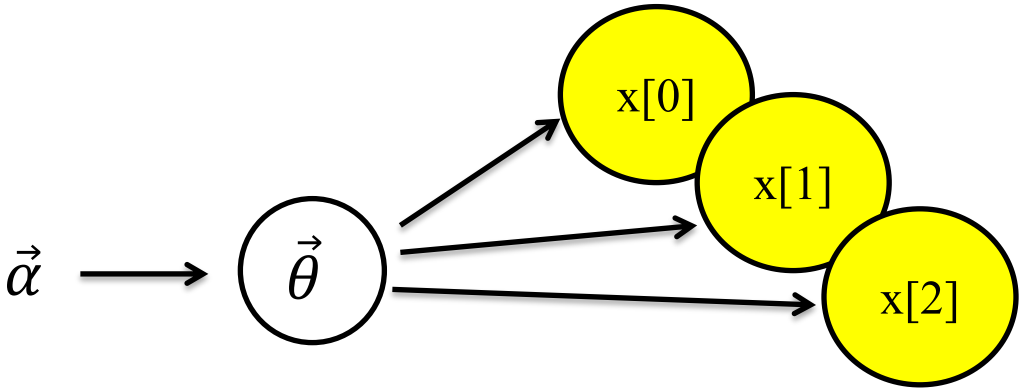
Query-Driven Probabilistic Programming

Niccolò Meneghetti
University of Michigan-Dearborn
niccolom@umich.edu

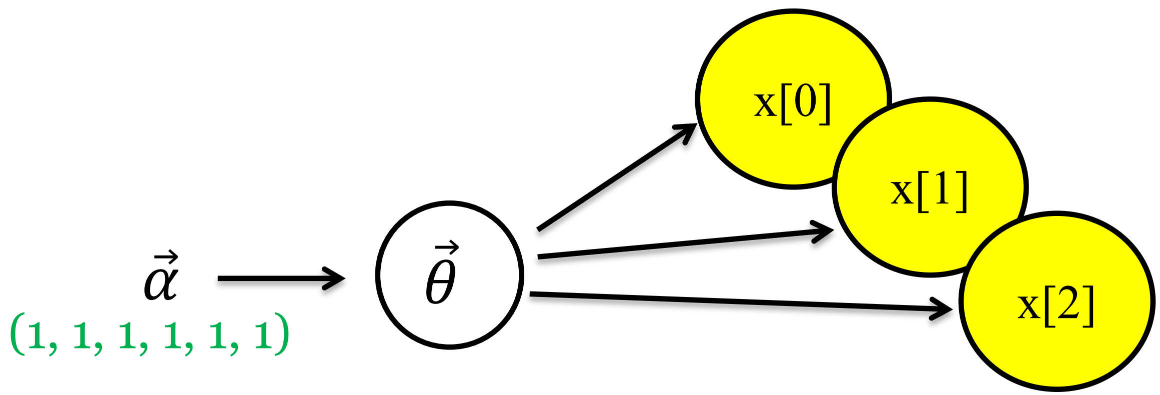
Bayesian Dice

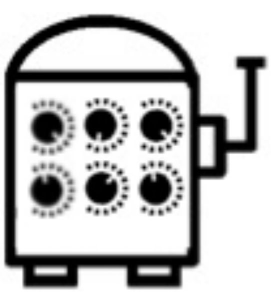
`Bayesian die` → a *Categorical* distribution (parametrized by $\vec{\theta}$) with a *Dirichlet* prior (parametrized by $\vec{\alpha}$)



Bayesian Dice

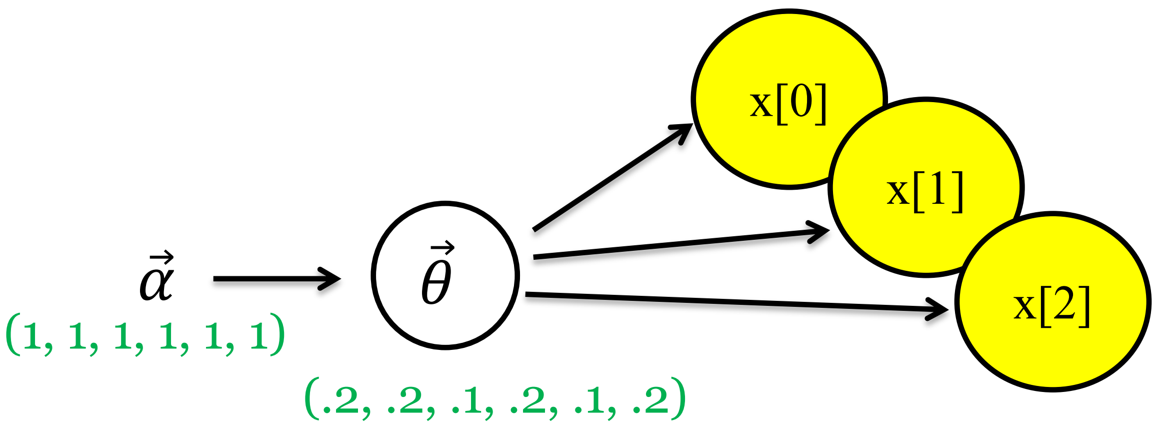
`Bayesian die` → a *Categorical* distribution (parametrized by $\vec{\theta}$) with a *Dirichlet* prior (parametrized by $\vec{\alpha}$)

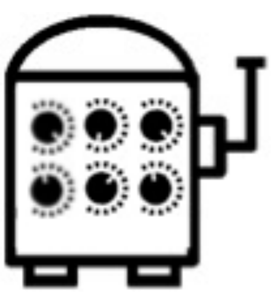


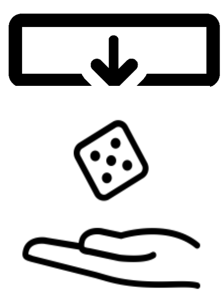


Bayesian Dice

`Bayesian die` → a *Categorical* distribution (parametrized by $\vec{\theta}$)
with a *Dirichlet* prior (parametrized by $\vec{\alpha}$)

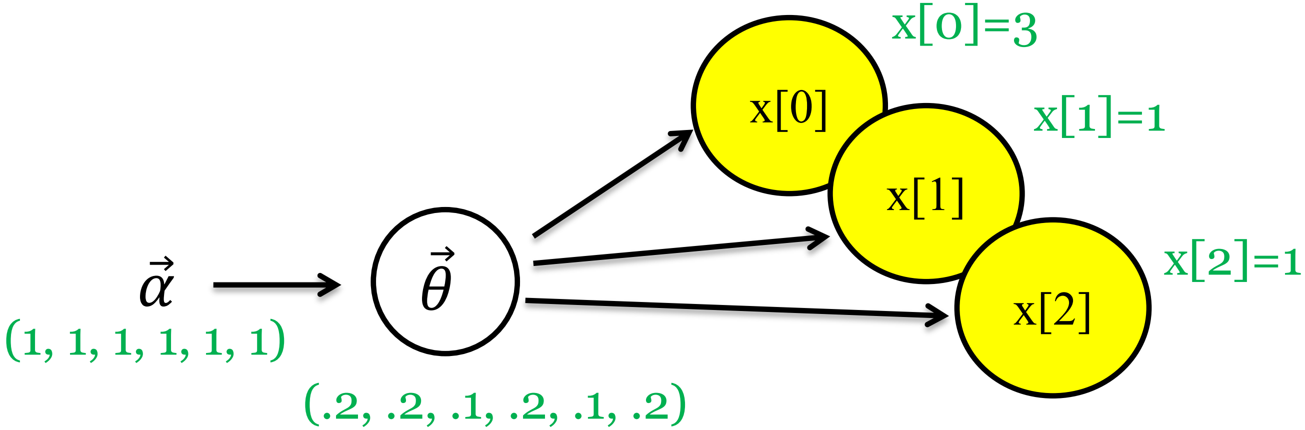


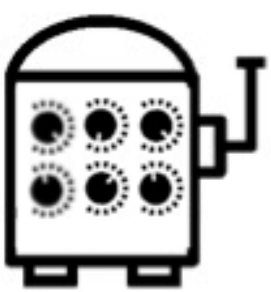


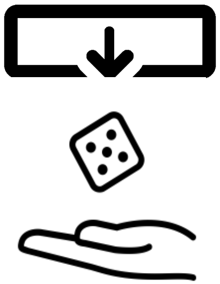


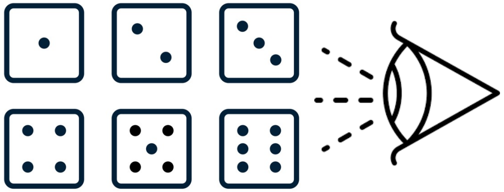
Bayesian Dice

`Bayesian die` → a *Categorical* distribution (parametrized by $\vec{\theta}$) with a *Dirichlet* prior (parametrized by $\vec{\alpha}$)



