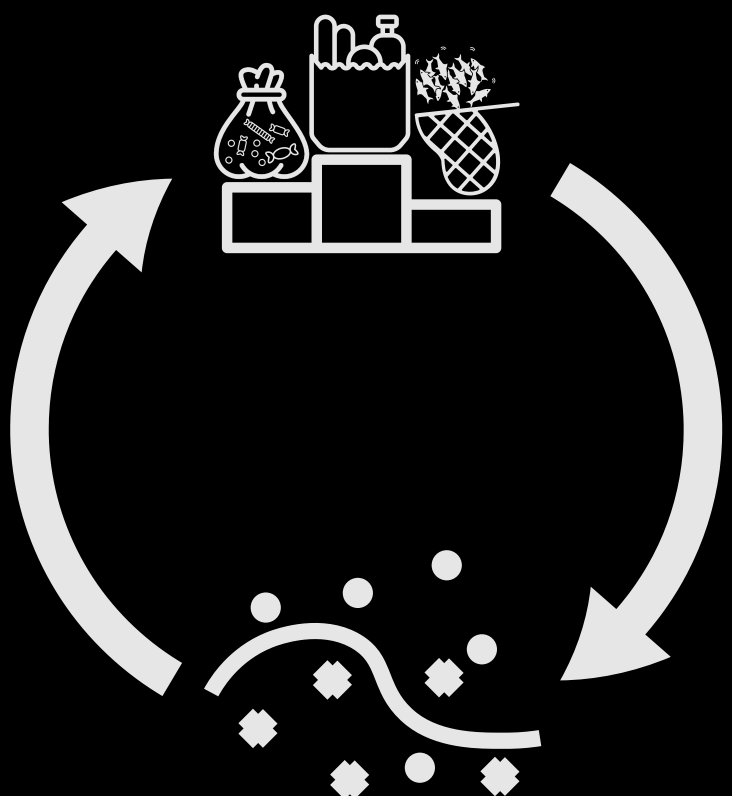
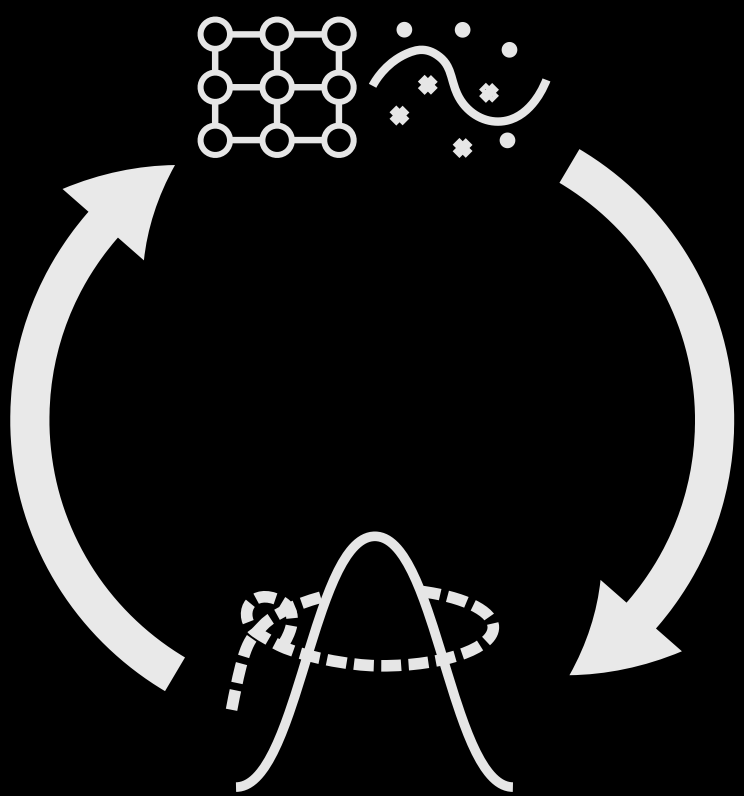
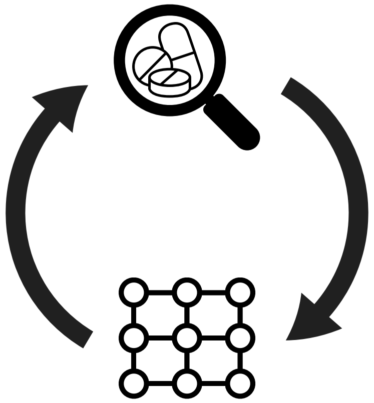
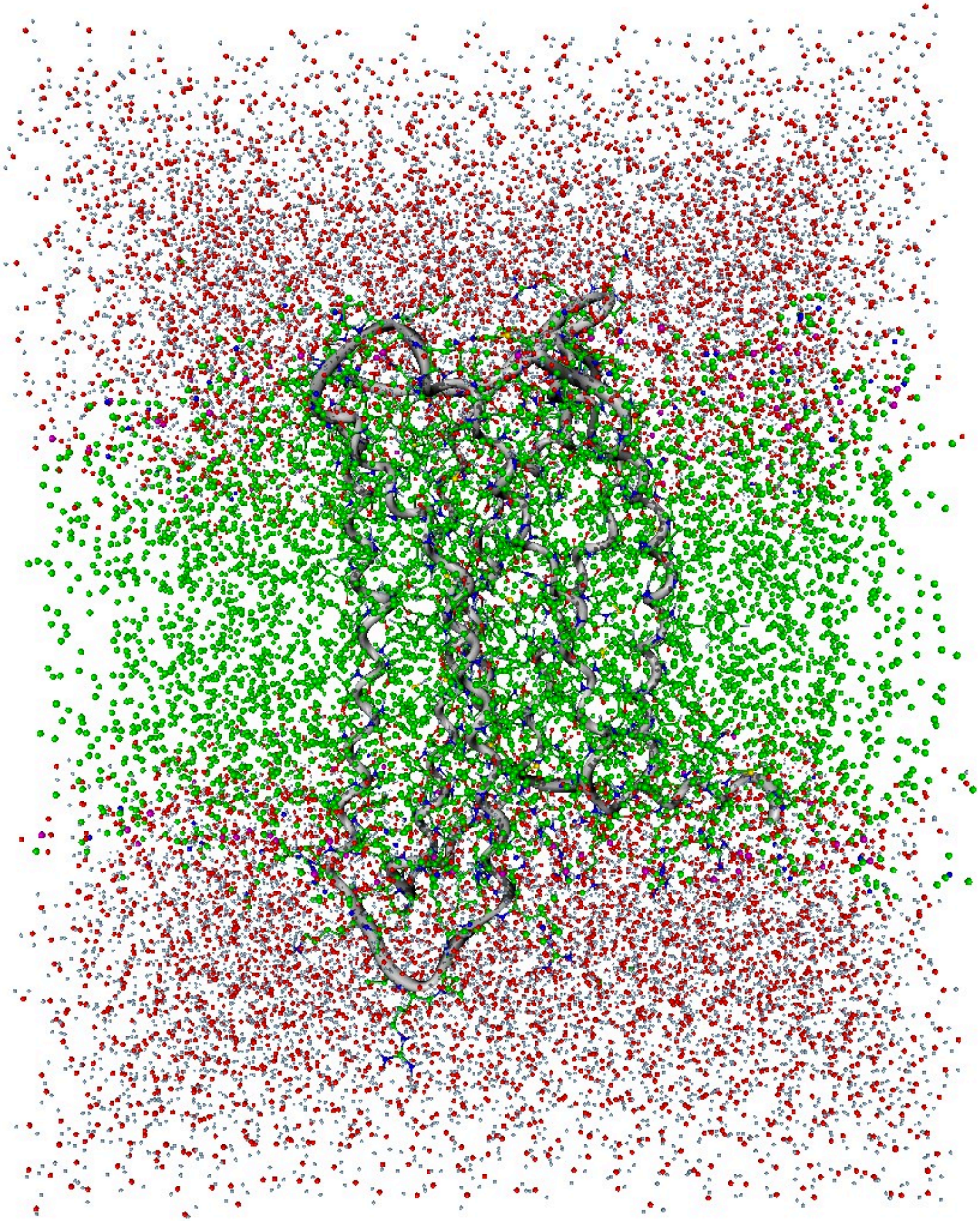
Probabilistic Modeling of Structure in Science: Statistical Physics to Recommender Systems

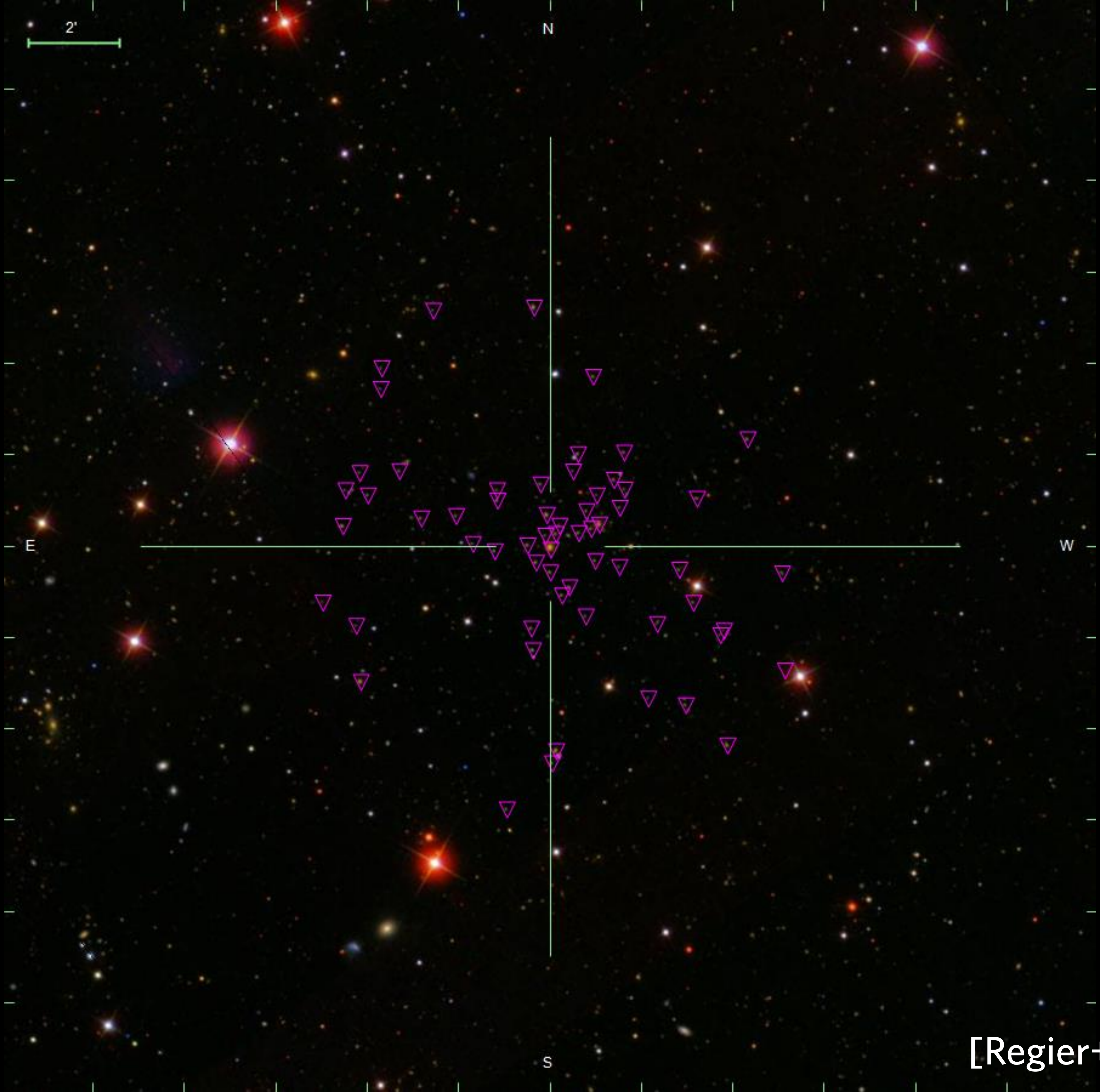
Jaan Altosaar

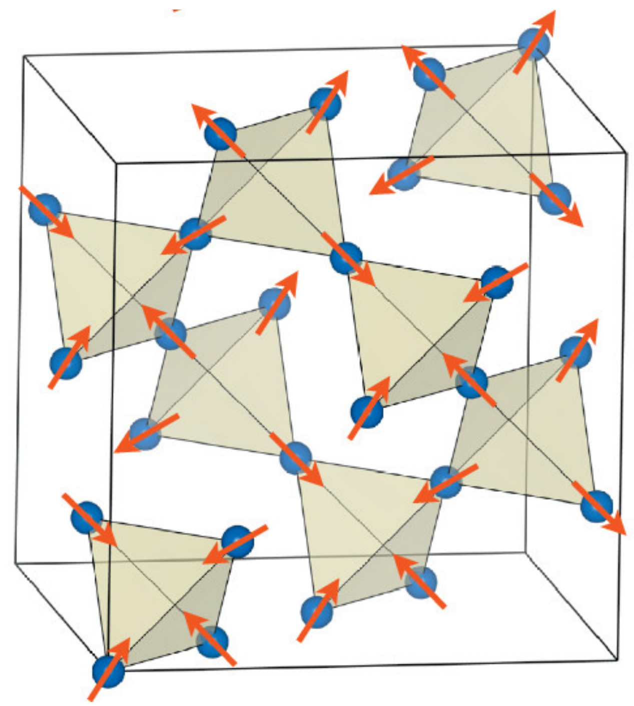




[Klimm+ 2018]



