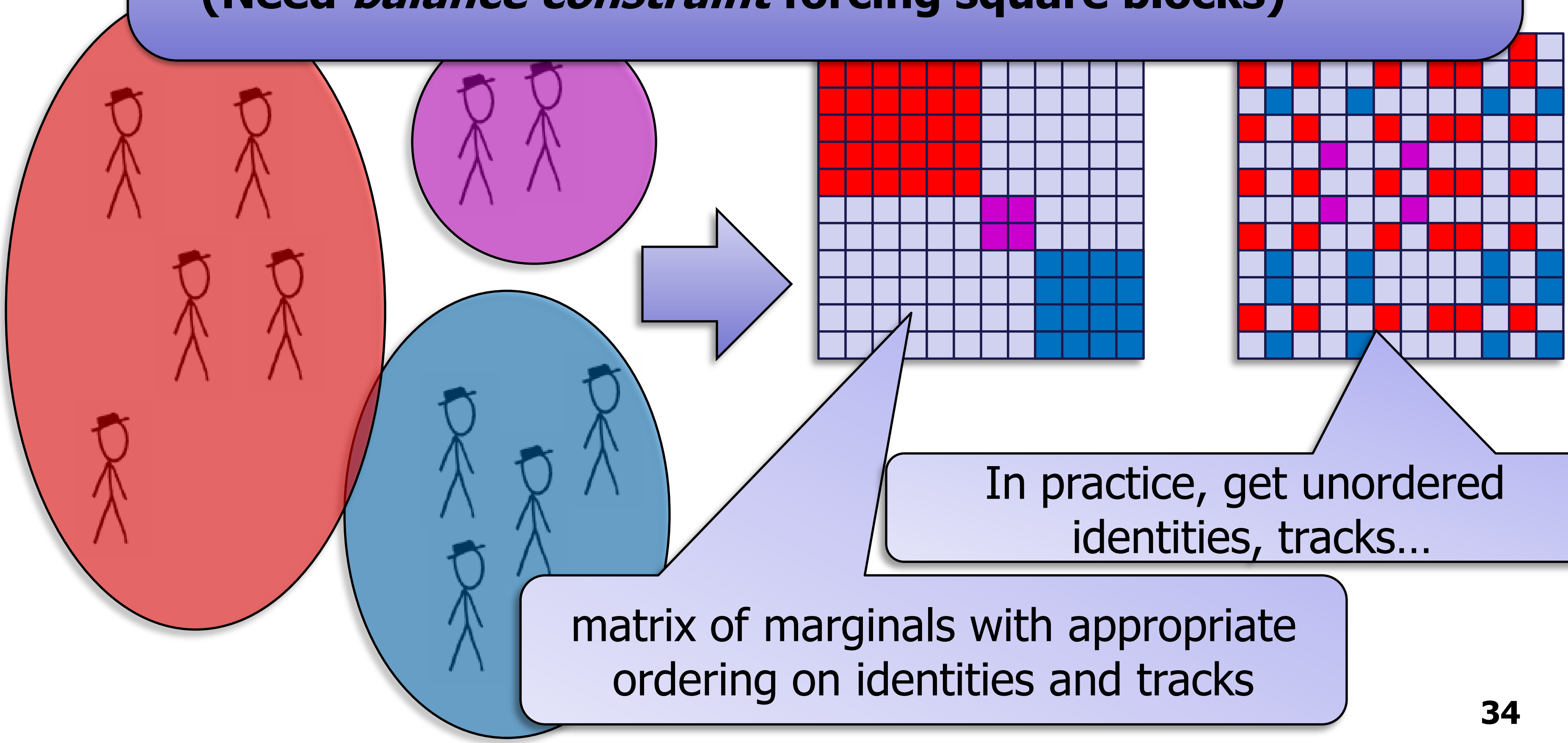
  - Conversely, we get a similar guarantee for splitting
  - (Usually get some higher order information too)