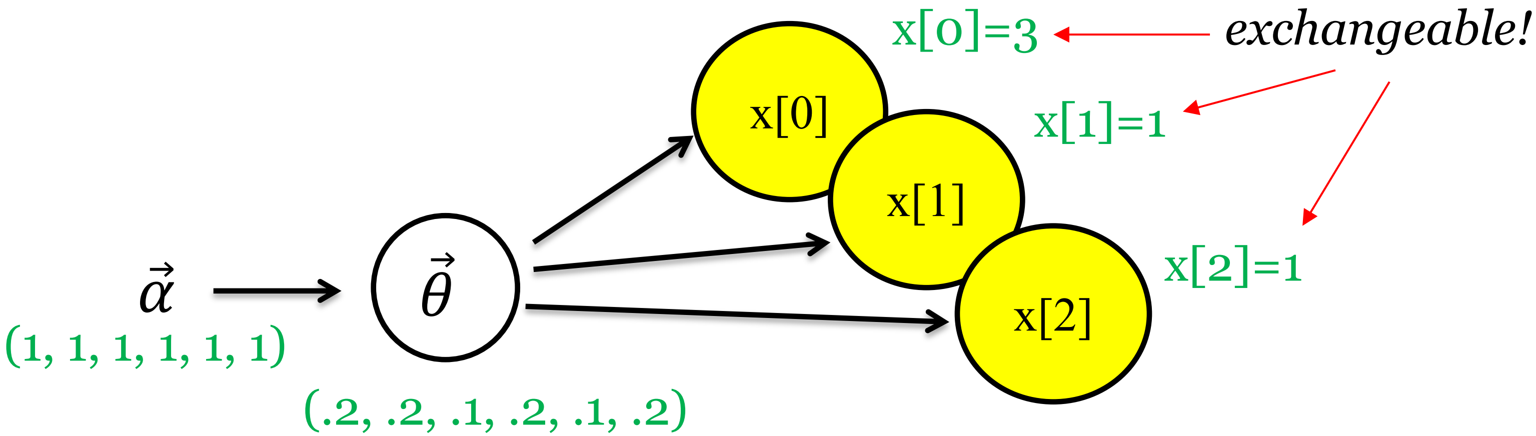


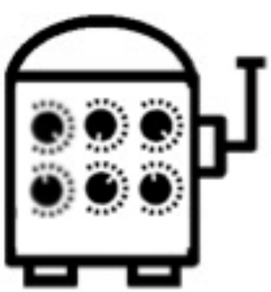


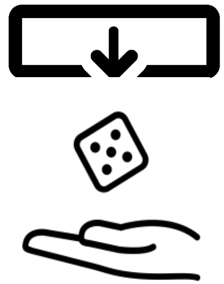


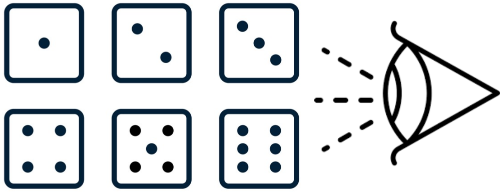
Bayesian Dice

`Bayesian die` → a *Categorical* distribution (parametrized by $\vec{\theta}$) with a *Dirichlet* prior (parametrized by $\vec{\alpha}$)



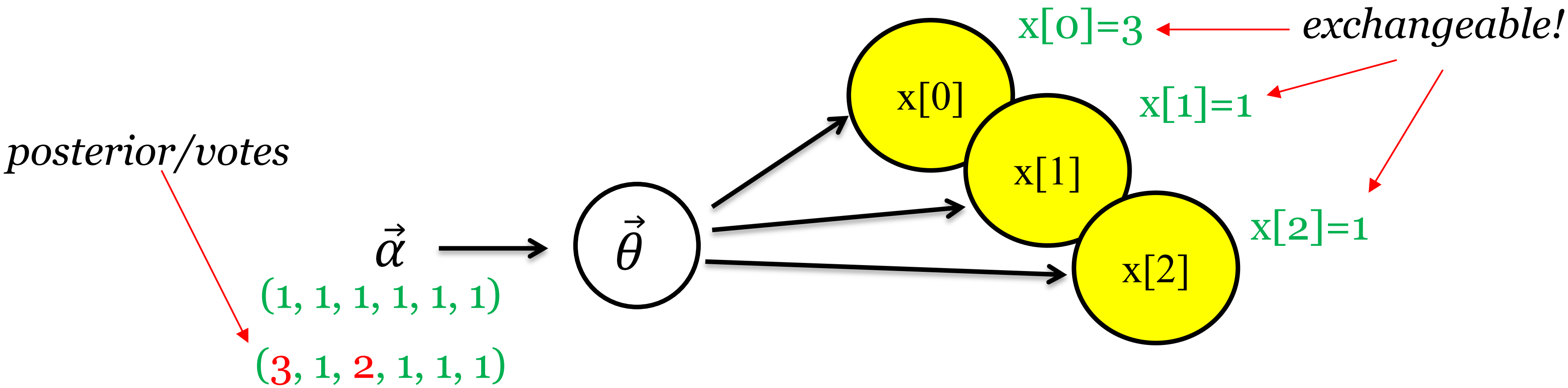


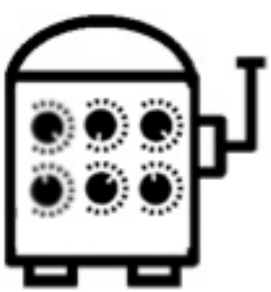


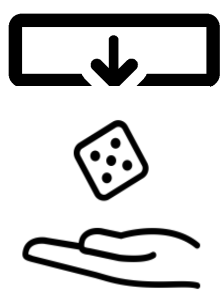


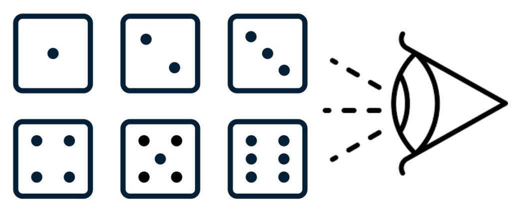
Bayesian Dice

`Bayesian die` → a *Categorical* distribution (parametrized by $\vec{\theta}$)
with a *Dirichlet* prior (parametrized by $\vec{\alpha}$)