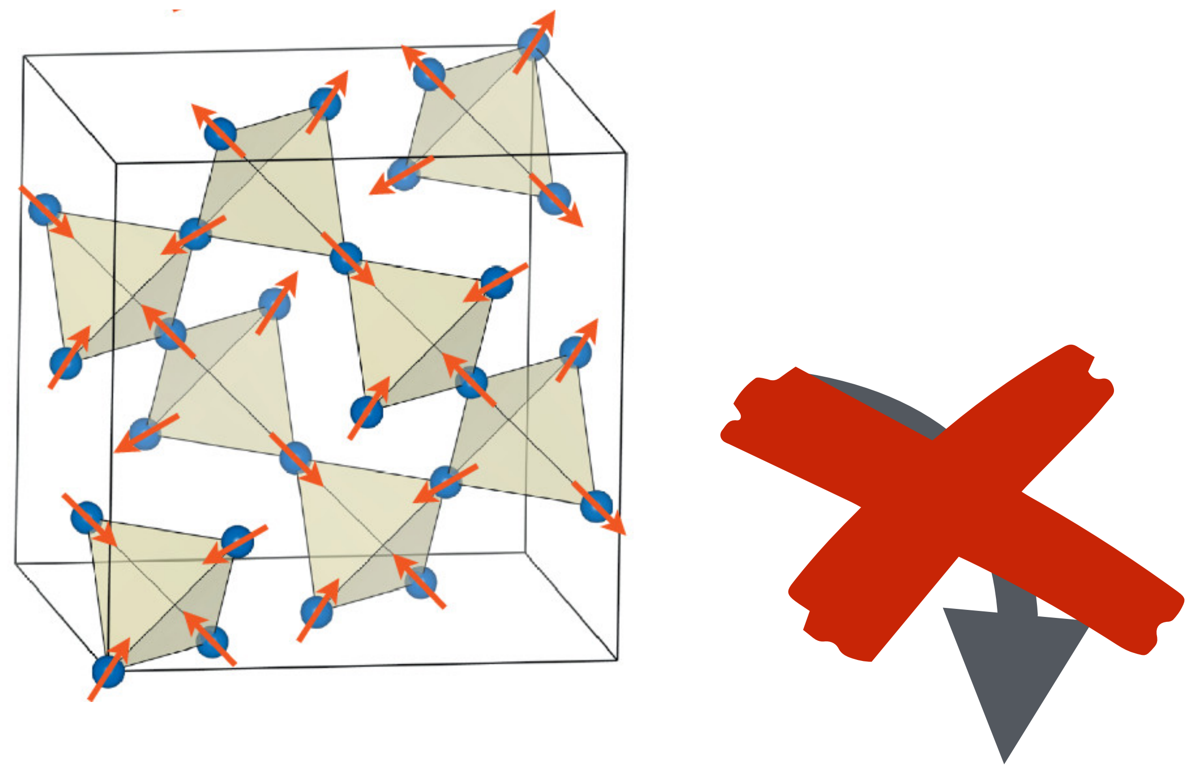
[Regier+ 2018]



$$C_v = f(\mathcal{Z})$$

$$p(z) = \frac{e^{-\beta H(z)}}{\mathcal{Z}}$$

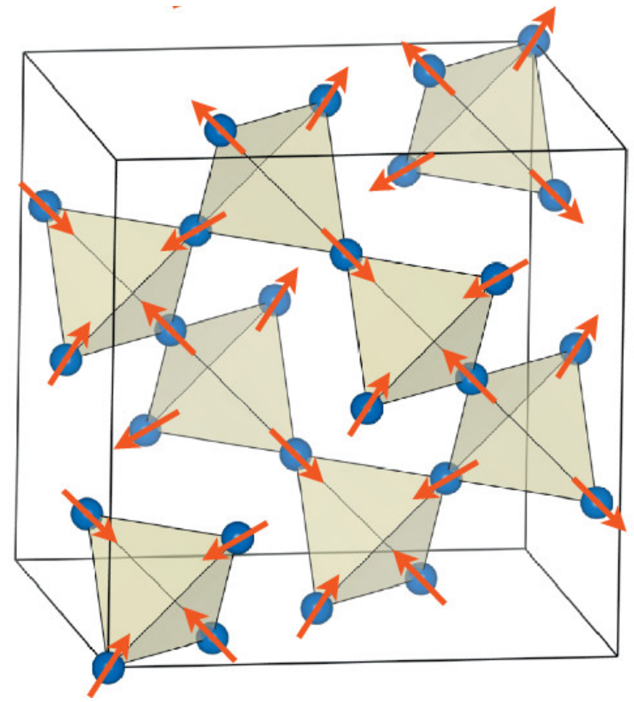
[Weibe+ 2015; Henelius+ 2016]



$$p(z) = \frac{e^{-\beta H(z)}}{\mathcal{Z}}$$

$$\mathcal{Z} = \sum_{i} \sum_{z_i = \pm 1} e^{-\beta H(z_i)}$$

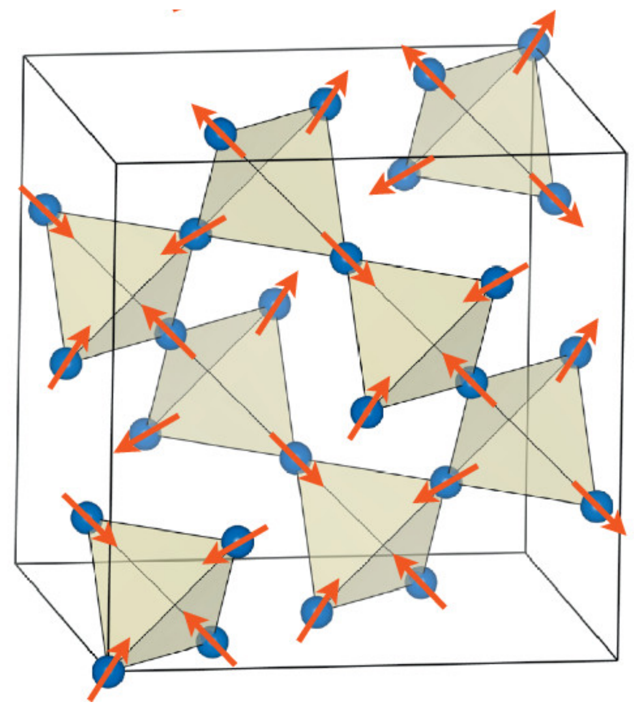
[Weibe+ 2015; Henelius+ 2016]



- Markov Chain Monte Carlo
  - Approximate $f(\mathcal{Z})$

[Neal 1993; Hastings 1970]

$$p(z) = \frac{e^{-\beta H(z)}}{\mathcal{Z}}$$



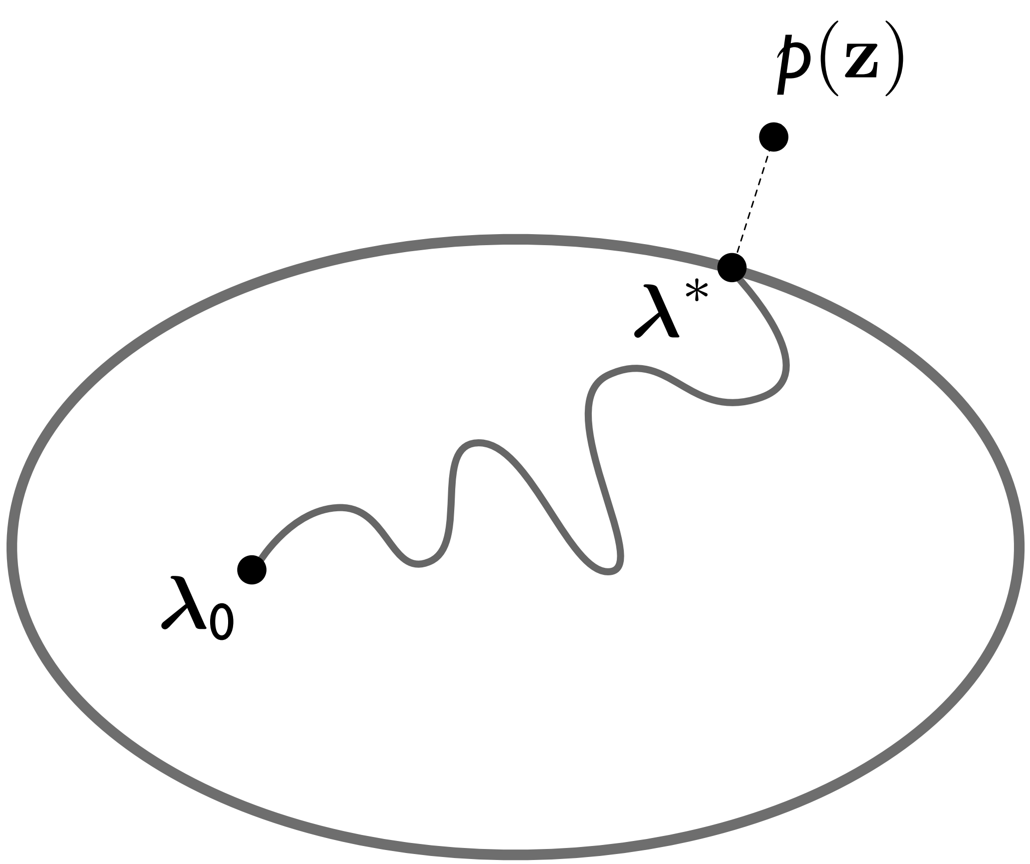
- Variational Inference
  - Learn approximation $q(\mathbf{z})$

    [Blei+ 2016; Peterson+ 1987]

$$p(z) = \frac{e^{-\beta H(z)}}{\mathcal{Z}}$$

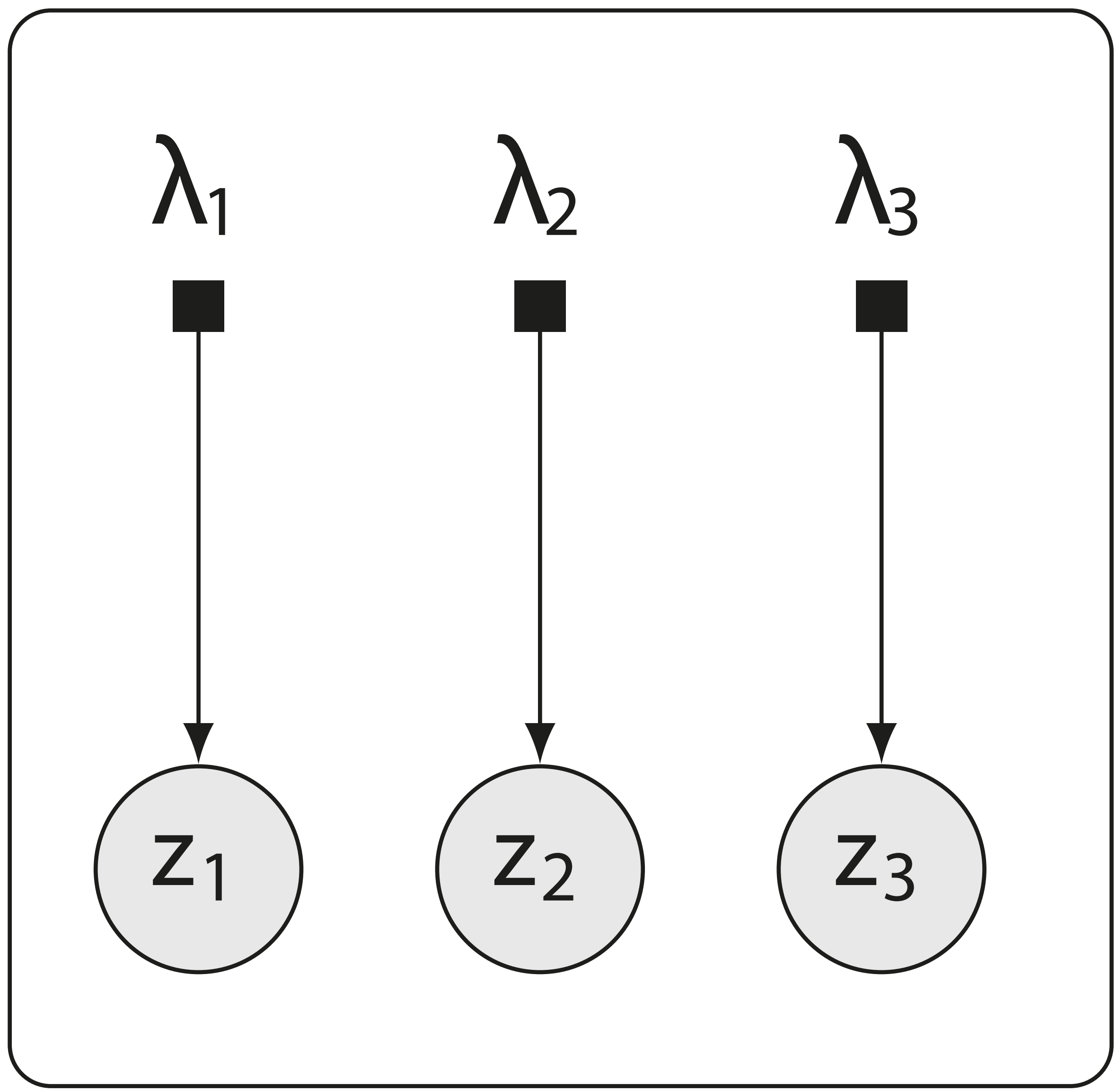
Variational Inference

$$\mathrm{KL}\left(q(\mathbf{z};\boldsymbol{\lambda}) \parallel p(\mathbf{z})\right) = \mathbb{E}_q[\log q(\mathbf{z};\boldsymbol{\lambda})] - \mathbb{E}_q[-\beta H(\mathbf{z})] + \log \mathcal{Z}$$