Detecting Independence

- To adaptively split large distributions, need to be able to

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

*** (Need *balance constraint* forcing square blocks)**



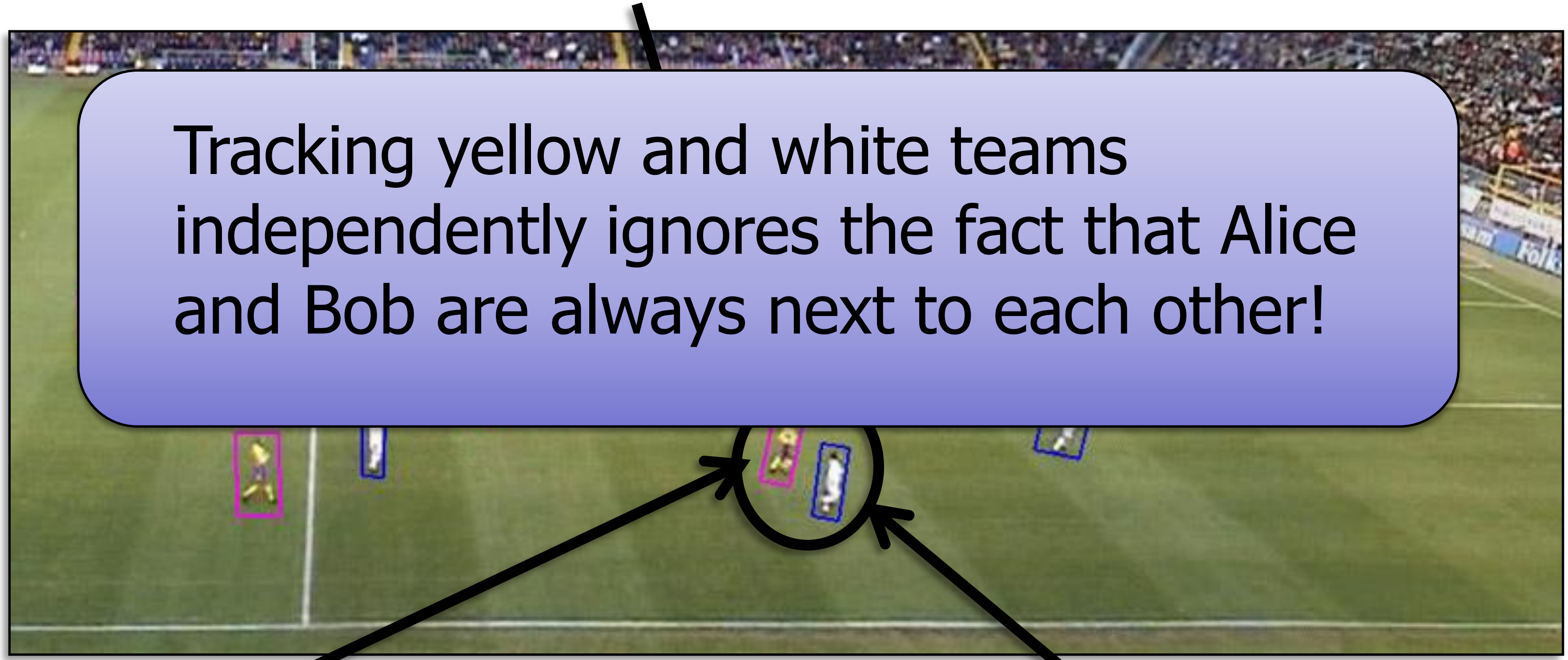
In practice, get unordered identities, tracks…

matrix of marginals with appropriate ordering on identities and tracks

First-order independence

- First-order condition is insufficient:

"Alice guards Bob"



Tracking yellow and white teams independently ignores the fact that Alice and Bob are always next to each other!

"Alice is in yellow team"

"Bob is in white team"