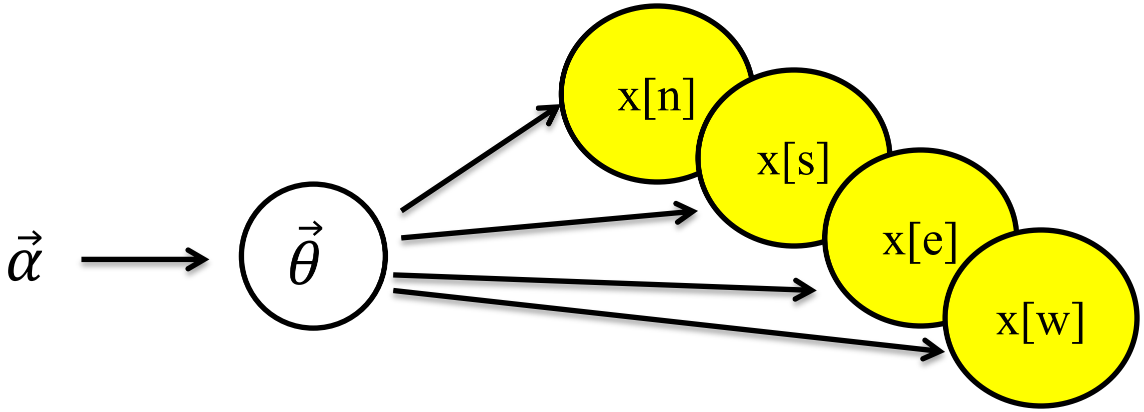


Bayesian Dice in Action: Ising Model

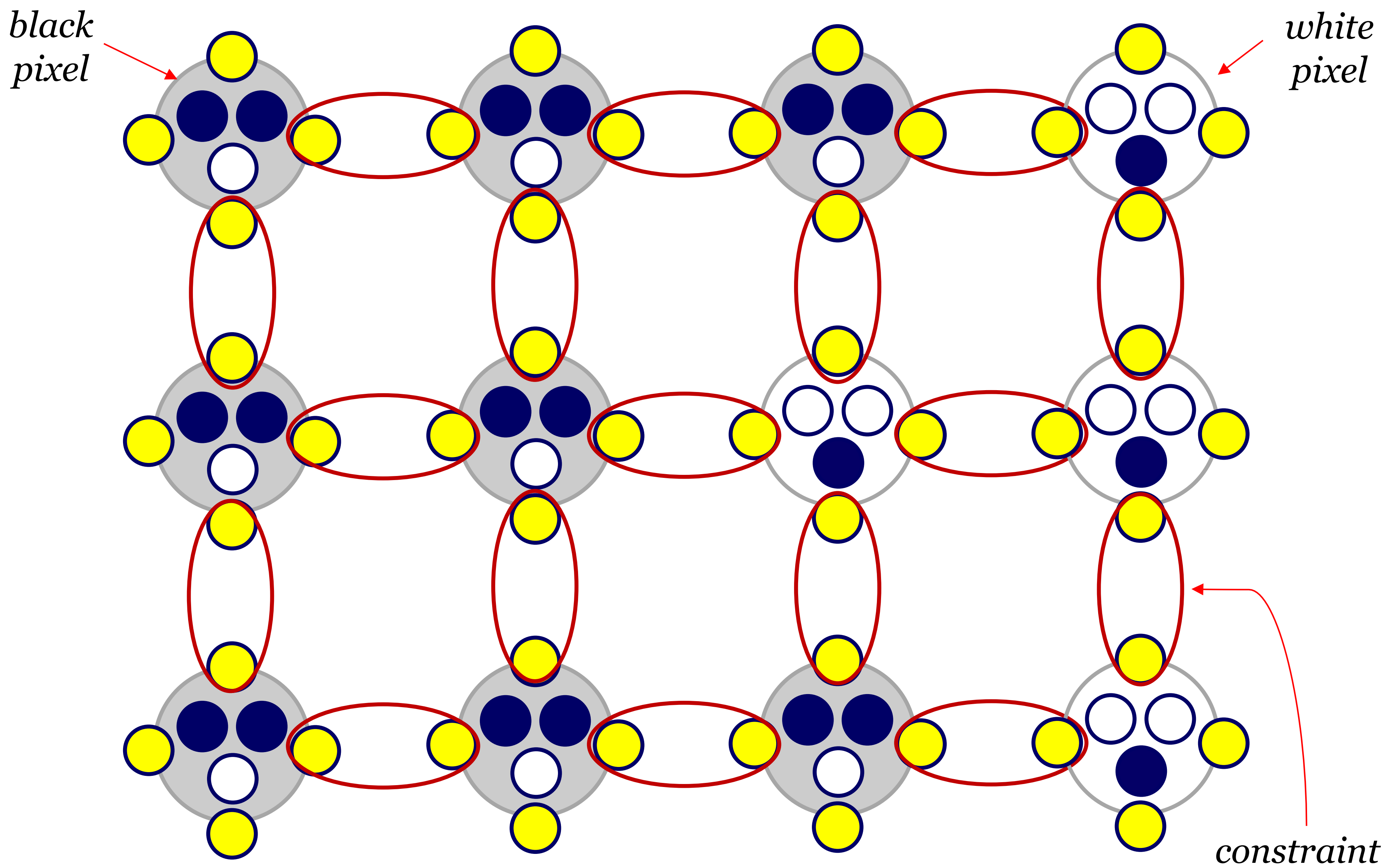
Setup:

- One Bayesian "coin" (d=2) per each pixel
- If the pixel is ON: $\vec{\alpha}$:= (2,1)
- If the pixel is OFF: $\vec{\alpha}$:= (1,2)

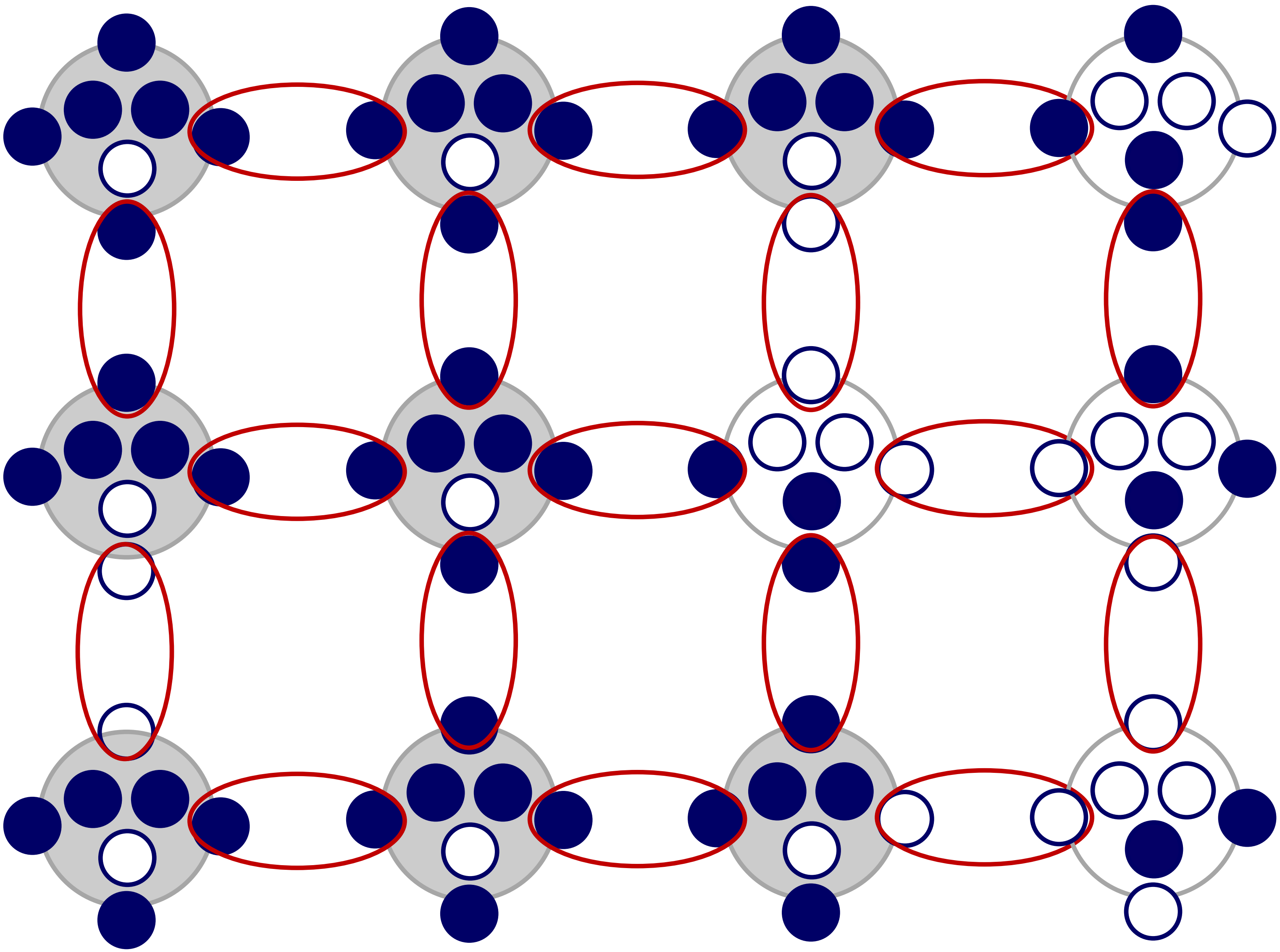
Constraint: "we make two new observations for every pair of neighboring pixels, and we assert they share the same configuration (either they are both ON or both OFF)"