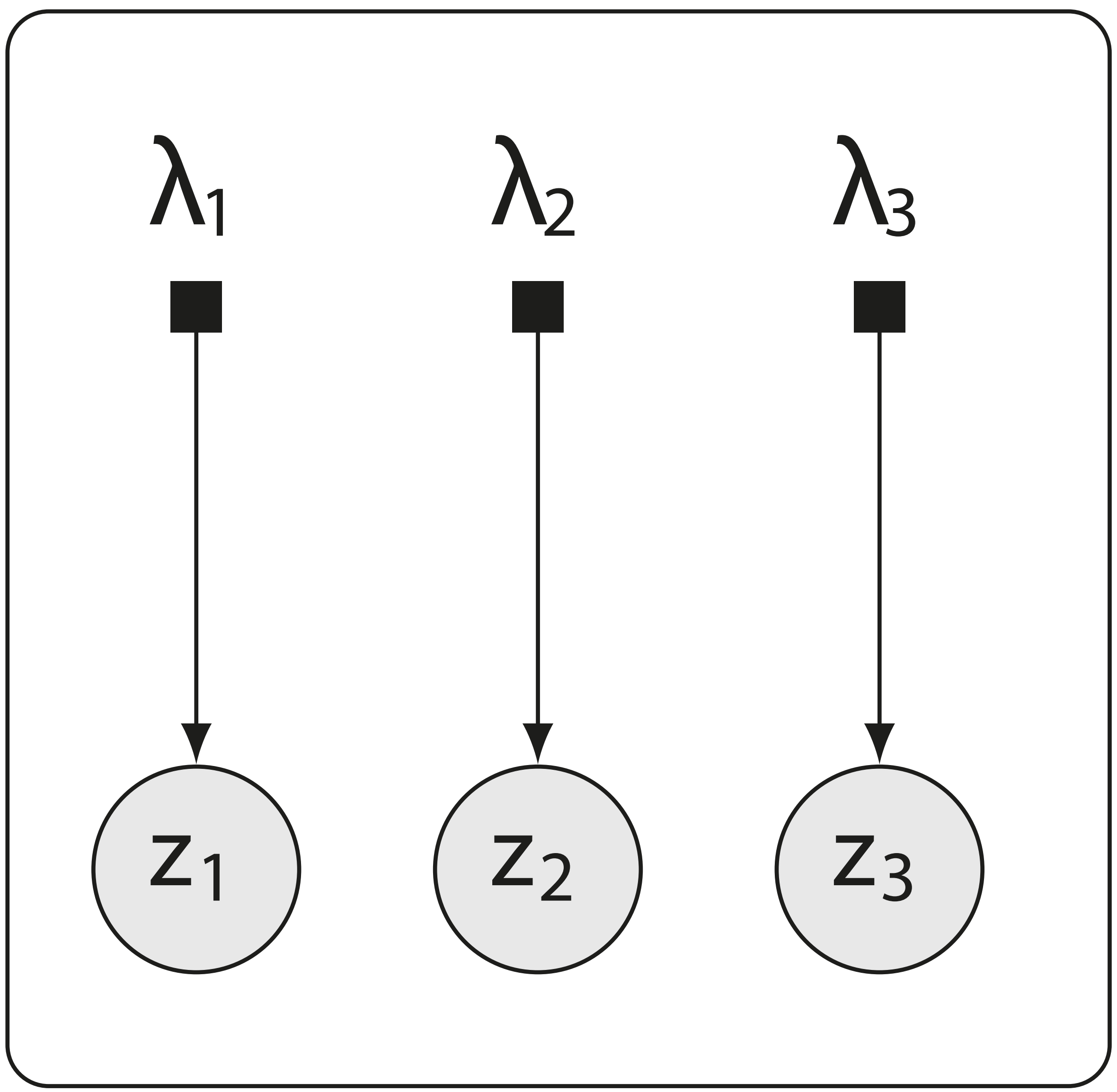
Mean Field Variational Family

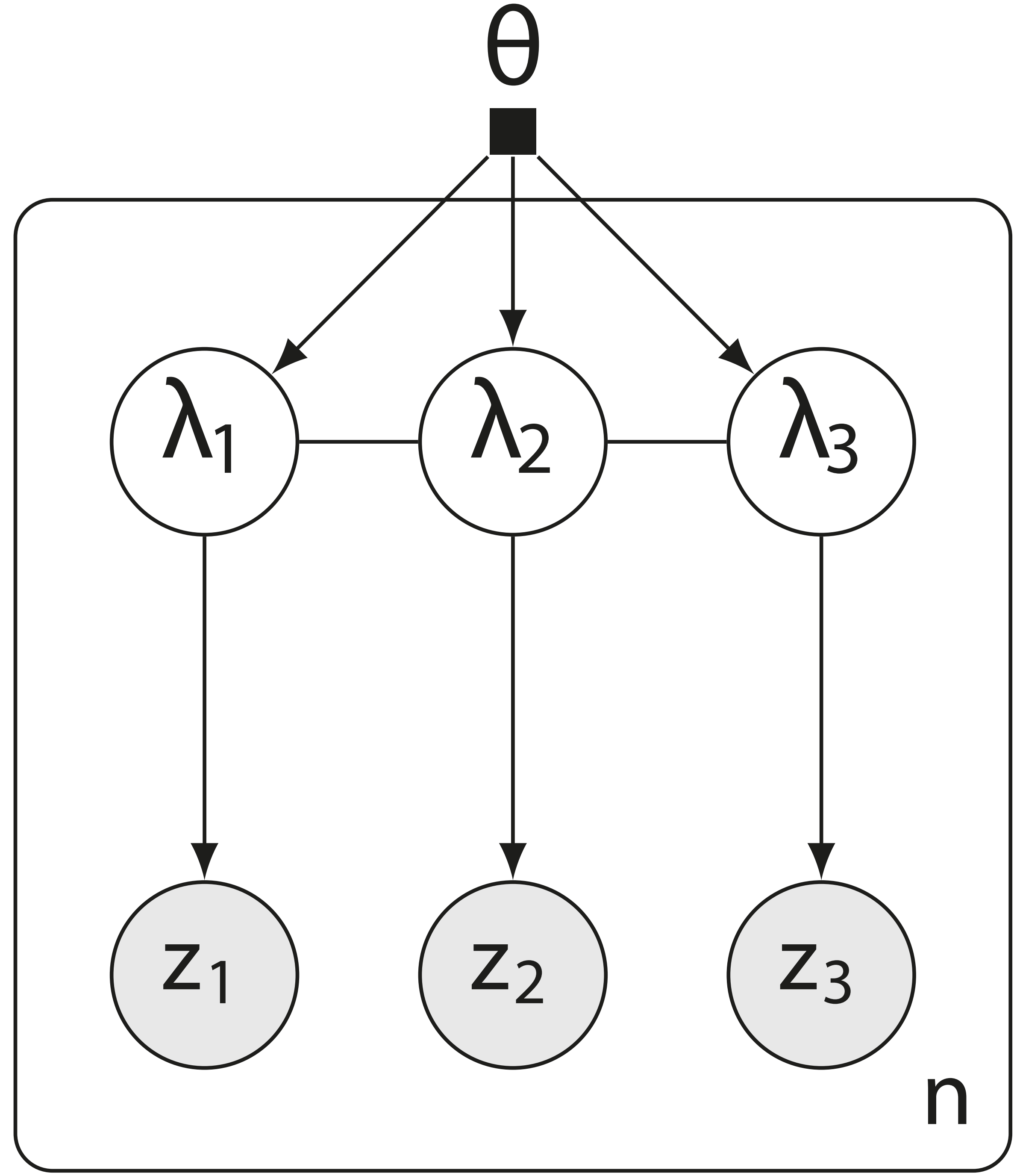
$$q(\mathbf{z};\boldsymbol{\lambda}) = \prod_{i=1}^{d} q(\mathbf{z}_i;\boldsymbol{\lambda}_i)$$



Hierarchical Variational Models

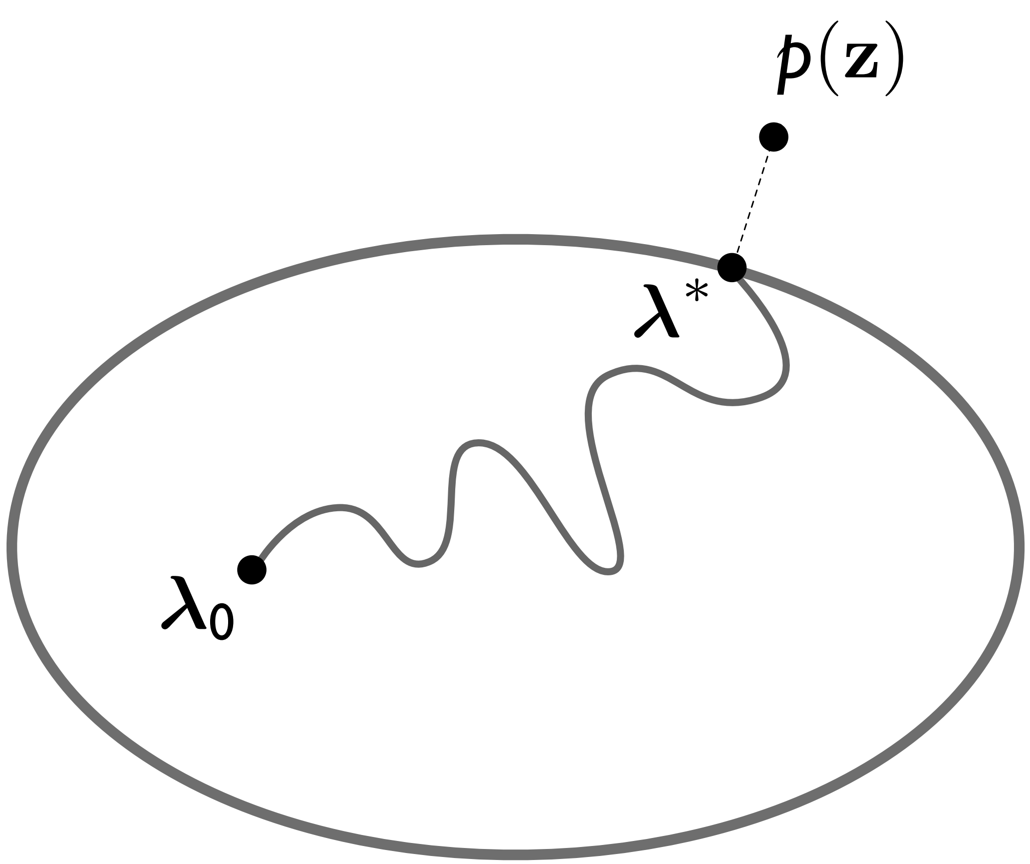
$$q(\mathbf{z};\boldsymbol{\theta}) = \int \prod_{i=1}^{d} q(\mathbf{z}_i \mid \boldsymbol{\lambda}_i)\, q(\boldsymbol{\lambda};\boldsymbol{\theta})\, d\boldsymbol{\lambda}$$





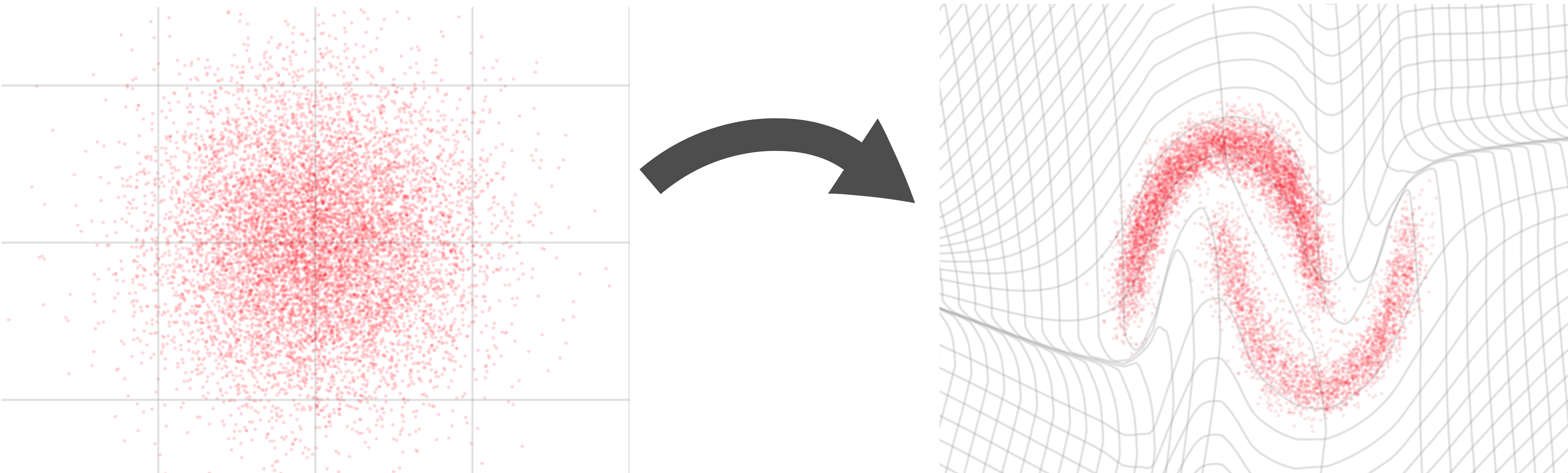
[Ranganath+ 2016]

Recursive Variational Posterior



[Ranganath+ 2016]

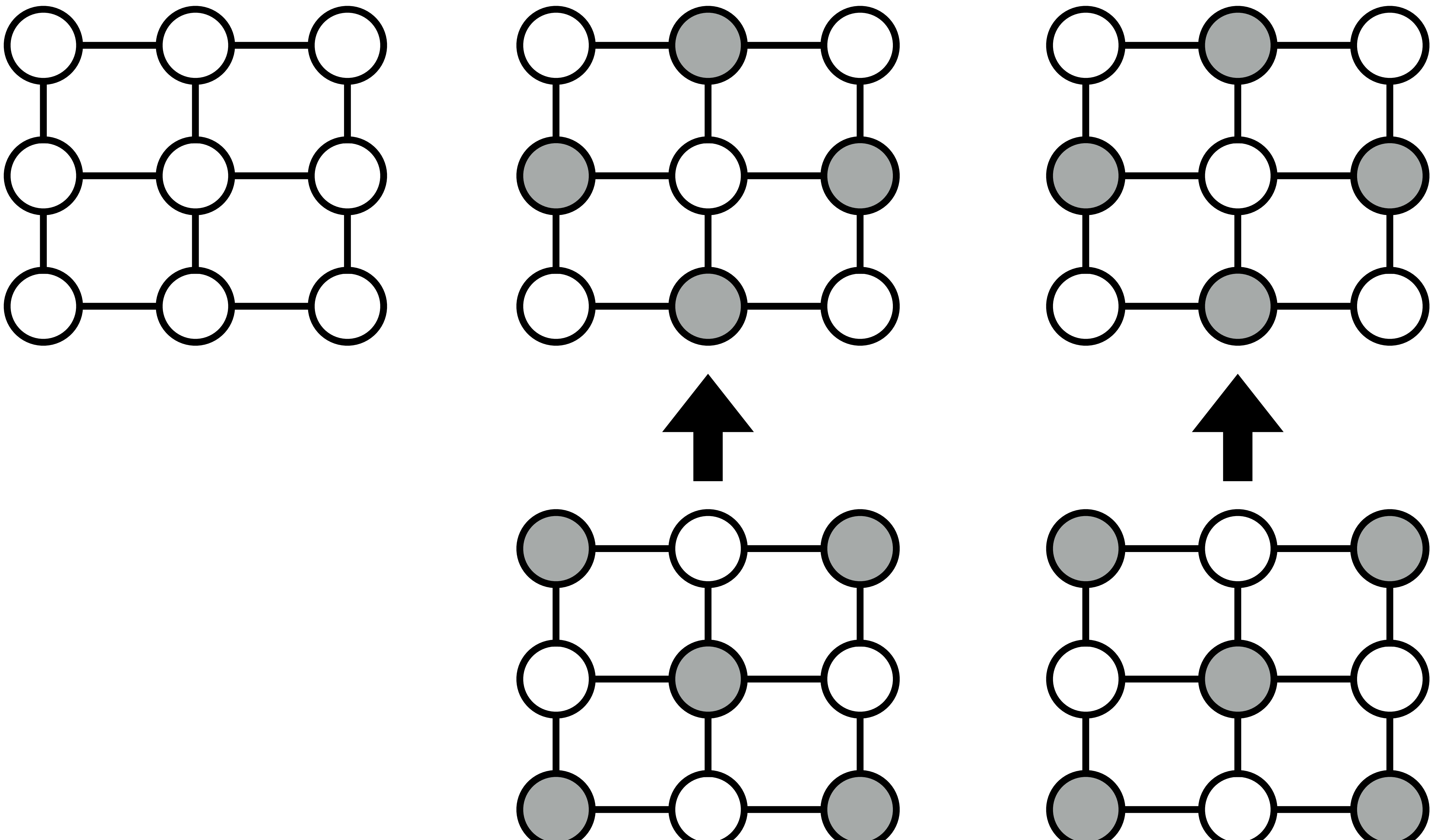
Generative Models: Flows



- Invertible transformations of random variables
- Cheap log density evaluation [Rezende+ 2015; Dinh+ 2016]
- In statistical physics: [Wu+ Phys. Rev. Lett. (2019)]
  - Variational autoregressive networks use PixelCNN [van den Oord+ 2016]