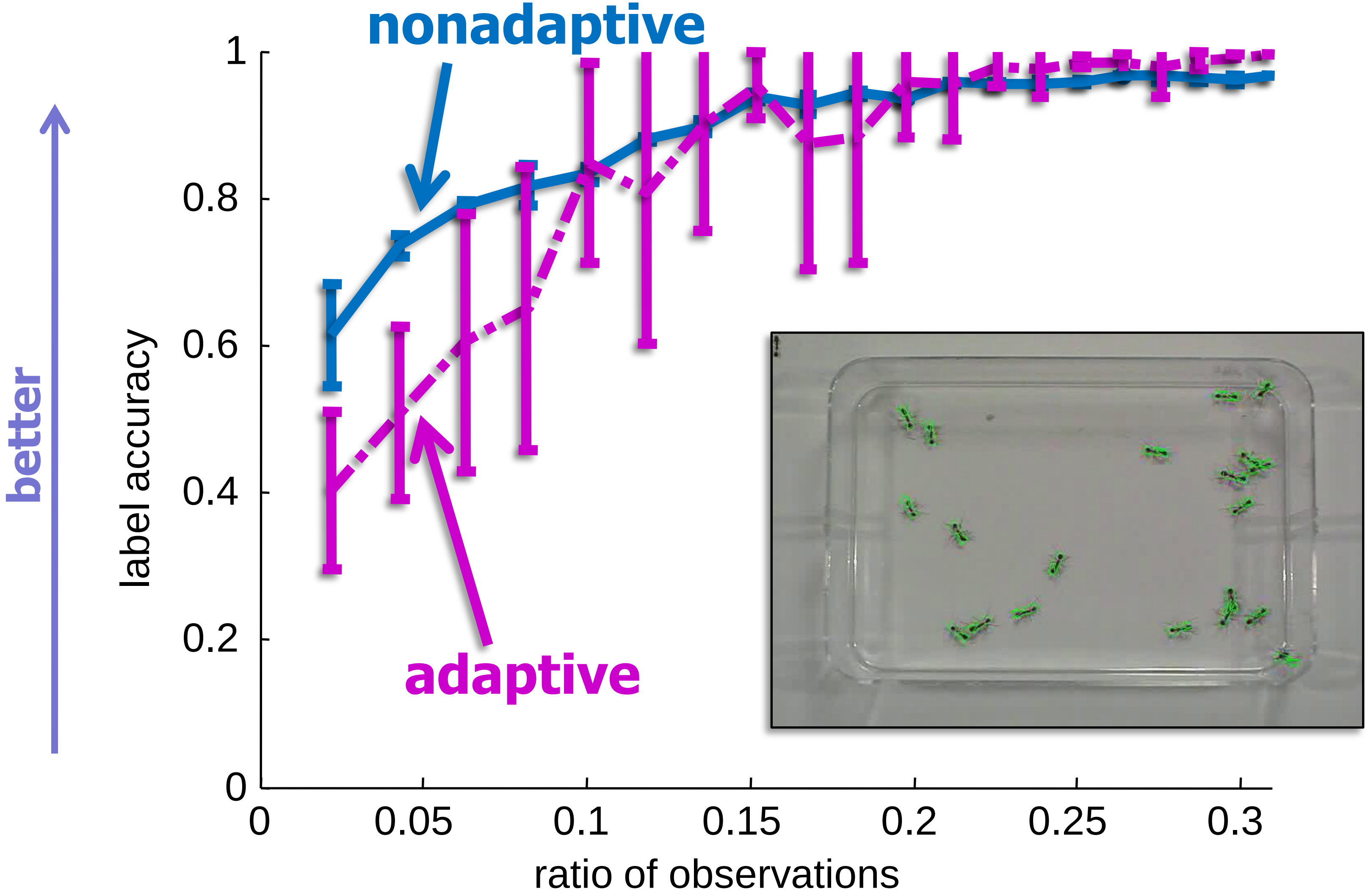
Handling Near-Independence

- We only detect at first-order, but:
  - We can measure departure from independence at higher orders
  - And even when higher order independence does not hold, we have the following result:

Theorem: *If first-order independence holds, we always obtain exact marginals of each subset of tracks.*