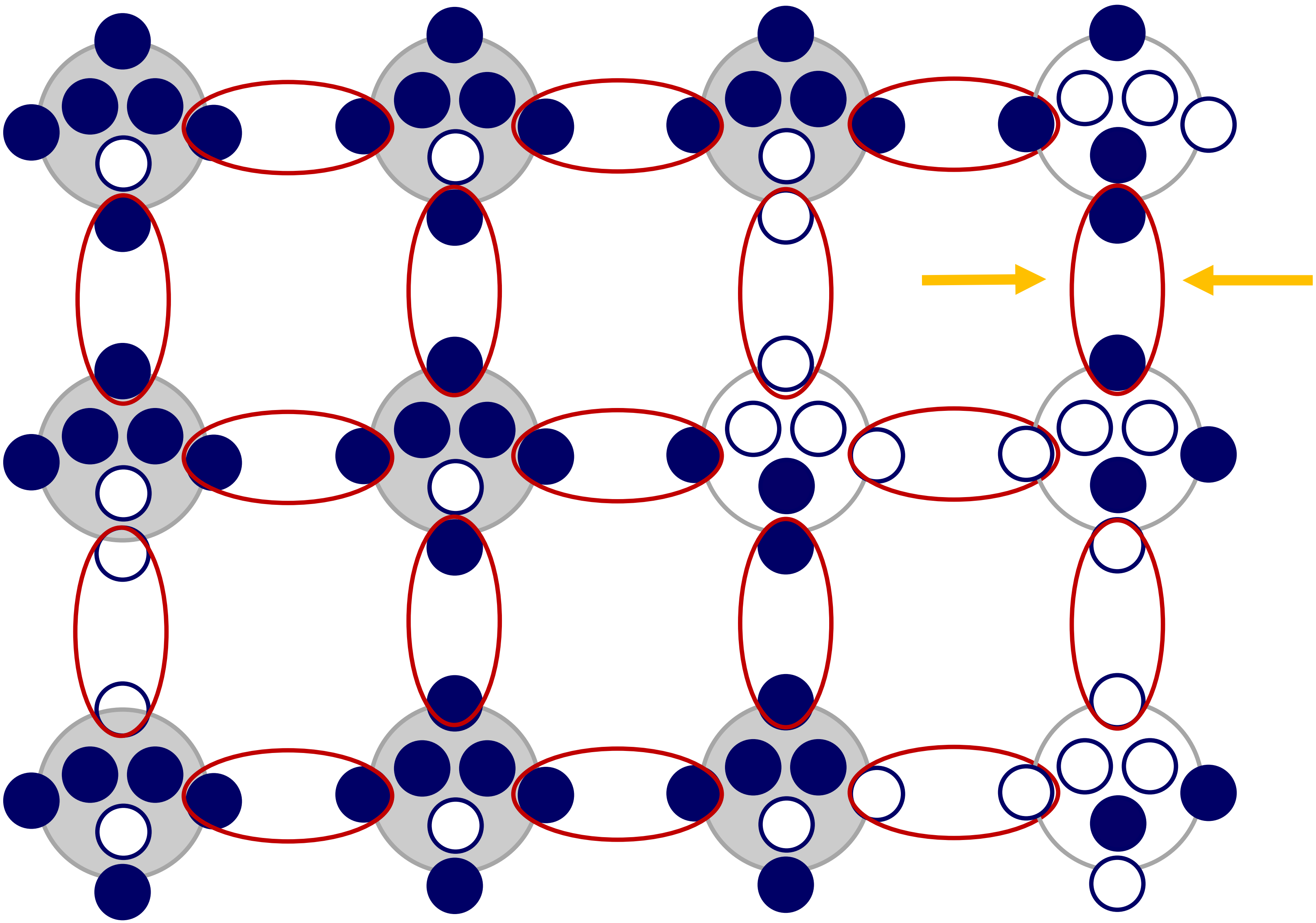
Ising Model (simplified)



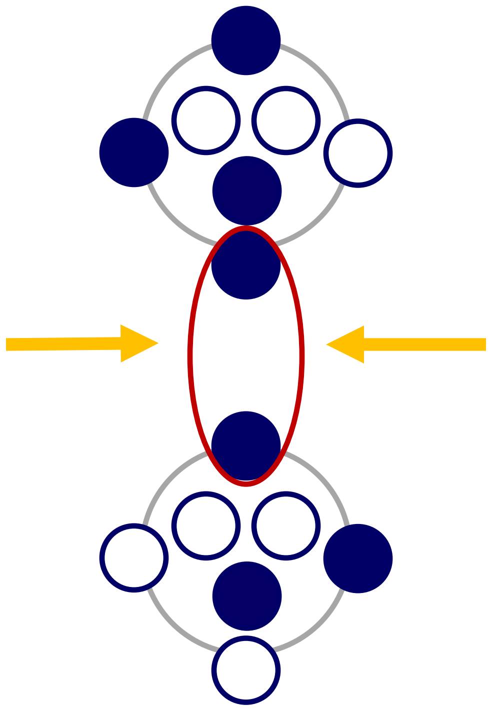
Ising Model (simplified)



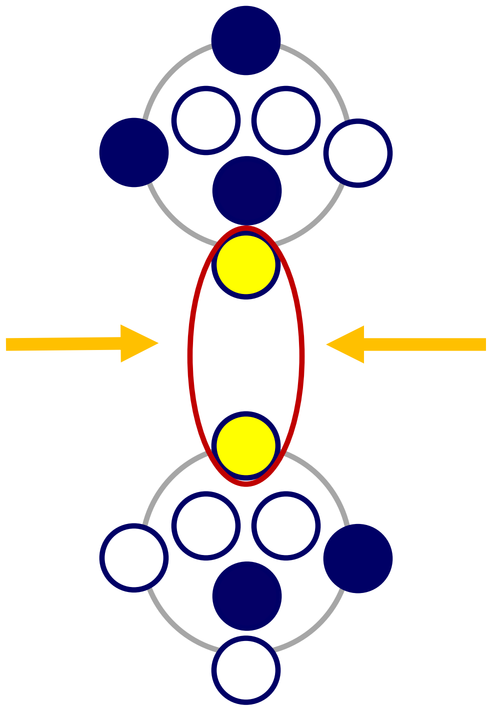
Ising Model (simplified)



Ising Model (simplified)