Building Hierarchical Variational Models

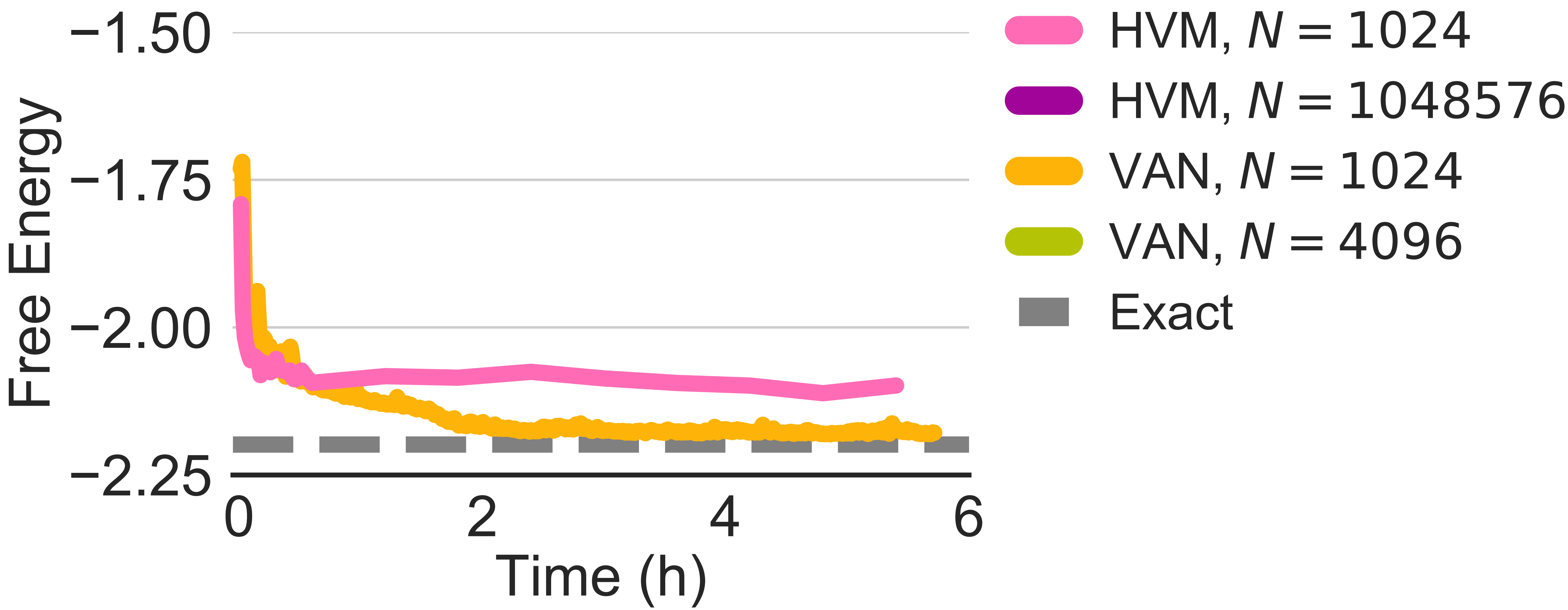
$p(\mathbf{z})$ $\quad q(\boldsymbol{\lambda}; \boldsymbol{\theta})$ $\quad r(\boldsymbol{\lambda} \mid \mathbf{z}; \boldsymbol{\phi})$



[Dinh+ 2016]

Ising Model

- HVM: 5400 parameters; VAN: 700k+
- Free energy: lower is better

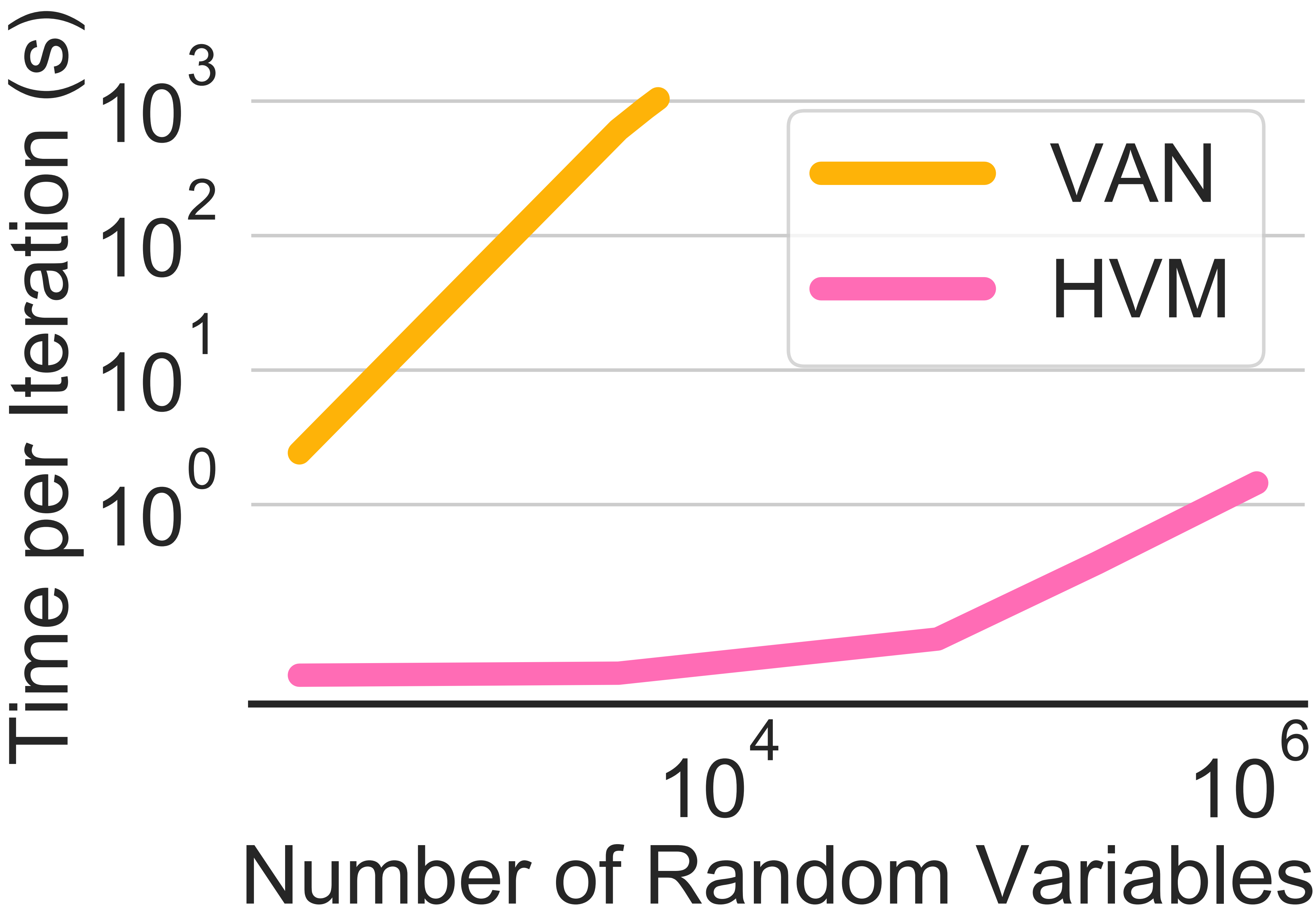


[Altosaar+ 2019]

Sherrington-Kirkpatrick Model

- HVM: 5400 parameters; VAN: 700k+ parameters
- Free energy: lower is better

Scaling to Large Systems



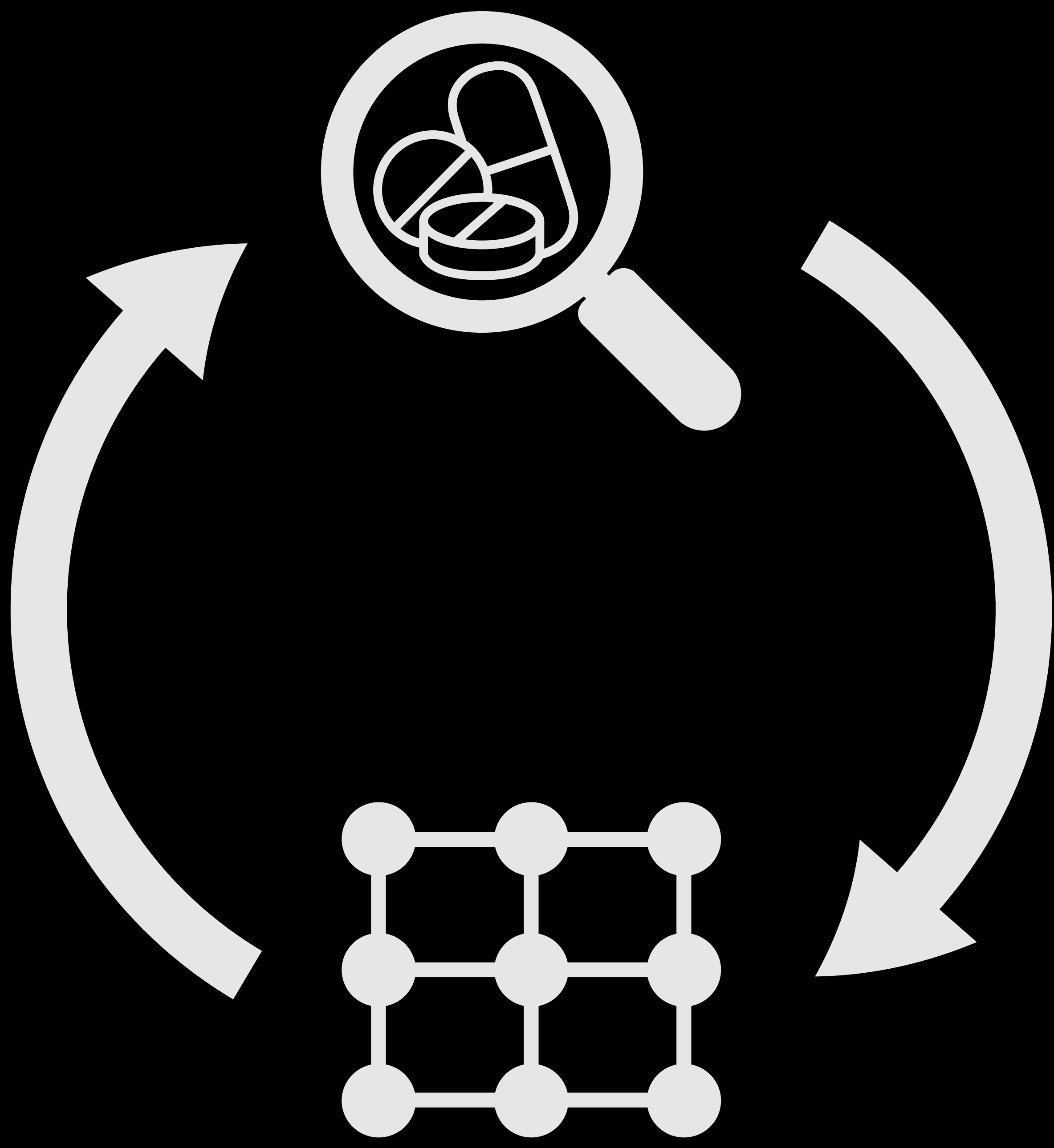
Next Steps

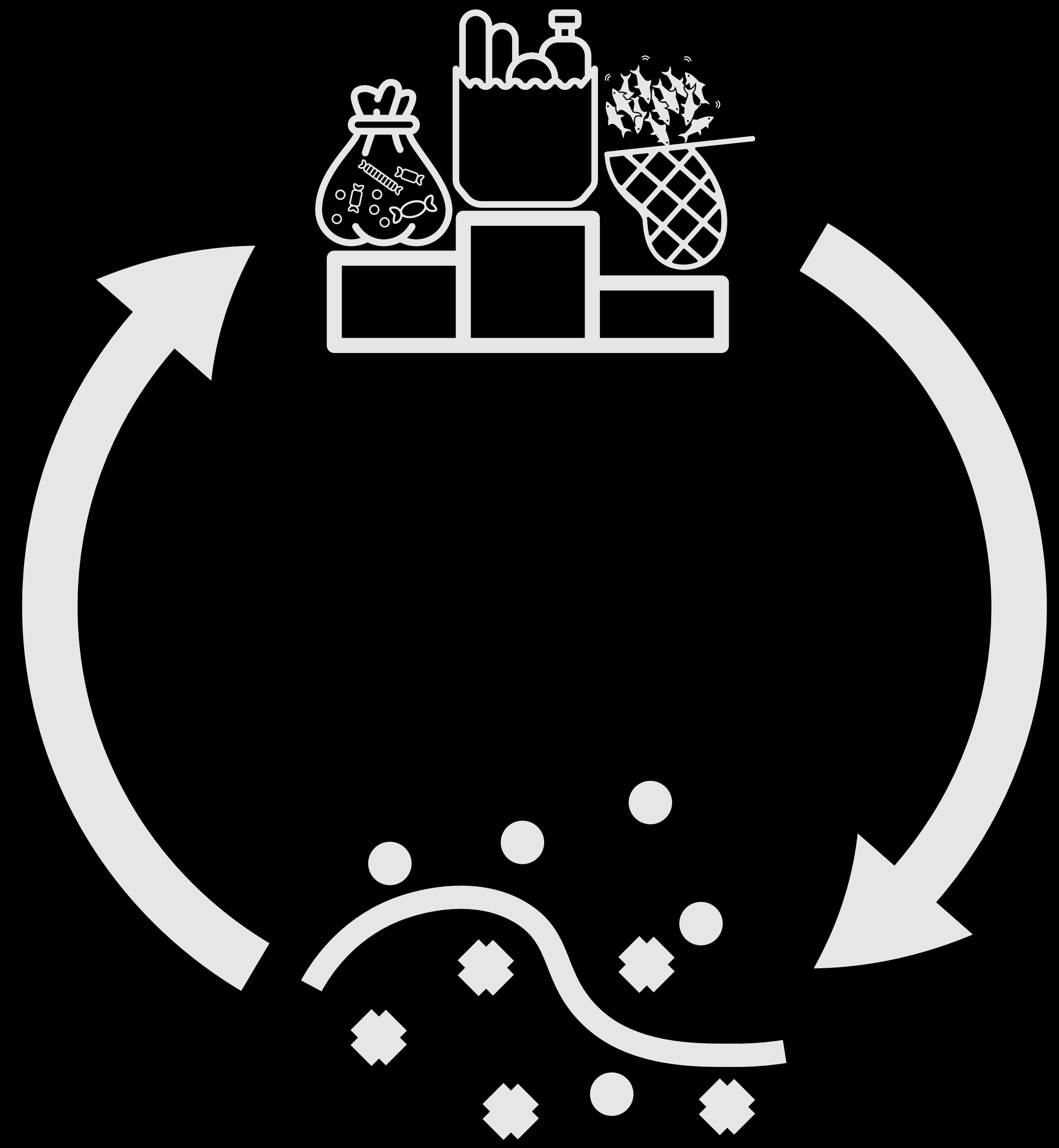
- Use hierarchical variational models as proposal for Monte Carlo methods to reduce burn-in time

- Test on quantum systems for drug screening

- Rao-Blackwellization helps → increase accuracy with tighter lower bounds [Tucker+ 2018]

- Proximity variational inference can also increase accuracy [Altosaar+ 2018]






99¢ ONLY
99 Thanks...!
99 ONLY


Neilson
crème à fouetter
toujours fraîche
173 mL
Nutrition Facts
Valeur nutritive
DUCK BREAST