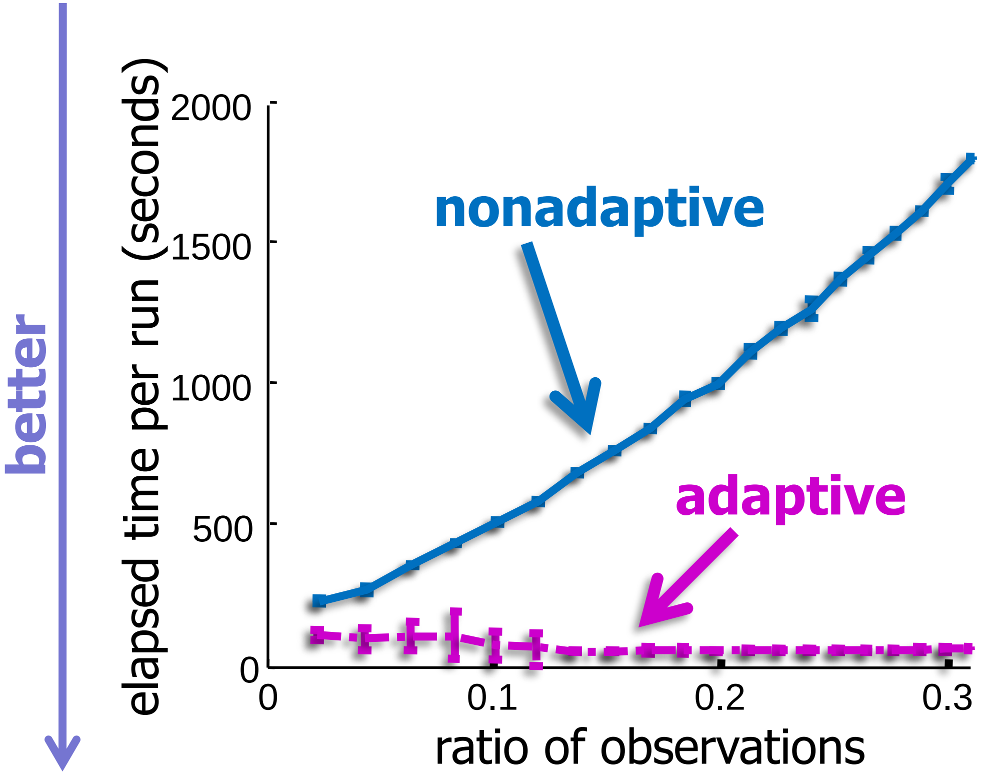
- (we get a marginal distribution for white team and a marginal distribution for yellow team)
- When first-order independence does not hold, we obtain approximate marginals.

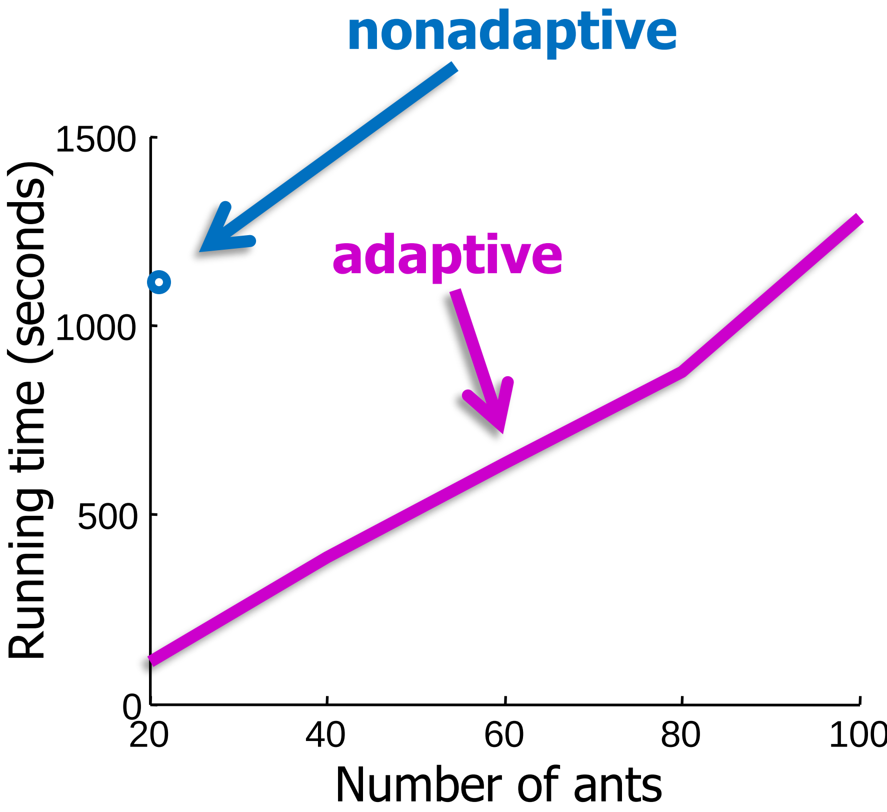
Experiments - Accuracy



dataset from [Khan et al. 2006]