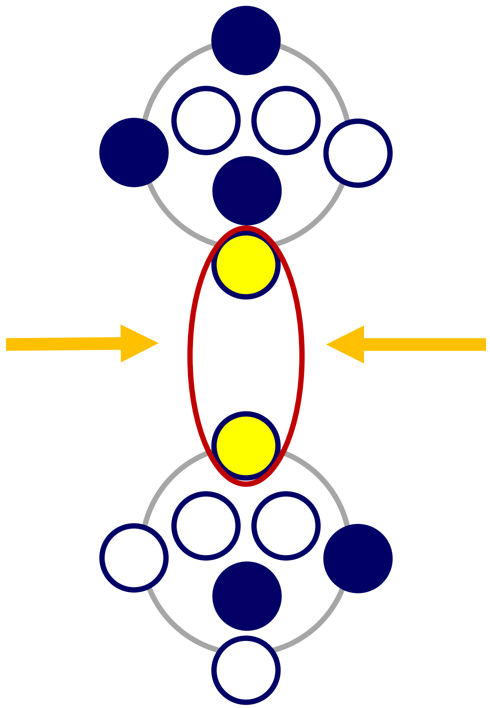


Ising Model (simplified)



Ising Model (simplified)

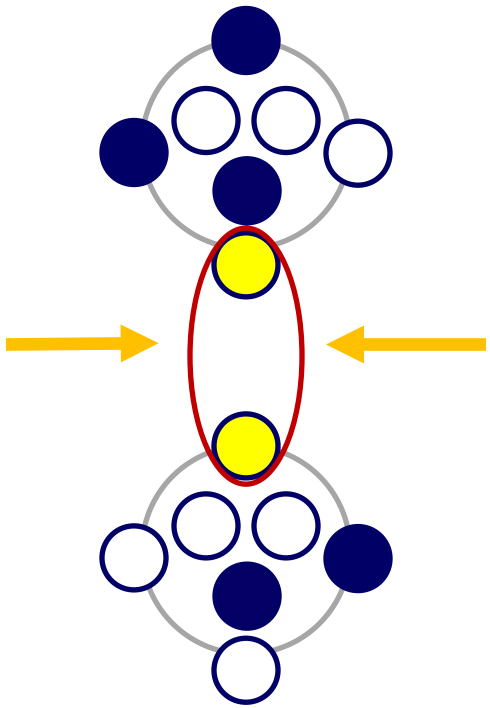
conf	***probability***
○ ○	3/6 * 4/6 = **1/3**
● ○	3/6 * 4/6 = **1/3**
○ ●	3/6 * 2/6 = **1/6**
● ●	3/6 * 2/6 = **1/6**



Ising Model (simplified)

conf	*probability*
○○	3/6 * 4/6 = **1/3**
●○	3/6 * 4/6 = **1/3**
○●	3/6 * 2/6 = **1/6**
●●	3/6 * 2/6 = **1/6**

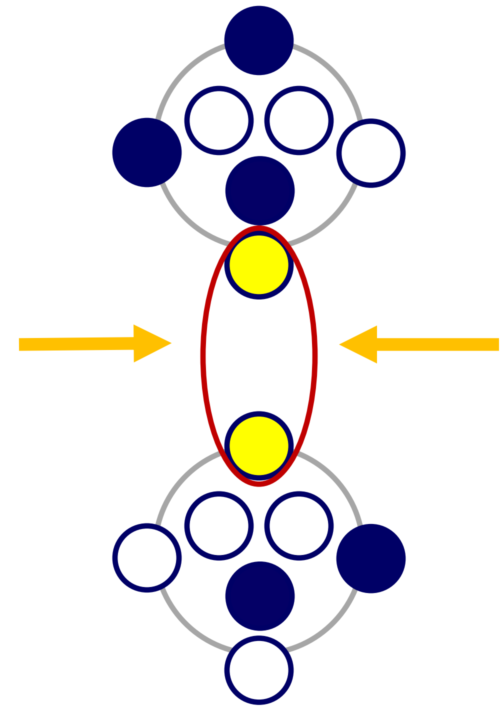
QA	*probability*
○○	(1/3)/(1/2)= **2/3**
●●	(1/6)/(1/2)= **1/3**



Ising Model (simplified)

conf	*probability*
○○	3/6 * 4/6 = **1/3**
●○	3/6 * 4/6 = **1/3**
○●	3/6 * 2/6 = **1/6**
●●	3/6 * 2/6 = **1/6**

QA	*probability*
○○	(1/3)/(1/2)= **2/3**
●●	(1/6)/(1/2)= **1/3**



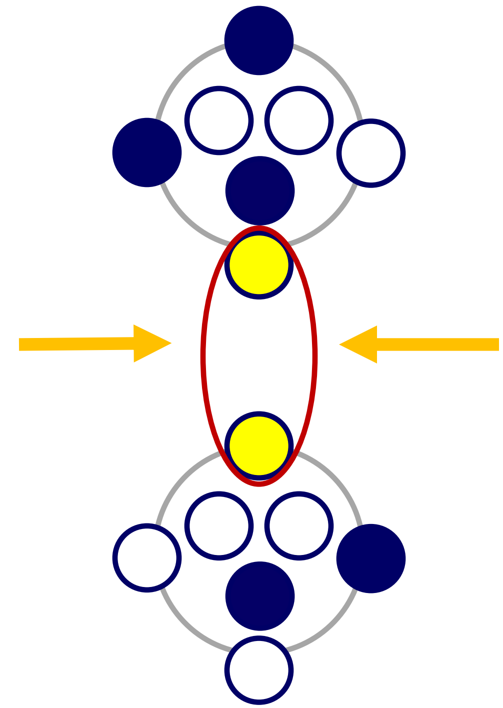
With probability **2/3** we choose ○○

With probability **1/3** we choose ●●

Ising Model (simplified)

conf	*probability*
○○	3/6 * 4/6 = **1/3**
●○	3/6 * 4/6 = **1/3**
○●	3/6 * 2/6 = **1/6**
●●	3/6 * 2/6 = **1/6**

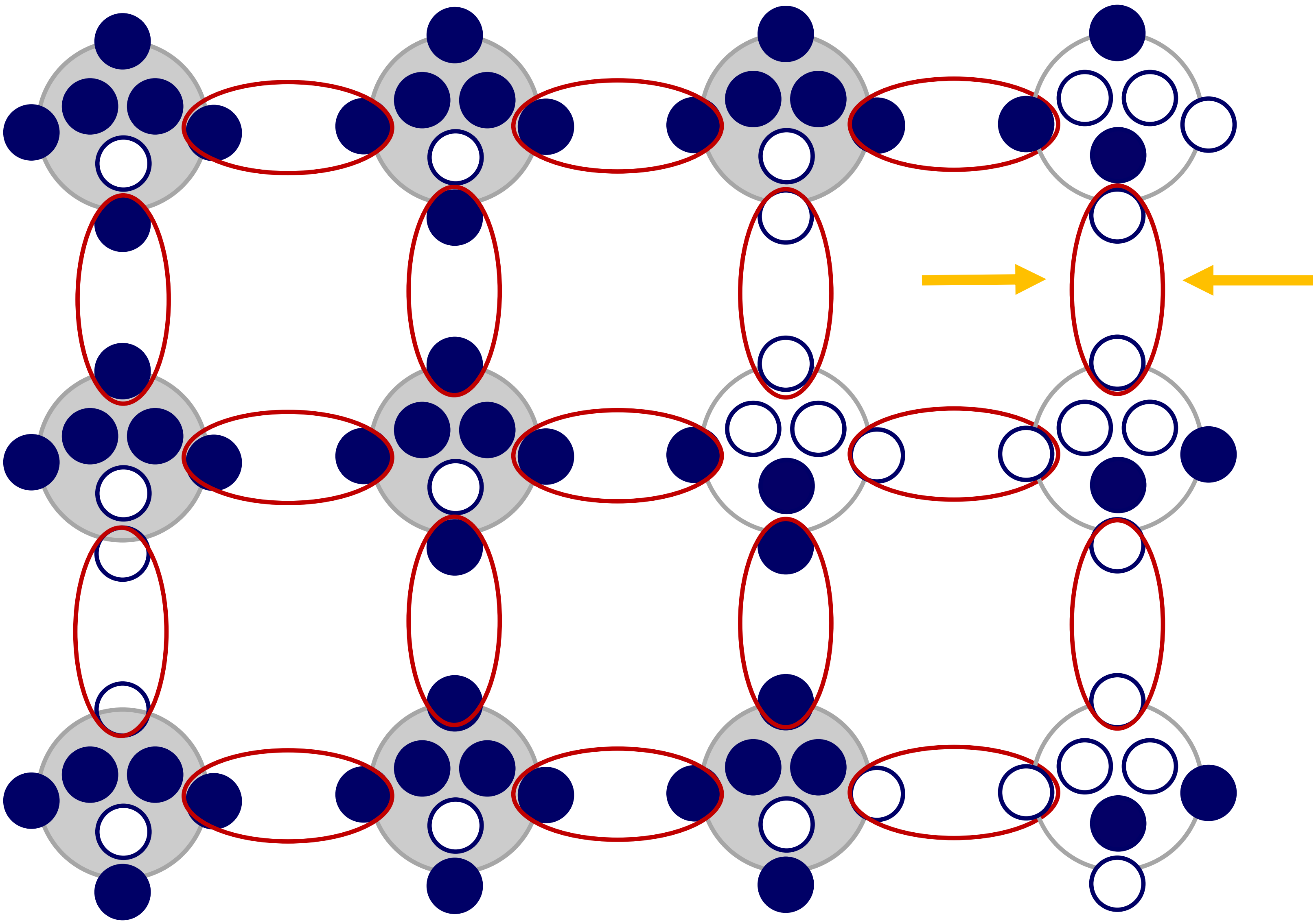
QA	*probability*
○○	(1/3)/(1/2)= **2/3**
●●	(1/6)/(1/2)= **1/3**