Emma Coats
@lawnrocket

Follow

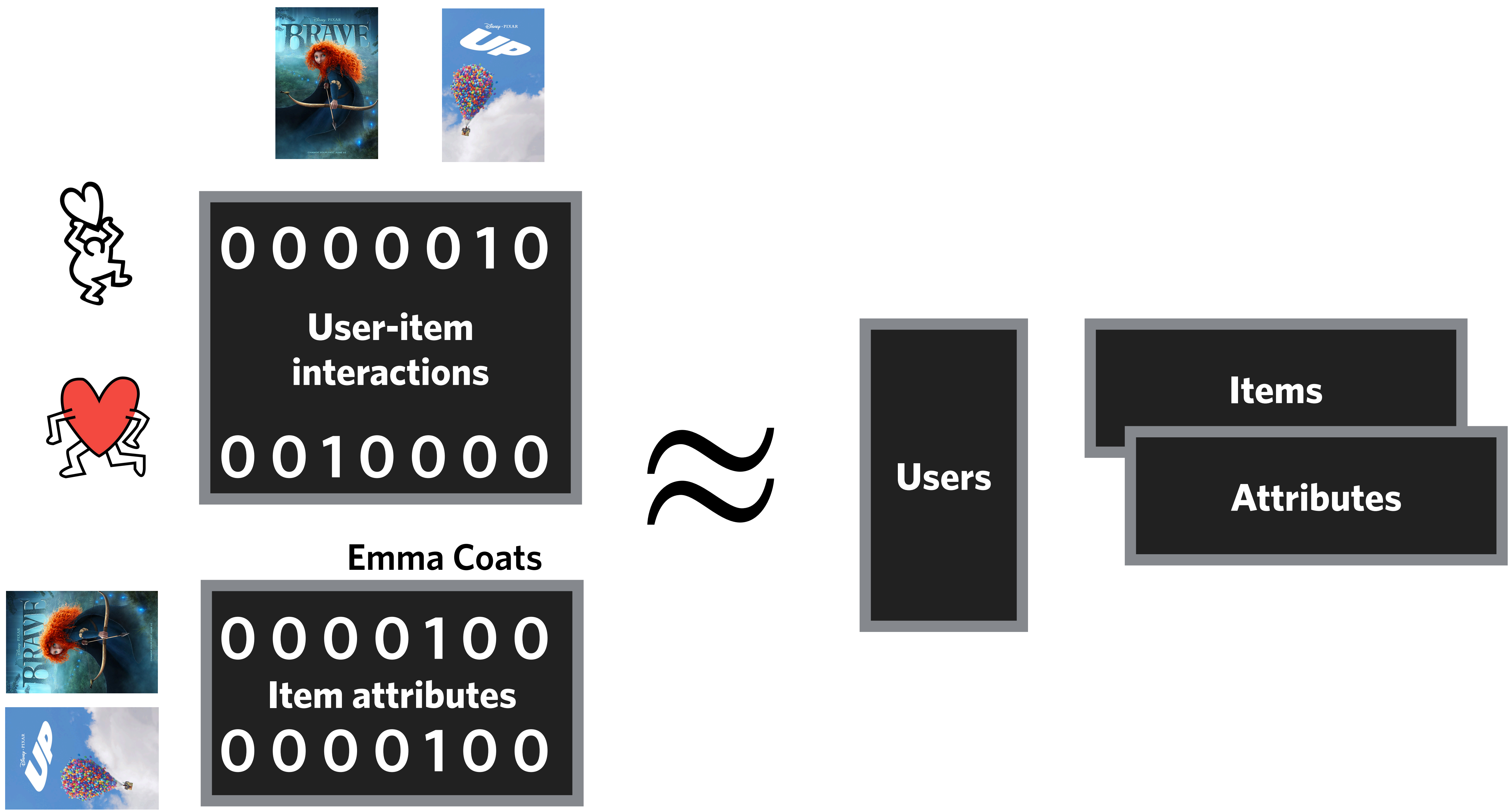
#4: Once upon a time there was ___. Every day, ___. One day ___. Because of that, ___. Because of that, ___. Until finally ___. #storybasics

11:37 AM - 11 May 2012

34 Retweets **33** Likes

4 34 33

Once upon a time there was matrix factorization.



Once upon a time there was matrix factorization. Every day, a new matrix factorization model appears.

Matrix factorization techniques for recommender systems

Y Koren, R Bell, C Volinsky - Computer, 2009 - computer.org

[PDF] datajobs.com

Findit@PUL

AB
fa
re
te

Factorization meets the item embedding: Regularizing matrix factorization w
item co-occurrence

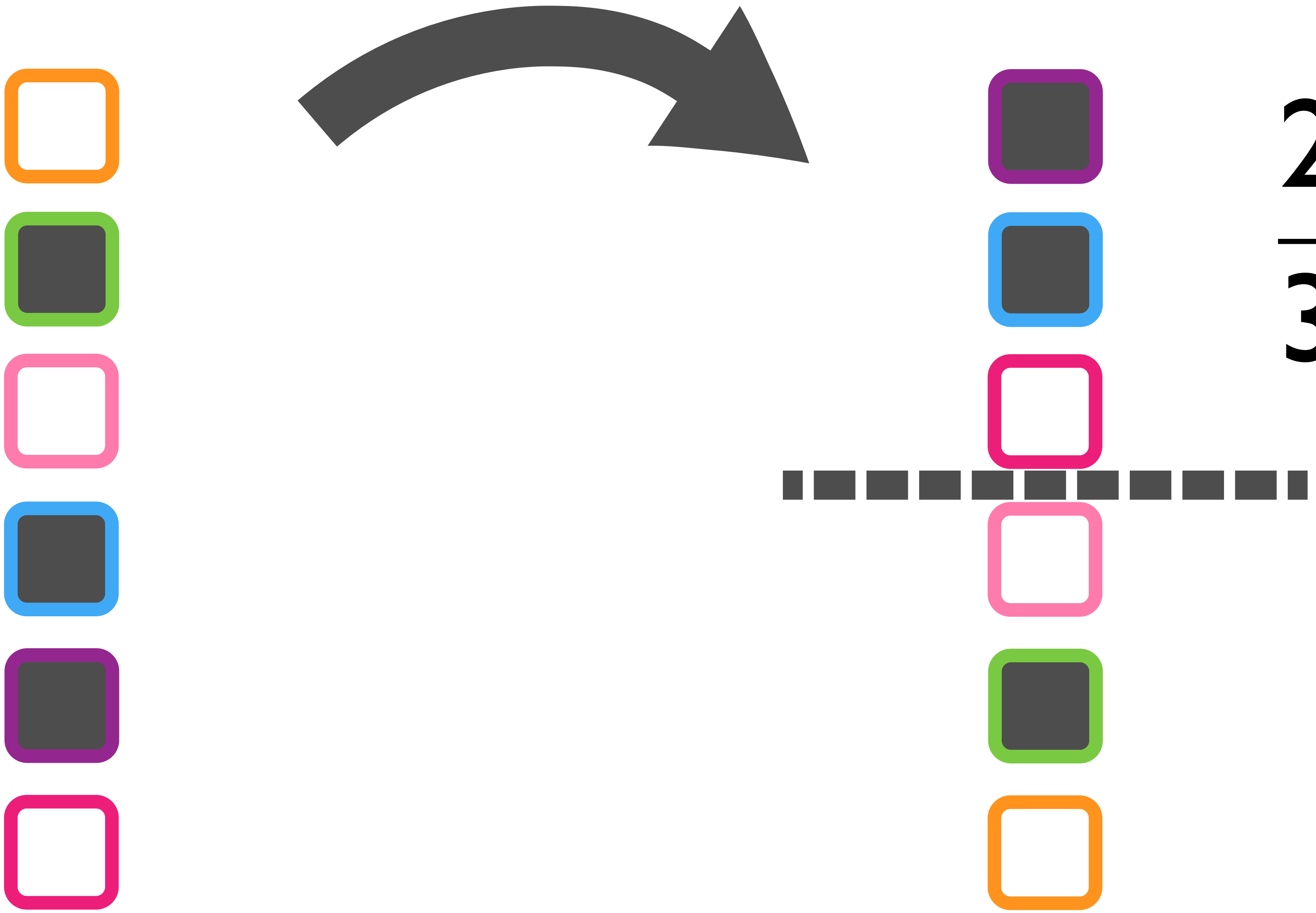
D Liang, J Altosaar, L Charlin, DM Blei - … of the 10th ACM conference on …, 2016 - dl.acm.org

Matrix factorization (MF) models and their extensions are standard in modern recommender systems. MF models decompose the observed user-item interaction matrix into user and item latent factors. In this paper, we propose a co-factorization model, CoFactor, which jointly decomposes the user-item interaction matrix and the item-item co-occurrence matrix with shared item latent factors. For each pair of items, the co-occurrence matrix encodes the number of users that have consumed both items. CoFactor is inspired by the recent success …

Cited by 137 Related articles All 10 versions

Once upon a time there was matrix factorization. Every day, a new matrix factorization model appears. One day, recall was used for evaluation.

$$p(y_{um} = 1 \mid u, m)$$

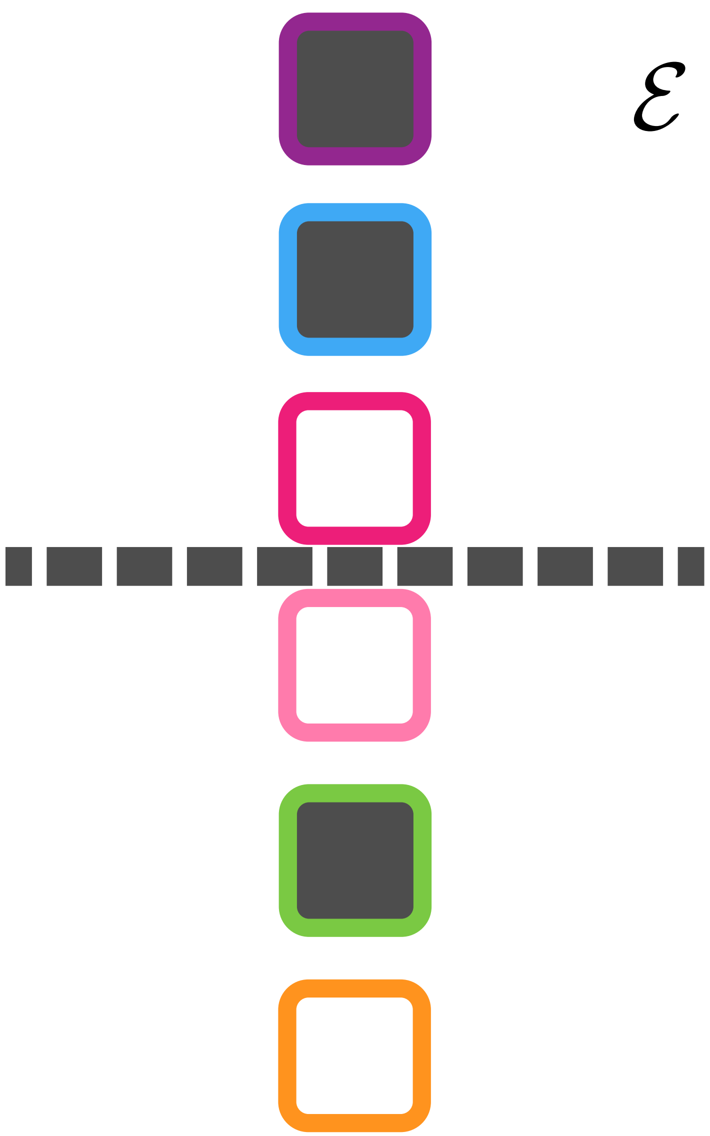


$$\frac{2}{3}$$

[Ebesu+ 2018; Yang+ 2018]

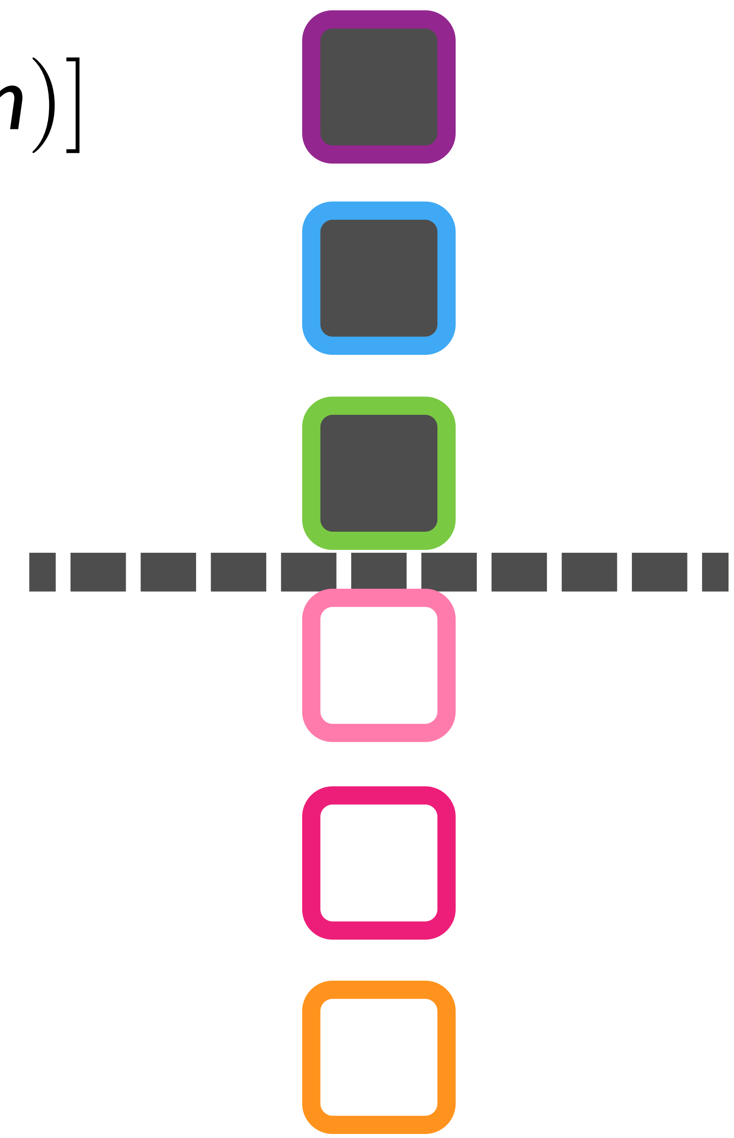
Once upon a time there was matrix factorization. Every day, a new matrix factorization model appears. One day recall was used for evaluation. Because of that, zero worst-case error classifiers are optimal.