Experiments – Running time





Conclusion

- 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 (*prediction/rollup*, *conditioning*, *join*, *split*)
- Addressed approximation and scalability issues
- Applied algorithms successfully on simulated and real data

- **Opens significant, new research opportunities in AI/ML**
  - **Some ideas generalize to other finite groups**

Thanks Thanks Thakns Thaksn Thasnk Thaskn Thnask Thnksa …

Jonathan Huang

NIPS Algebraic Methods Workshop (AML '08)
12/11/08

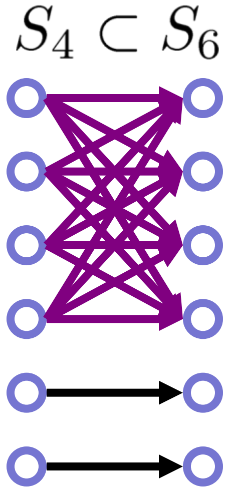
Algorithm Summary

- **Initialize prior** Fourier coefficient matrices $\hat{P}^{(0)}$
- For each timestep t = 1,2,…,T
  - **Prediction/Rollup:**
    - For all coefficient matrices $\hat{P}_i^{(t)}$
      - $\hat{P}_i^{(t)} \leftarrow \hat{Q}_i^{(t)} \cdot \hat{P}_i^{(t-1)}$
  - **Conditioning**
    - For all pairs of coefficient matrices $(\hat{P}_i^{(t)}, \hat{L}_j^{(t)})$
      - Compute $\hat{P}_i^{(t)} \otimes \hat{L}_j^{(t)}$ and reproject to the orthogonal Fourier basis
  - **Drop high frequency coefficients** of $\hat{P}^{(t)}$
  - **Project** $\hat{P}^{(t)}$ to relaxed Marginal polytope using a Quadratic program
- Return marginal probabilities for all timesteps

Input: **Fourier coefficients** of **mixing** and **observation** models

Mixing and Observation Models

- Fourier-theoretic framework can handle a variety of probabilistic models
- **But…** need to be able to ***efficiently* compute Fourier coefficients** for mixing/observation models…