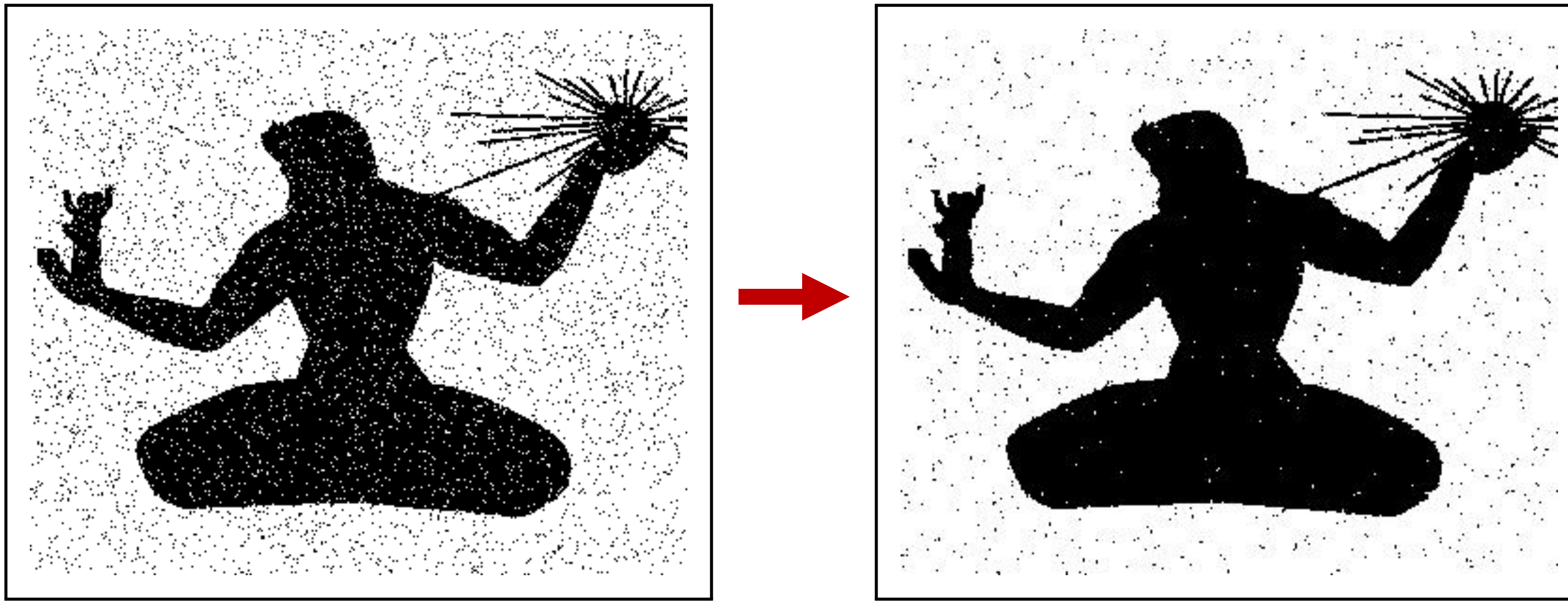
With probability **2/3** we choose ○○

With probability **1/3** we choose ●●

Ising Model (simplified)



Ising Model (simplified)



Niccolò Meneghetti, Ouael Ben Amara
Learning From Exchangeable Query-Answers
EDBT 2022

From Bayesian Dice to Probabilistic DBs

(probabilistic program)

$$R_{ES} := I \,::\!\bowtie\, \sigma_{dir=E|S}(S)$$

$$R_{NW} := I \,::\!\bowtie\, \sigma_{dir=N|W}(S)$$

$$(R_{ES} \bowtie_c R_{NW}) \subseteq \emptyset$$

$c := (R_{ES}.x+1= R_{NW}.x \wedge R_{ES}.y= R_{NW}.y$
$\wedge R_{ES}.dir=E \wedge R_{NW}.dir=W)$
or $(R_{ES}.y+1=R_{NW}.y \wedge R_{ES}.x= R_{NW}.x$
$\wedge R_{ES}.dir=S \wedge R_{NW}.dir=N))$
and $(R_{ES}.state \neq R_{NW}.state)$

IMAGE (I)

	$\vec{\alpha}$	X	Y	STATE
$x_{0,0}$	2,1	0	0	ON(v_0)
				OFF(v_1)
$x_{0,1}$	1,2	0	1	ON(v_0)
				OFF(v_1)
$x_{\dots,\dots}$	…	…	…	…
				…

SAMPLES (S)

X	Y	DIR
0	0	N
0	0	S
0	0	W
0	0	E
0	1	N
0	1	S
0	1	W
0	1	E
…	…	…

***In a nutshell**: it's just like Pyro, PyMC or Stan,… but with databases!*

From Bayesian Dice to Probabilistic DBs

(probabilistic program)

$$R_{ES} := I :: \bowtie \sigma_{dir=E|S}(S)$$

$$R_{NW} := I :: \bowtie \sigma_{dir=N|W}(S)$$

$$(R_{ES} \bowtie_c R_{NW}) \subseteq \emptyset$$

APACHE ARROW Acero

(ground constraints)

$(x_{0,0}[s]=v_0 \wedge x_{0,1}[n]=v_0) \vee (x_{0,0}[s]=v_1 \wedge x_{0,1}[n]=v_1)$

$(x_{0,0}[e]=v_0 \wedge x_{1,0}[w]=v_0) \vee (x_{0,0}[e]=v_1 \wedge x_{0,1}[w]=v_1)$

$(x_{1,0}[s]=v_0 \wedge x_{1,1}[n]=v_0) \vee (x_{1,0}[s]=v_1 \wedge x_{1,1}[n]=v_1)$

$(x_{0,1}[e]=v_0 \wedge x_{1,1}[w]=v_0) \vee (x_{0,1}[e]=v_1 \wedge x_{1,1}[w]=v_1)$

$(x_{2,0}[s]=v_0 \wedge x_{2,1}[n]=v_0) \vee (x_{2,0}[s]=v_1 \wedge x_{2,1}[n]=v_1)$

...

...