$\mathcal{E} > 0$



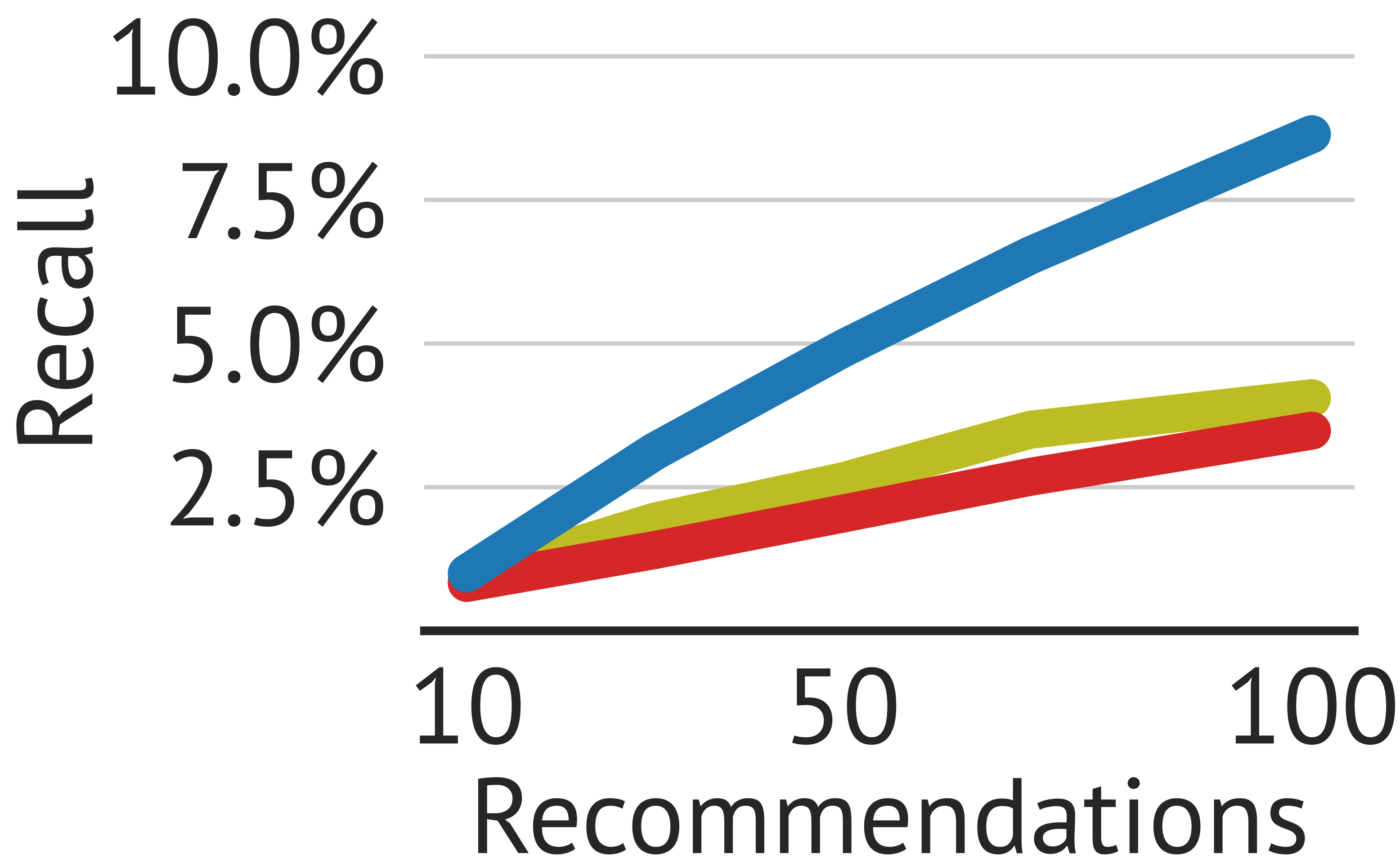
$$\mathcal{E} = \max_{(u,m)\in\mathcal{D}} \mathbb{I}\left[\hat{y}(u,m) \neq y(u,m)\right]$$

$\mathcal{E} = 0$



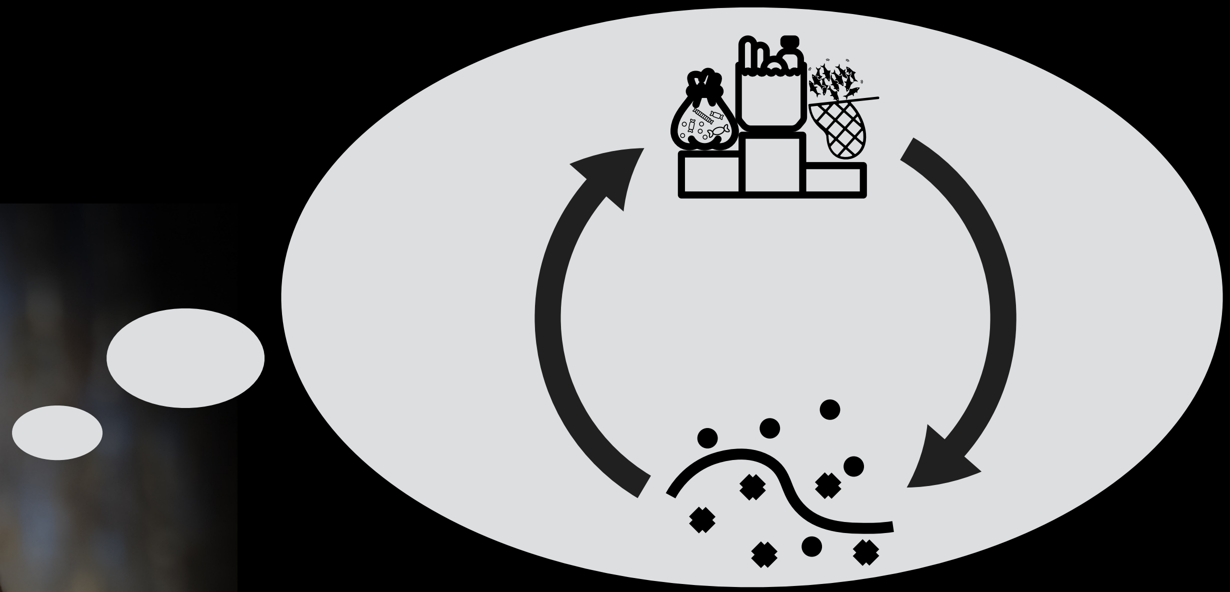
[Altosaar+ 2020]

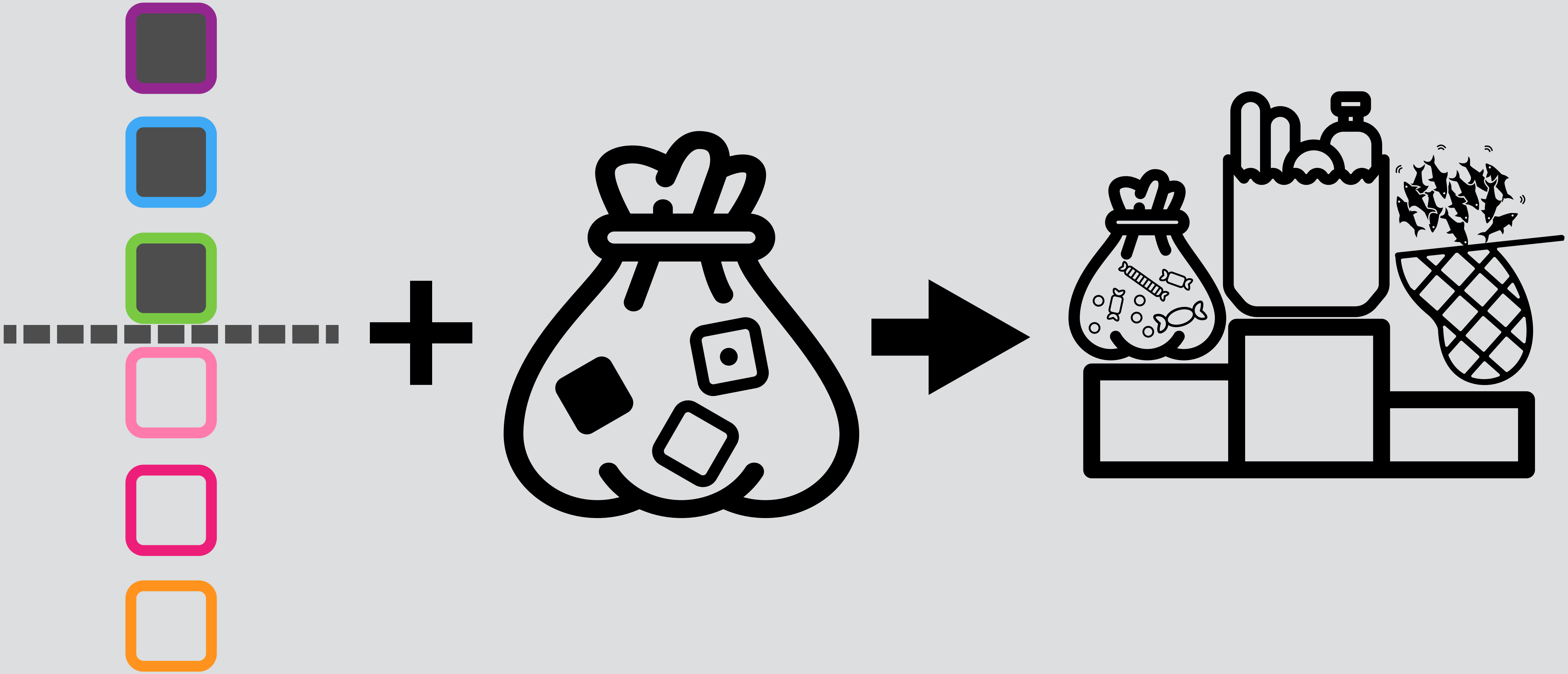
Once upon a time there was matrix factorization. Every day, a new matrix factorization model appears. One day recall was used for evaluation. Because of that, zero worst-case error classifiers are optimal. Because of that, they outperform matrix factorization.



[Altosaar+ 2020]

Once upon a time there was matrix factorization. Every day, a new matrix factorization model appears. One day recall was used for evaluation. Because of that, zero worst-case error classifiers are optimal. Because of that, they outperform matrix factorization. Until finally Jaan thought twice before dismissing classifiers as recommenders.





	Attributes (foods)									Users				
Items (meals)	Pizza	Eggs	Taco	Salad	Avocado	Chicken	Sardines	Beer	Coffee	1	2	3	4	5
Breakfast pizza with coffee	✔	✔							✔		✔		✔	
Dinner pizza	✔						✔	✔		✔			✔	
Small salad				✔	✔		✔					✔		✔
Big salad		✔		✔	✔	✔			✔			✔	✔	
Taco			✔		✔	✔			✔					✔
Sardine taco			✔				✔			✔	✔		✔	

RankFromSets

$$p(y_{um} = 1 \mid u, m) = \sigma\left(f(u, x_m)\right)$$

- f is a neural network, invariant to permutation of $\boldsymbol{x_m}$
- This choice of model can maximize recall [Altosaar+ 2020]

RankFromSets

$$p(y_{um} = 1 \mid u, m) = \sigma\left(f(u, x_m)\right)$$

$$f(u, x_m) =$$

RankFromSets

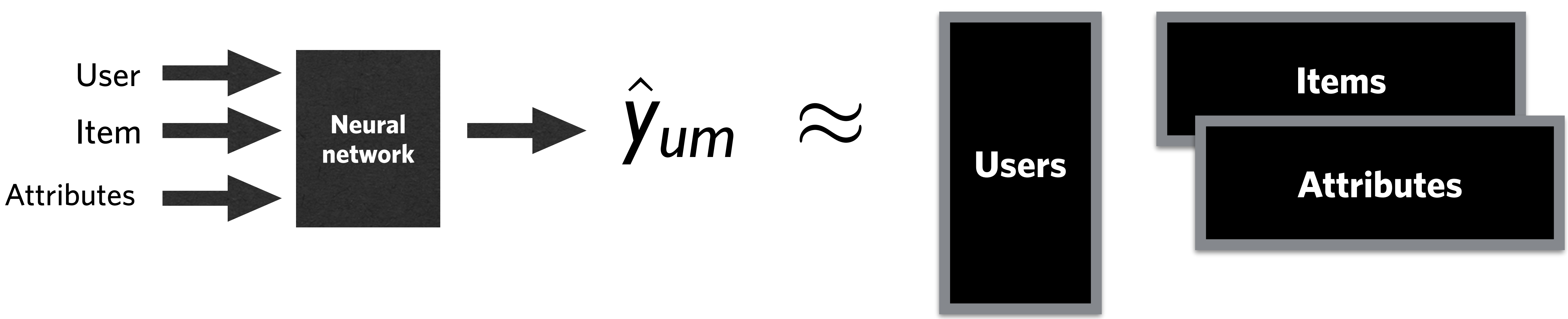
$$p(y_{um} = 1 \mid u, m) = \sigma\left(f(u, x_m)\right)$$

Universal approximation: RankFromSets can approximate any order-invariant model.

$$f(u, x_m)$$

Examples of models invariant to permutation of item attributes:

- Matrix factorization
- Permutation-marginalized recurrent neural networks

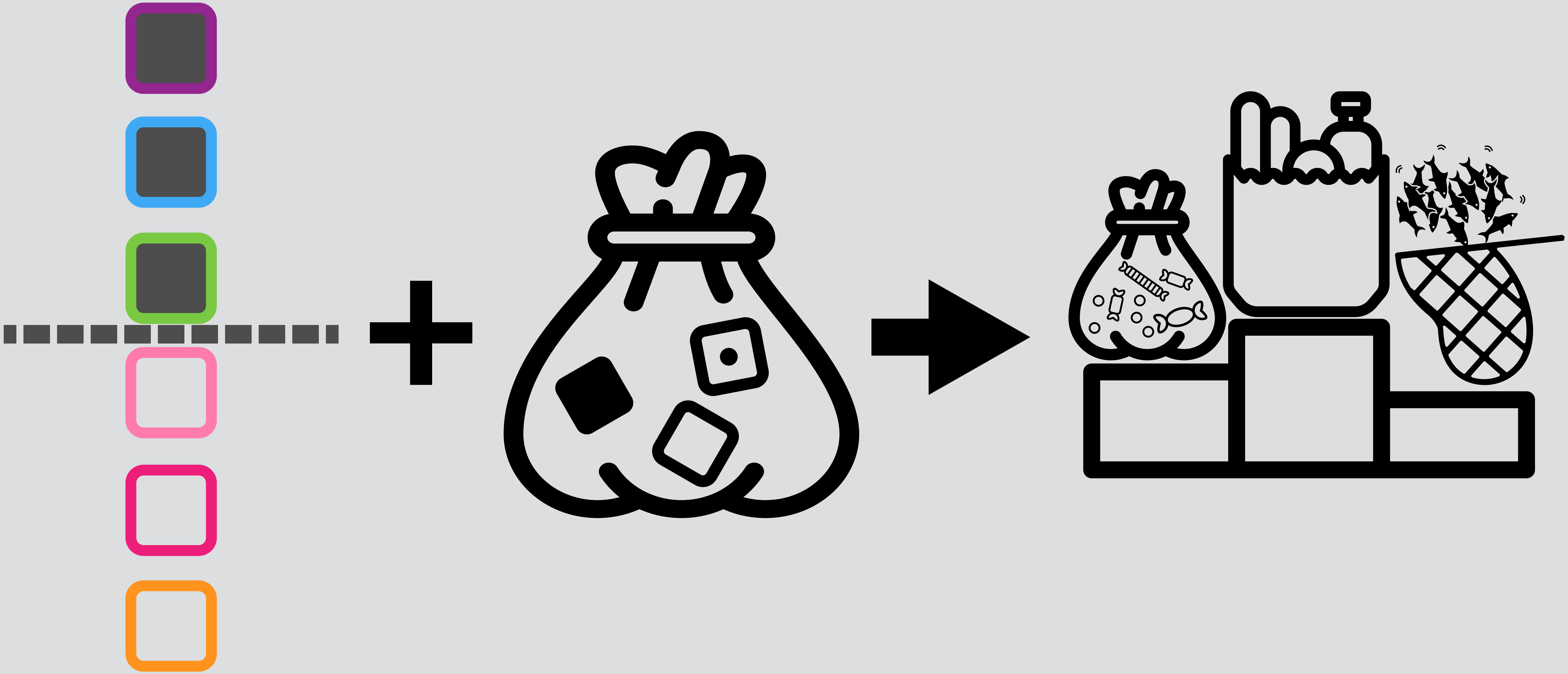


[Zaheer+ 2017; Altosaar+ 2020]