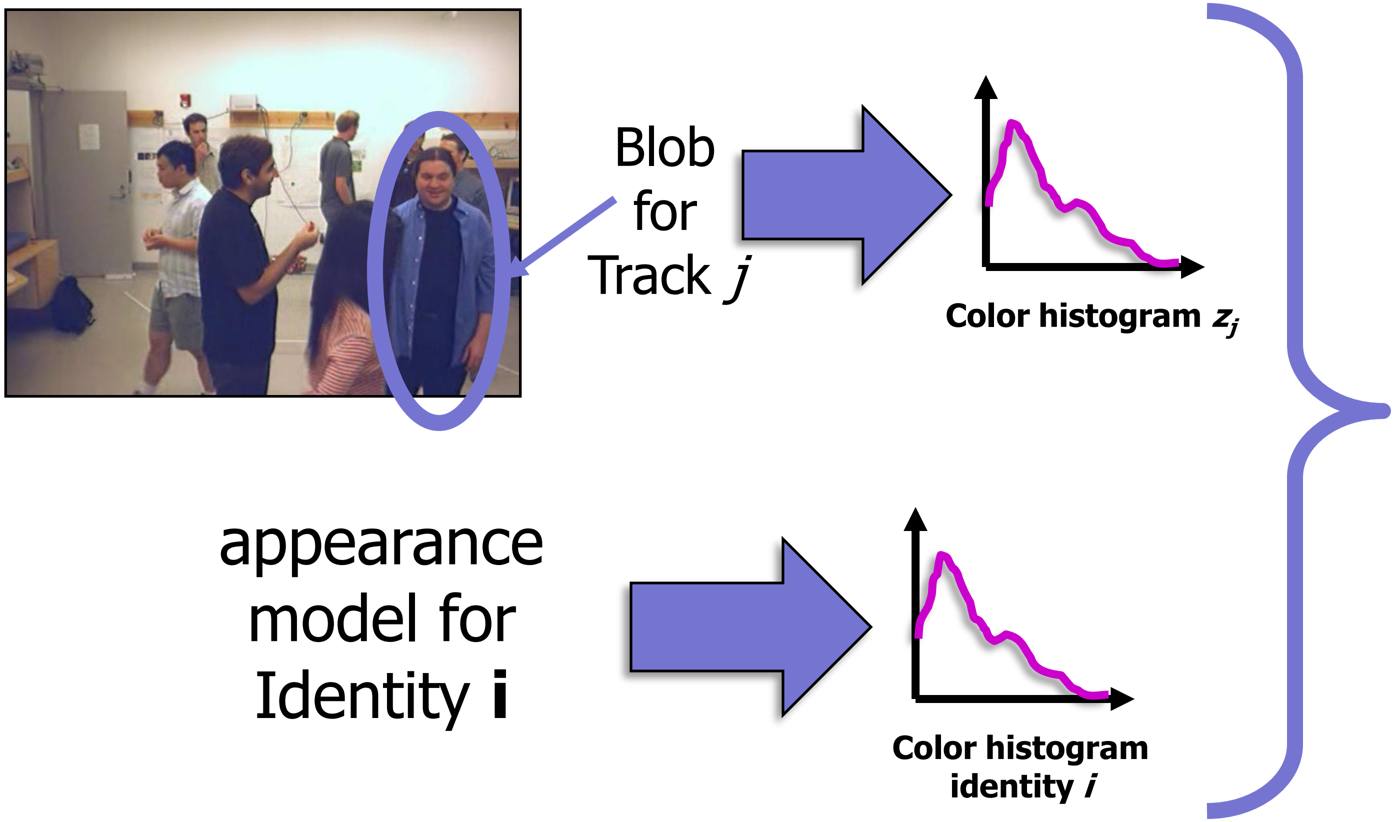

$S_4 \subset S_6$

- Useful family of function "primitives":
  - Can **efficiently** Fourier transform the indicator function of subgroups of the form $S_k \subset S_n$:

$$\delta_{S_k}(\sigma) = \begin{cases} 1 & \text{if } \sigma(i) = i \text{ for all } k < i \leq n \\ 0 & \text{otherwise} \end{cases}.$$

  - Fourier coefficient matrices of S_k-indicators are **diagonal**, with all nonzero entries equal to k!

1st order observation model for tracking



Blob for Track *j*

Color histogram z_j

appearance model for Identity **i**

Color histogram identity *i*

If Identity **i** is on Track **j**, prob. z_j is Gaussian with mean = appearance

If we make one such observation per track, P(z|σ) is proportional to $\delta_{S_{n-1}}$ and can be represented **exactly by 1st-order Fourier parameters**

S_k-Indicator Primitives

- Most mixing/observation models can be written as (sparse) I s!

Associated **subgroup** of the S_n

Indicator function of of $S_k x S_{n-k}$ is the **convolution** of indicators of S_k and S_{n-k}

P		identities”
k-		2,3,4,5}”

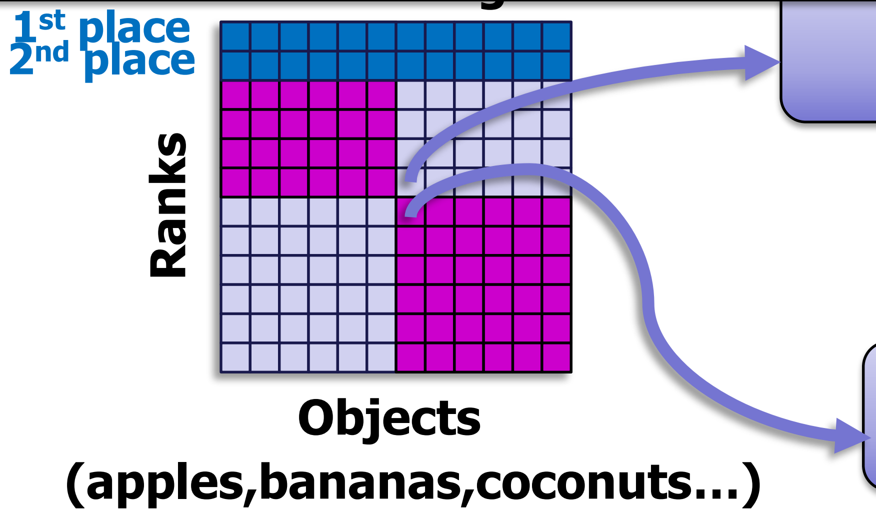
Observation Models		
Singletrack	S_{n-1}	“Alice is at track 2”
Multitrack	S_{n-k}	“Alice is at track 2, Bob is at track 3”
Bluetooth	$S_k x S_{n-k}$	“Red team is at tracks {1,3,5,6,8,9}”
Pairwise ranking	S_{n-2}	“Apples are better than oranges”