(many)

LLVM/ClangJIT

Example: Topics Modeling (Latent Dirichlet Allocation)

N documents → **N** red dice

K topics → **K** blue dice

Red die: generate numbers between 1 and K

Blue die: generate words from a fixed vocabulary

For each word in our corpus:

1) We throw the red die corresponding to the current document.
2) We select the blue die corresponding to the value observed at point (1)
3) Finally, we throw the blue die and obtain a word.

We *condition* this generative process w.r.t. the observation of the corpus.

Example: Topics Modeling (Latent Dirichlet Allocation)

Posterior probability of the blue dice

team game season player play games point run coach hit

government zzz_united_states official military war zzz_u_s palestinian leader zzz_israel zzz_american

family home friend father mother son children told wife night

campaign zzz_george_bush president zzz_al_gore election political zzz_bush vote republican democratic

film movie music show play song director actor character movies

...

Example: Topics Modeling (Latent Dirichlet Allocation)

Corpus (C)

dID	ps	wID	Φ
D_1	1	The	$e_{1,1}$
D_1	2	Cat	$e_{1,2}$
D_1	3	Naps	$e_{1,3}$
...	...	...	...
D_2	1	Once	$e_{2,1}$
D_2	2	Upon	$e_{2,2}$
...	...	...	...
D_D	L	End	$e_{D,L}$

Topics (T)

tID	wID	Φ	A
T_1	Abate	$b_1 = v_1$	$\beta_{1,1}$
T_1	Abdicate	$b_1 = v_2$	$\beta_{1,2}$
...	...	...	...
T_1	Zealous	$b_1 = v_W$	$\beta_{1,w}$
...	...	...	...
T_k	Abate	$b_k = v_1$	$\beta_{k,1}$
...	...	...	...
T_k	Zealous	$b_k = v_W$	$\beta_{k,w}$

Documents (D)

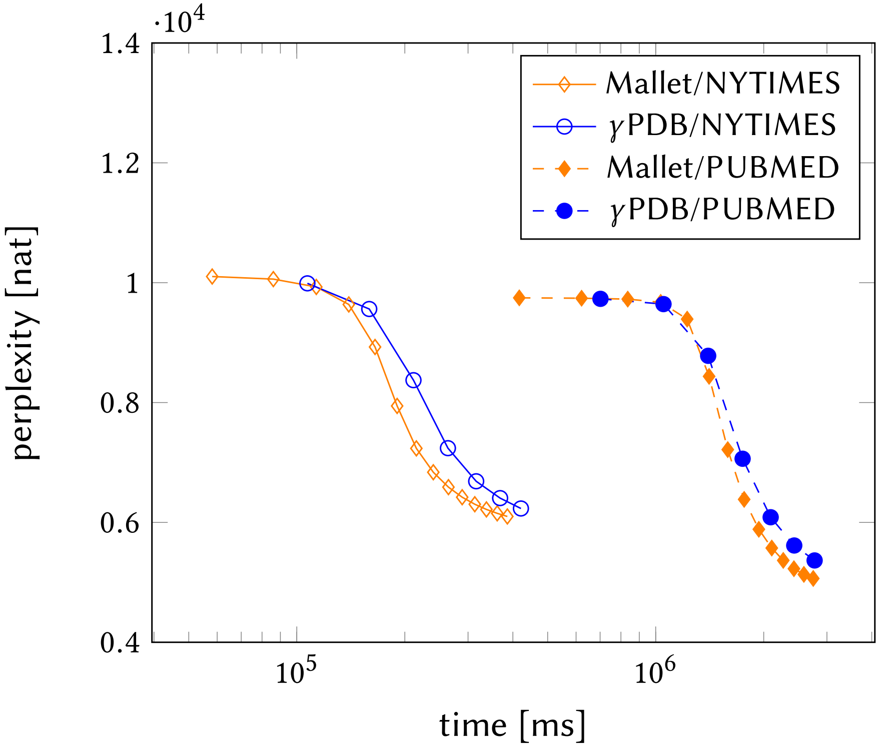
dID	tID	Φ	A
D_1	T_1	$a_1 = t_1$	$\alpha_{1,1}$
D_1	T_2	$a_1 = t_2$	$\alpha_{1,2}$
...	...	...	...
D_1	T_K	$a_1 = t_K$	$\alpha_{1,K}$
...	...	...	...
D_D	T_1	$a_D = t_1$	$\alpha_{D,1}$
...	...	...	...
D_D	T_K	$a_D = t_K$	$\alpha_{D,K}$

$$\boldsymbol{q}_{\text{lda}} = \pi_{\text{dID, ps, wID}}\left((C \bowtie\!:: D) \bowtie\!:: T\right)$$

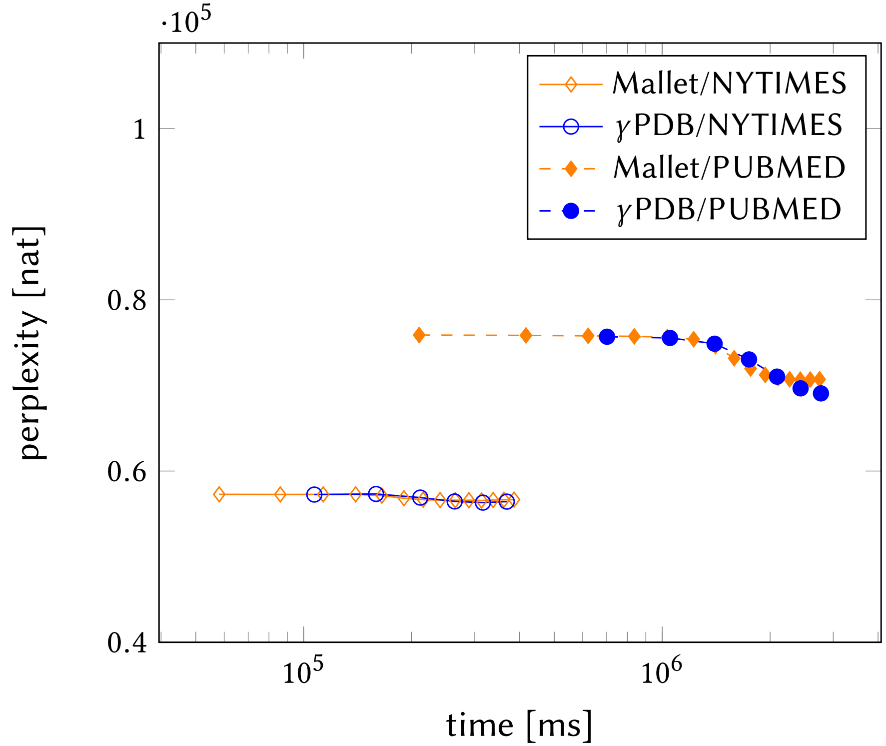
$$\downarrow$$

$$\phi_{d,p,w} = \bigvee_{i=1}^{K} \left[(a_d[e_{d,p}] = t_i) \wedge (b_i[(a_d[e_{d,p}] = t_i)] = v_w)\right]$$