RankFromSets Objective

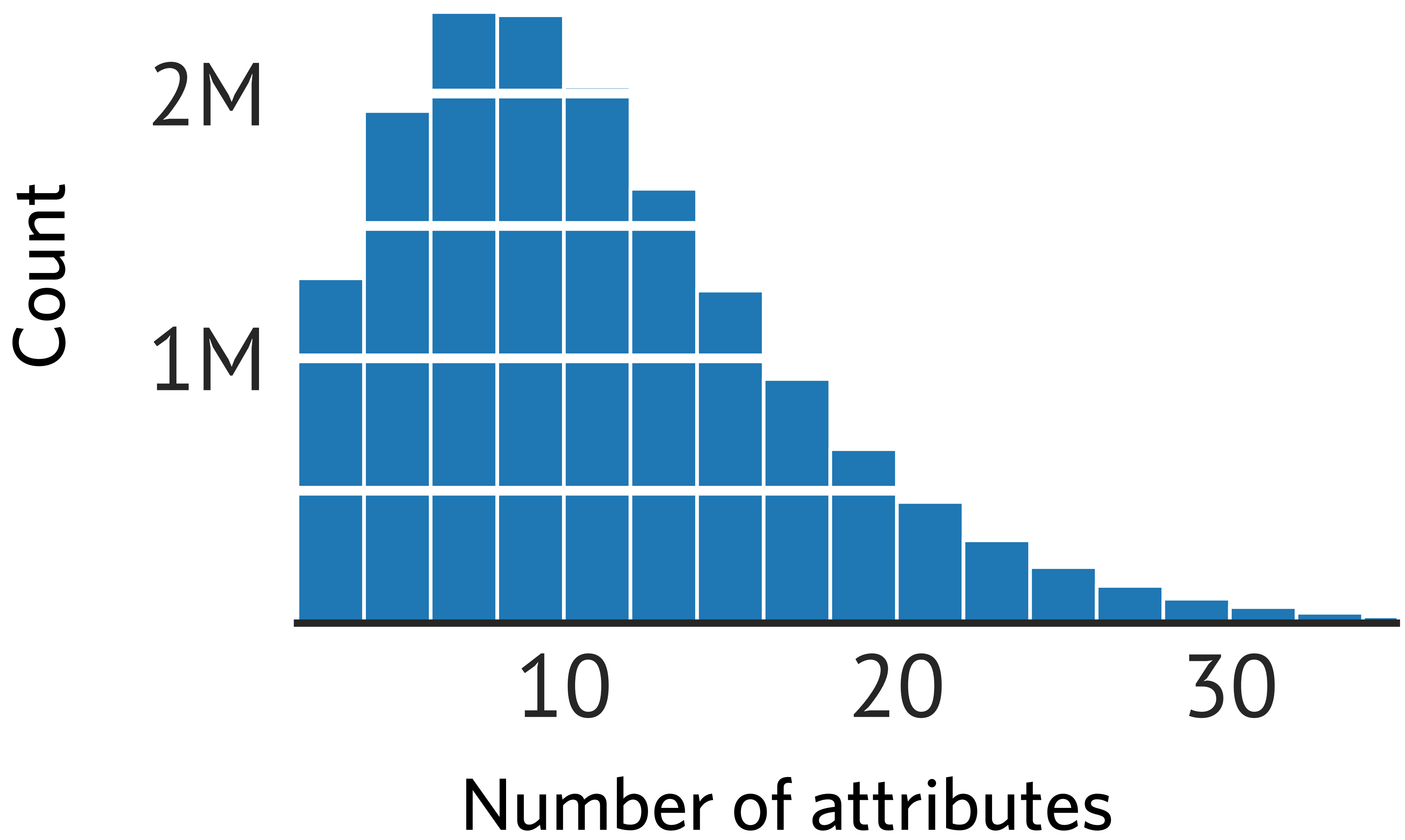
$$\mathcal{L}(\gamma, \lambda_u) = \mathbb{E}_u \Big[\mathbb{E}_{m \sim \mathcal{D}_u \mid y_{um}=1} \left[\log p(y_{um} = 1 \mid x_m; \gamma) \right] \\ + \lambda_u \mathbb{E}_{k \sim \mathcal{D}_u \mid y_{uk}=0} \left[\log p(y_{uk} = 0 \mid x_k; \gamma) \right] \Big]$$

- Parameters γ: user embeddings, attribute embeddings, item embeddings, weights and biases
- λ_u balances negative examples for every user




99¢ ONLY
99 Thanks..!
99 ONLY
Rolo
Reese's

Neilson
crème à fouetter
toujours fraîche
173 mL
Nutrition Facts
Valeur nutritive
DUCK BREAST

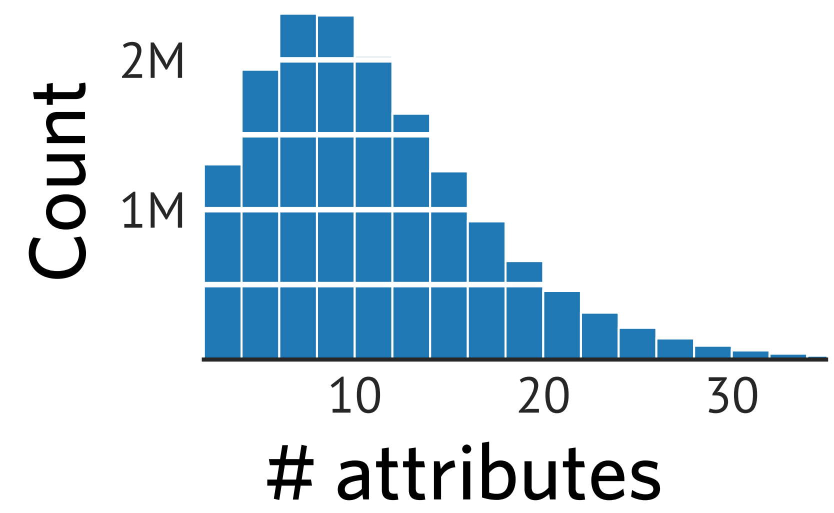


Count
2M
1M
10
20
30
Number of attributes

Lose It!

One year of data from 55,000 users of the Lose It! food tracking app

- 15 million meals
- Item attributes: 9,963 food words

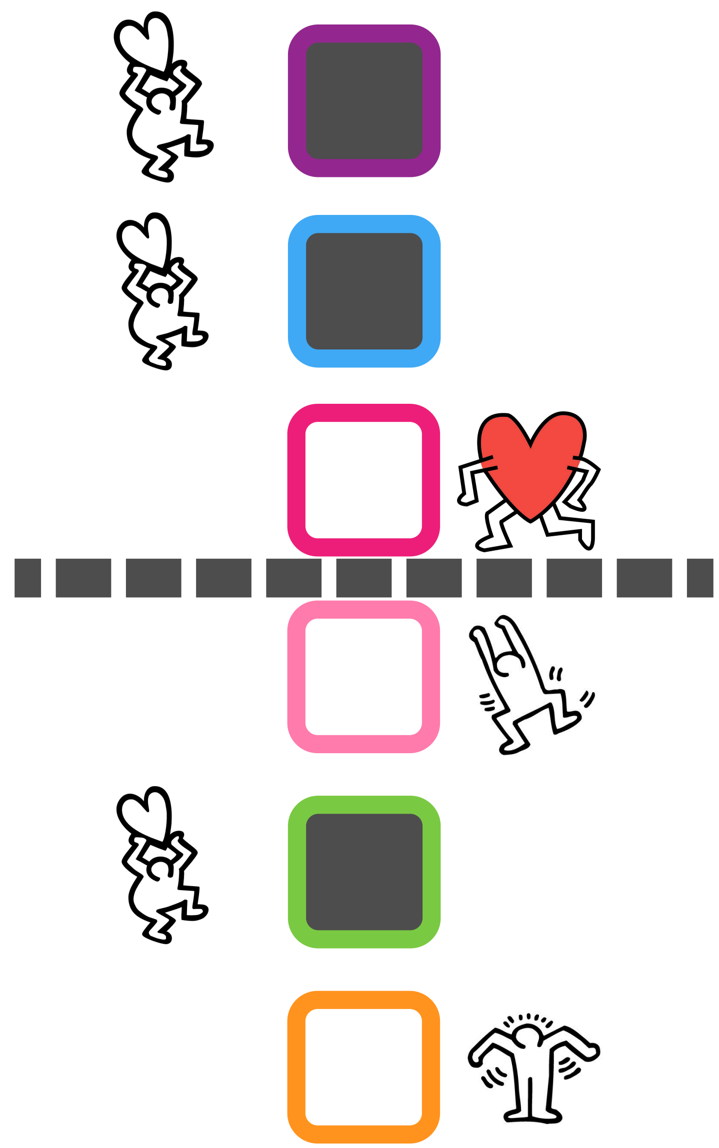


Example meals:

- *Two scoops of Raisin Bran cereal, organic Moroccan green tea, almond milk, light honey, tap water, large banana, large strawberries*
- *Boston roast pork, mackerel, artichoke hearts, spinach, pimiento-stuffed Manzanilla olives, carrots, mushrooms, peppercorn ranch dressing*

Scalable Evaluation

For every held out item, sample negative labels from other users.



[Ebesu+ 2018; Yang+ 2018]

Lose It!

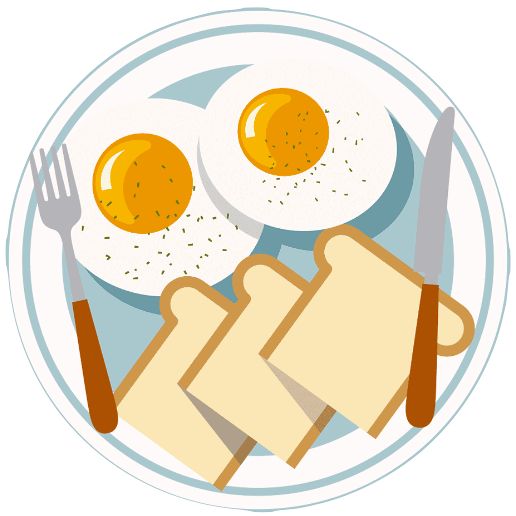
RFS f: residual
RFS f: inner product
CTPF
Word embedding
StarSpace
LSTM
Random

- CTPF: Collaborative Topic Poisson Factorization [Gopalan+ 2014]
- StarSpace [Wu+ 2018]
- LSTM [Bansal+ 2016]
- Word embedding [Bojanowski+ 2016]

Query

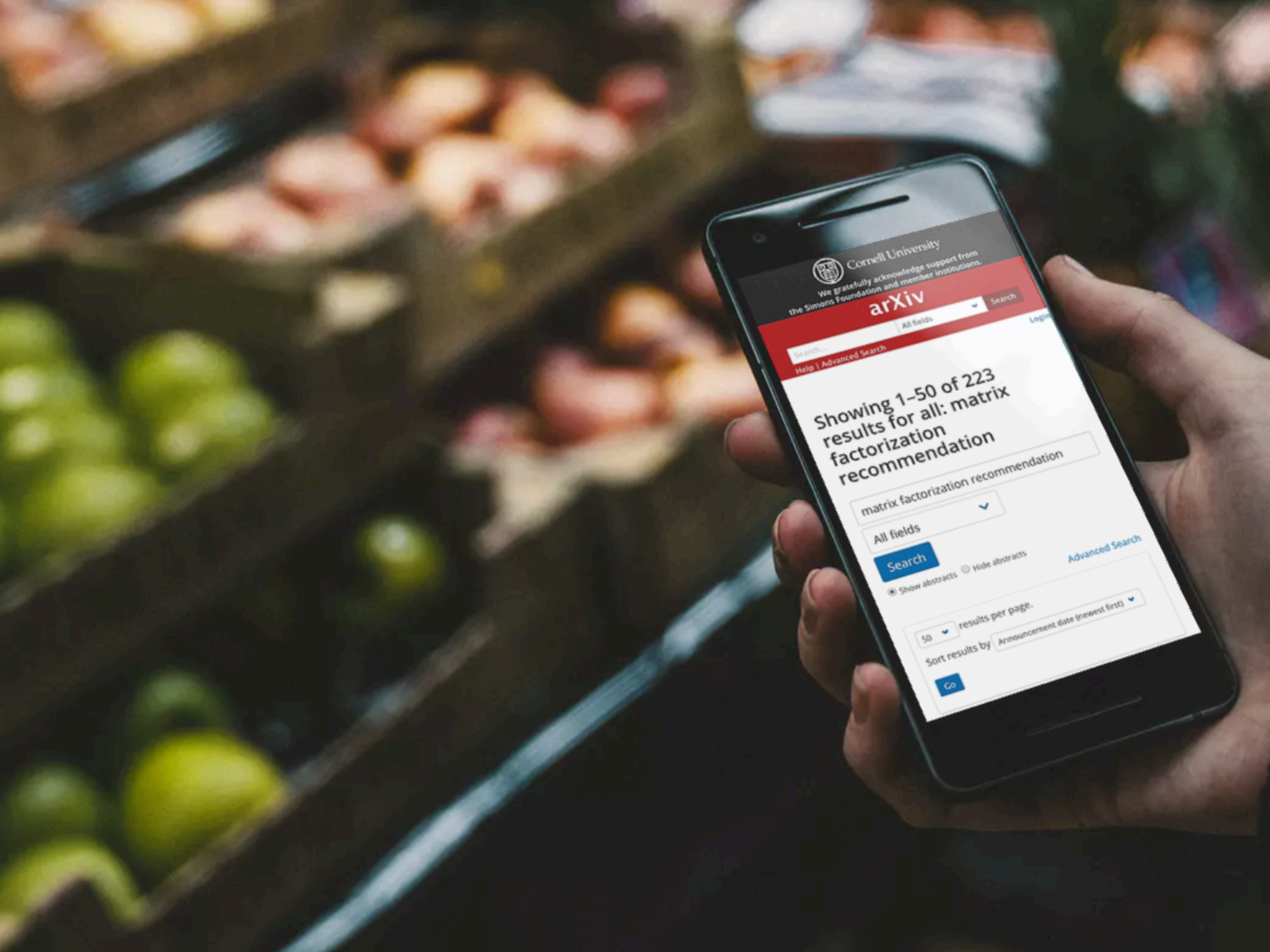
Two scoops of Raisin Bran cereal, organic Moroccan green tea, almond milk, light honey, tap water, large banana, large strawberries

Nearest Neighbor



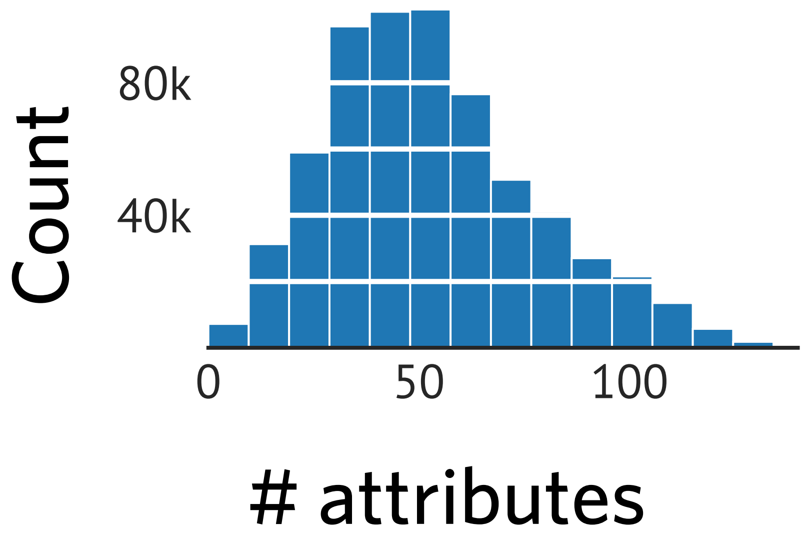
Vita Bee bread, salted butter, fresh medium tomatoes, large fried whole egg, small banana