Generalized Independence

- **Observation:** We care:
  - **more** about interactions between first and second place, and
  - **less** about interactions between first and last place.

- Independence allows us to capture something like this,
  b

Generalized Independence: Can we exploit some kind of alternative structure?



"Guava is ranked 5th with **zero** probability"

"Guava is ranked 6th with some probability"

Rank Independence

- Candidate idea: Instead of factoring into independent distributions over ranks, factor into distributions over **relative ranks**
- Example:
  - σ = [7 3 2 6 5 4 8 1 9]
  - ¿ = [1 2 3 4]
  - Relative ranking of ¿ in σ:

$$RR_\sigma(\tau) = [4\ 2\ 1\ 3]$$

- Definition:

Define (1,…,p) and (p+1,…n) to be *rank independent* if:

$$h(\sigma) = f(RR_\sigma([1,\ldots,p])) \cdot g(RR_\sigma([p+1,\ldots,n]))$$

Rank Independence

$$h(\sigma) = f(RR_\sigma([1,\ldots,p])) \cdot g(RR_\sigma([p+1,\ldots,n]))$$

- R
  i
- T
  c

$h($ $]))$

Rank independence:

Does rank independence hold in real ranked data?

Can we exploit it for fast inference?

Are there conditional generalizations of rank independence?

(think of shuffling two independent permutations together)

- Some connections to *Radon transforms*…

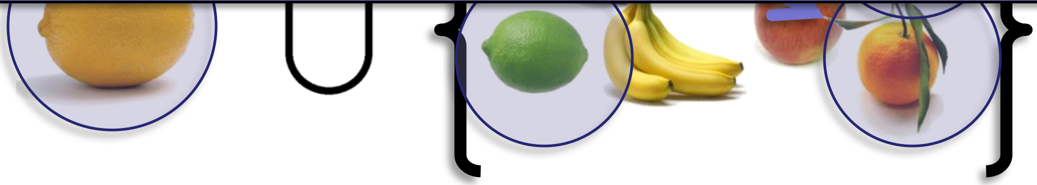
Generalization to unseen objects

- Consider a distribution P over user preference rankings on fruits:

Generalization to unseen objects:

Allow for objects to be associated with side information (features)

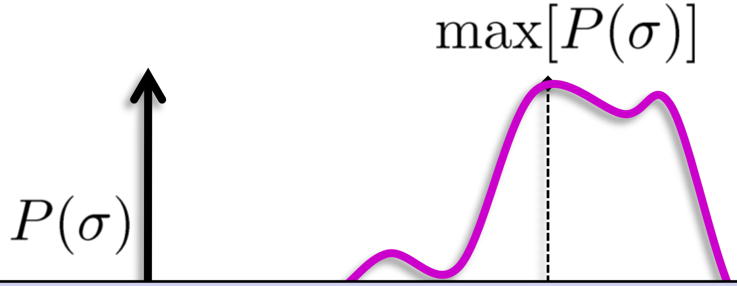
Allow for observation models to depend on features



- What if we know that the new object is a citrus fruit?

Optimization

- Is MAP inference easier given a bandlimited function?



Optimization: Can we formulate Fourier domain optimization algorithms that work well in practice?

- O
- Optimizing a **1st-order** function is reduces to bipartite matching and can be done in polynomial time…
- Unfortunately:

Theorem: *Any instance of the traveling salesman problem can be reduced to optimizing a second-order function on permutations in polynomial time.*