Experimental Evaluation: Latent Dirichlet Allocation



(a) Training-set perplexity



(b) Test-set perplexity

Current work: Variational Inference

(a) Noisy Image

(b) Gibbs sampler denoising

(c) Variational inference denoising

Next Steps

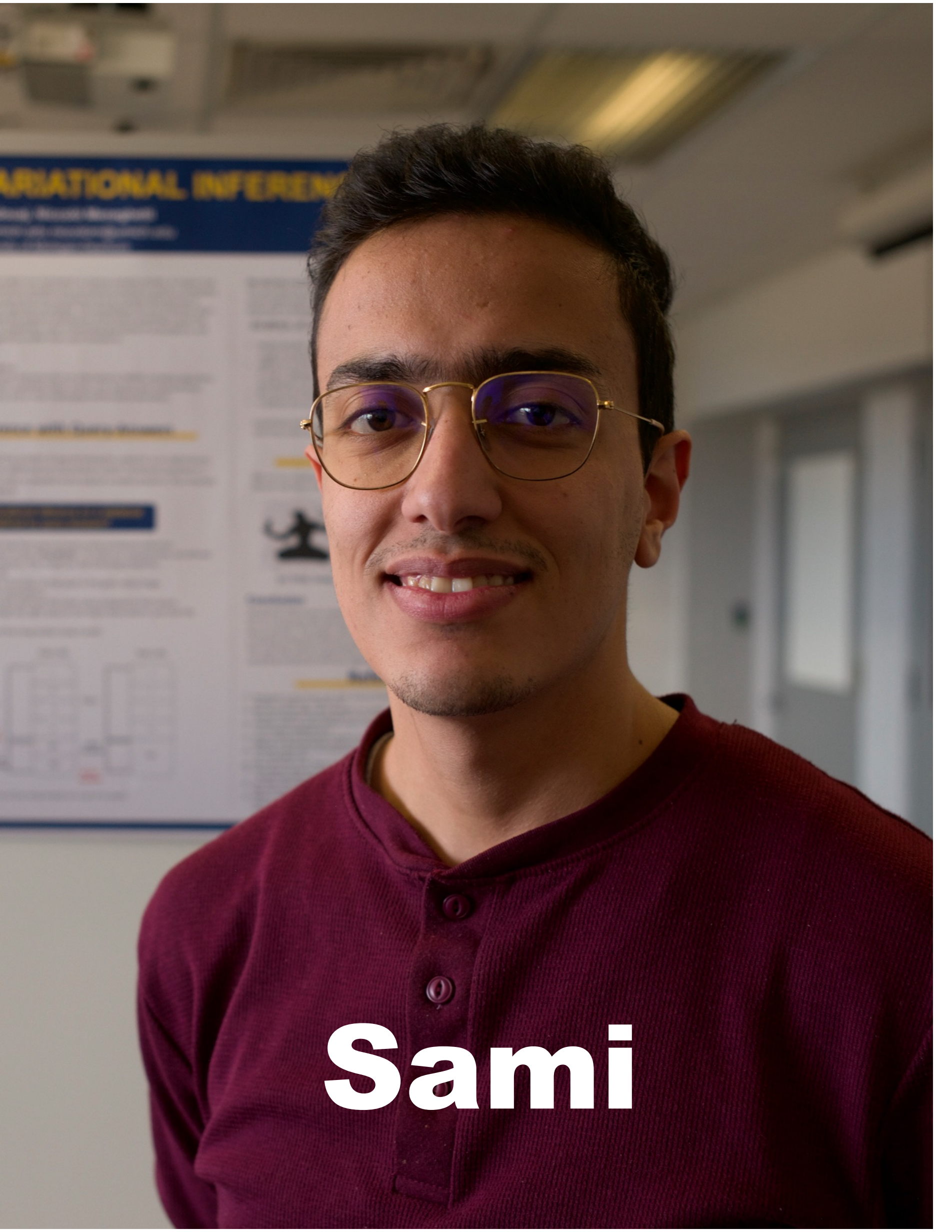
(1) Factorized Inference

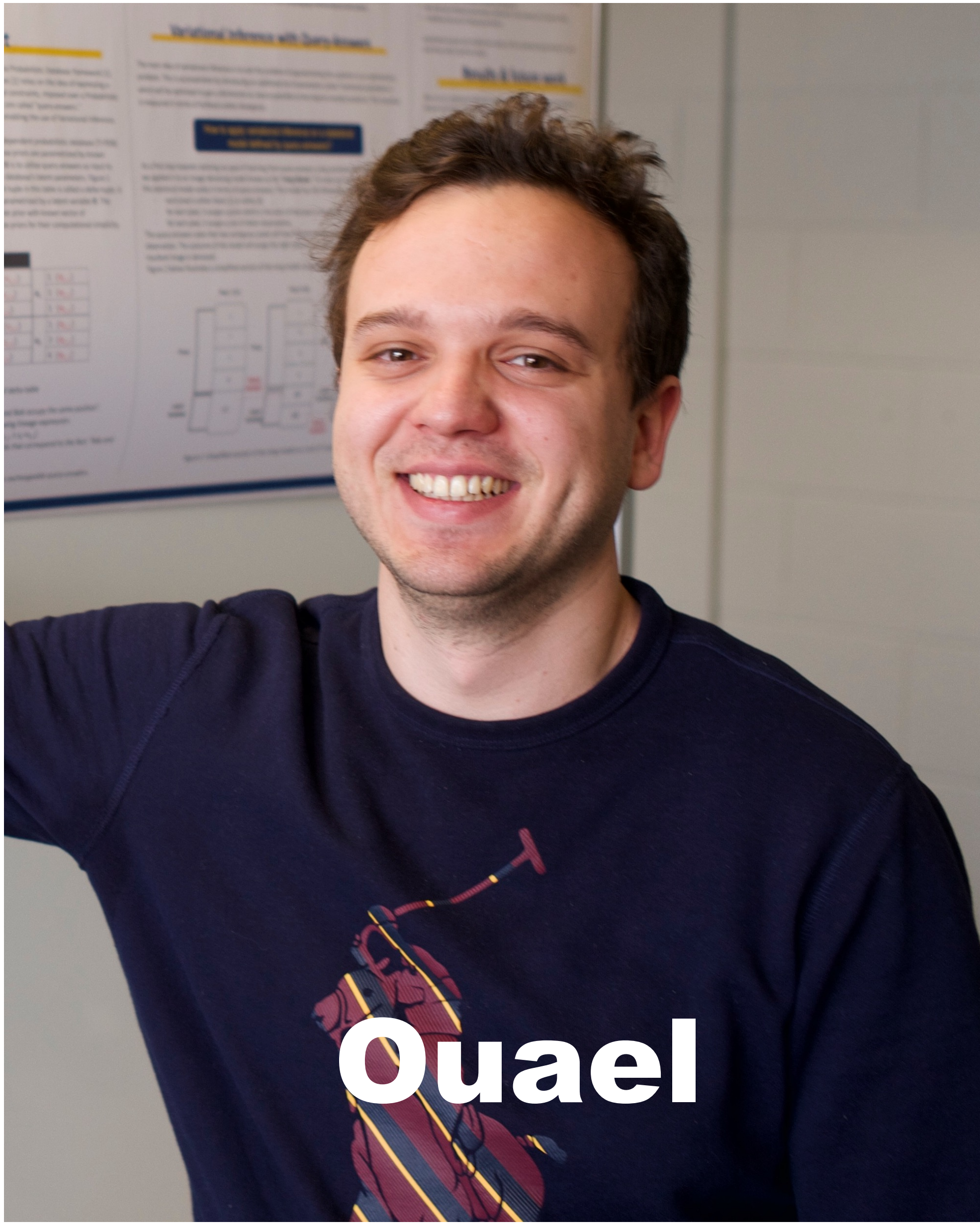
Yurochkin, Mikhail, and XuanLong Nguyen.
"Geometric Dirichlet means algorithm for topic inference"
Advances in Neural Information Processing Systems 29 (2016).

(2) Fairness through rel. constraints

Salimi, Babak, Luke Rodriguez, Bill Howe, and Dan Suciu.
"Interventional fairness: Causal database repair for algorithmic fairness"
In *Proceedings of the 2019 International Conference on Management of Data*, pp. 793-810. 2019.

We have two posters!





References

(1) Probabilistic Programming Datalog

Bárány, Vince, Balder Ten Cate, Benny Kimelfeld, Dan Olteanu, and Zografoula Vagena. "Declarative probabilistic programming with datalog." *ACM Transactions on Database Systems (TODS)* 42, no. 4 (2017): 1-35.

(2) SimSQL/BUDS

Cai, Zhuhua, Zografoula Vagena, Luis Perez, Subramanian Arumugam, Peter J. Haas, and Christopher Jermaine. "Simulation of database-valued Markov chains using SimSQL." In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pp. 637-648. 2013.

Thank you!

(Questions?)