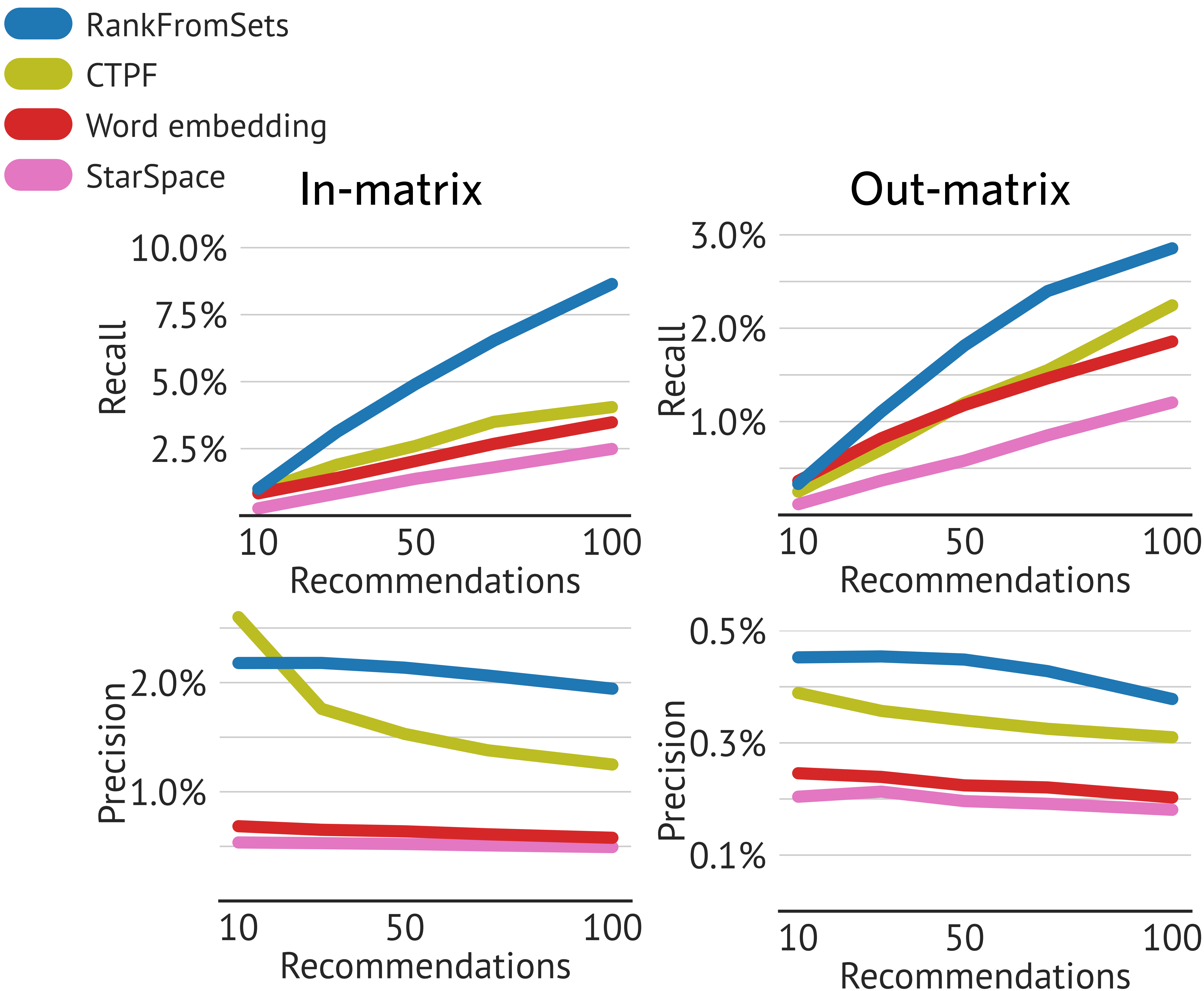
Query	Nearest Neighbor
	
Iceberg lettuce, cantaloupe cubes, diced honeydew melon, cherry tomatoes, olives, dry-cooked unsalted hulled sunflower seed kernels, chopped hard-boiled egg, cucumbers, dried cranberries, fat-free ranch dressing	Green leaf lettuce, chopped sweet red bell peppers, crumbled feta cheese, large hard-boiled egg, chopped cucumber, oil-roasted salted sunflower seeds, sliced radishes, sliced strawberries, pitted Calamata olives, fat-free balsamic vinegar


Cornell University
We gratefully acknowledge support from
the Simons Foundation and member institutions.
arXiv
All fields
Search
Login
Search...
Help | Advanced Search
Showing 1–50 of 223 results for all: matrix factorization recommendation
matrix factorization recommendation
All fields
Search
Show abstracts
Hide abstracts
Advanced Search
50
results per page.
Sort results by
Announcement date (newest first)
Go

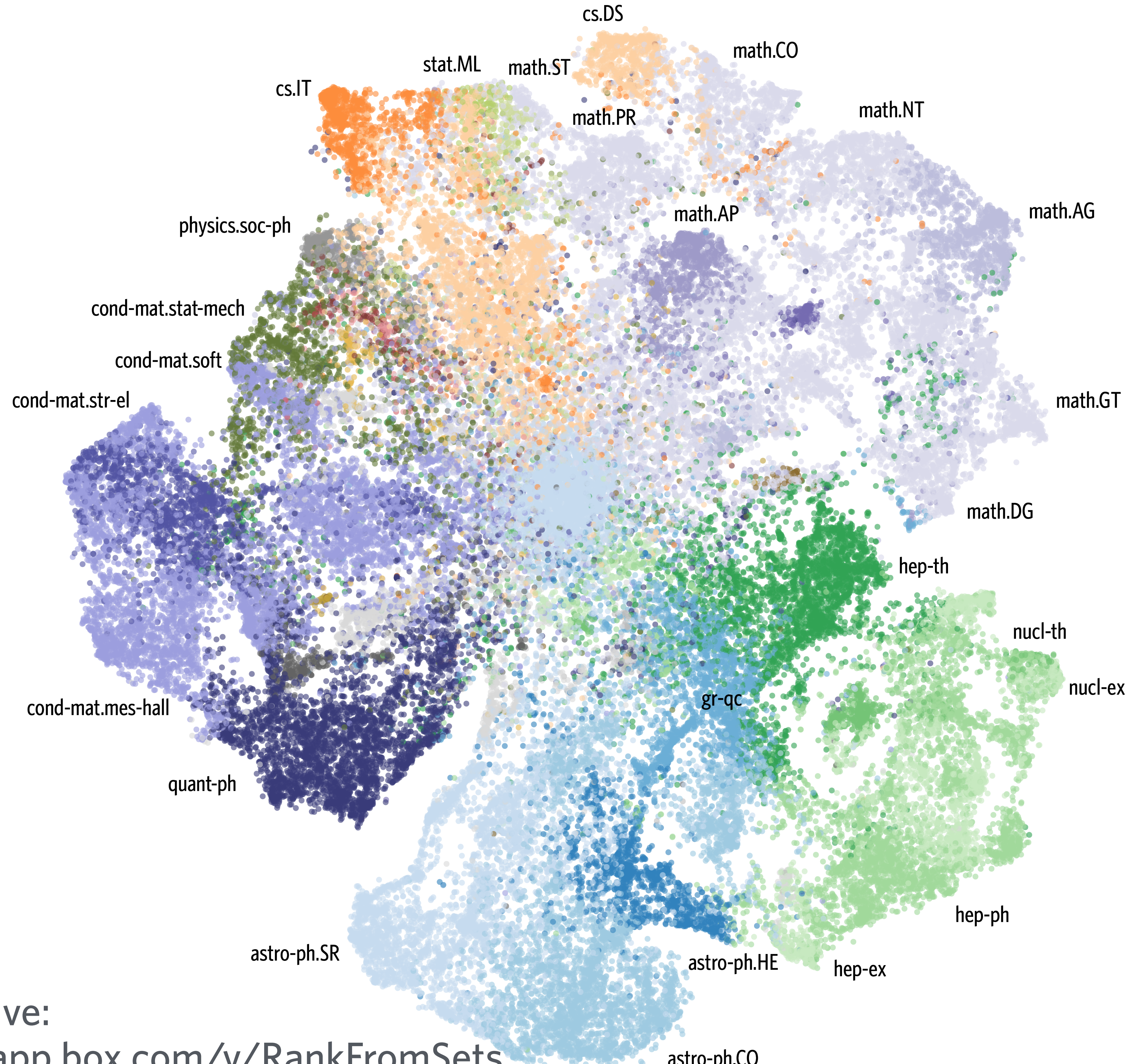
One year of data, 64,978 users

- 636,622 papers
- Vocabulary of 14,000 in abstracts

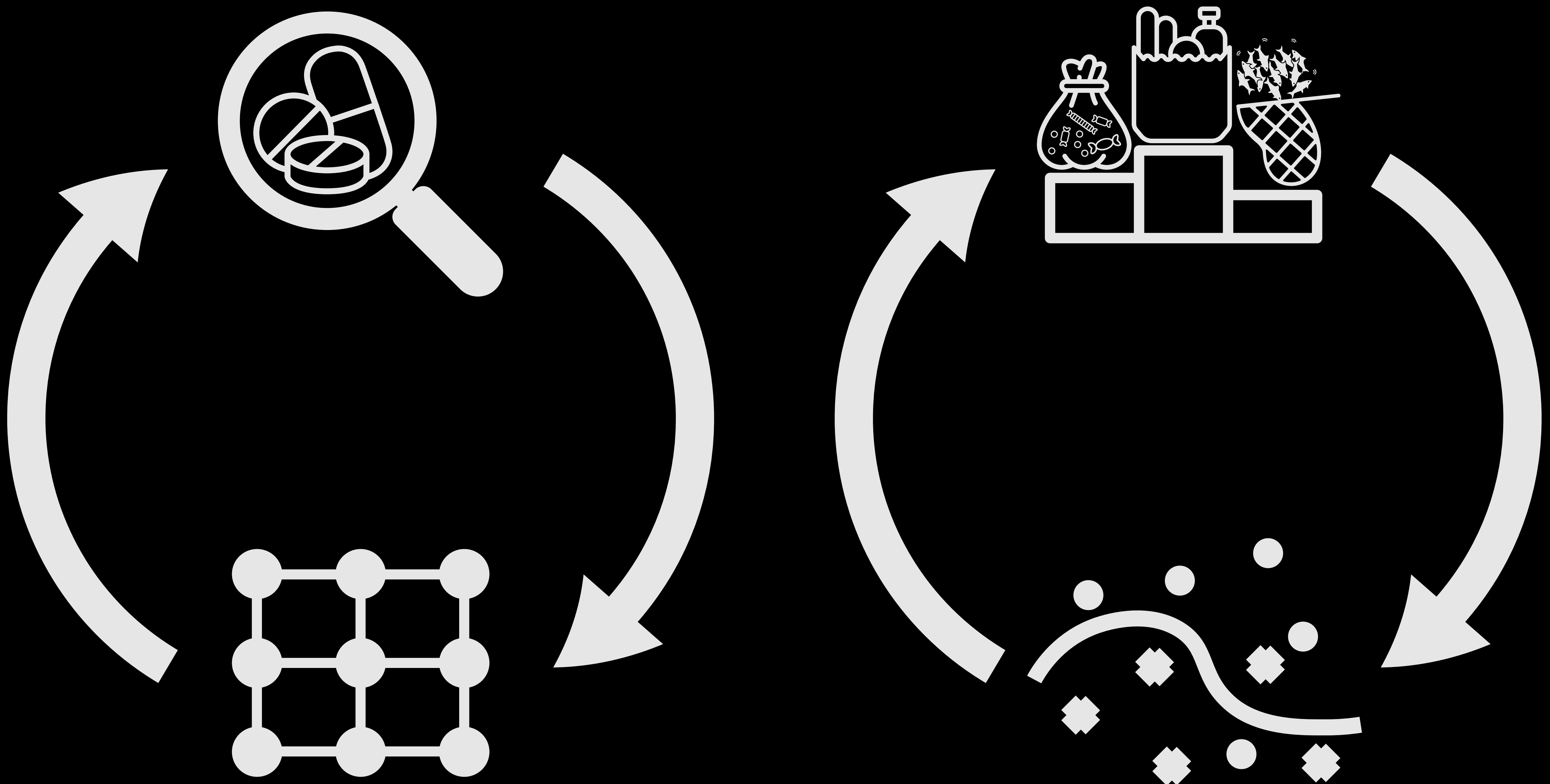


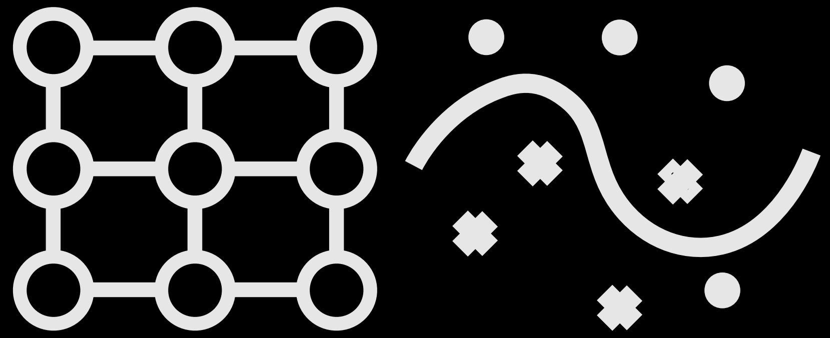


RankFromSets
CTPF
Word embedding
StarSpace
In-matrix
Out-matrix
Recall
10.0%
7.5%
5.0%
2.5%
10
50
100
Recommendations
Recall
3.0%
2.0%
1.0%
10
50
100
Recommendations
Precision
2.0%
1.0%
10
50
100
Recommendations
Precision
0.5%
0.3%
0.1%
10
50
100
Recommendations



Interactive:
https://app.box.com/v/RankFromSets



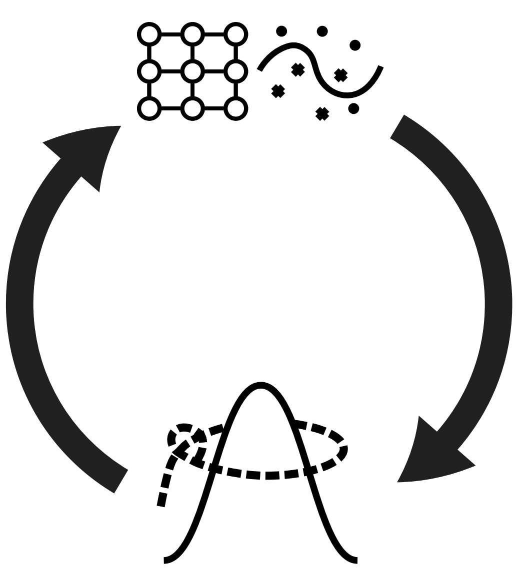


Bernoulli Factor Model

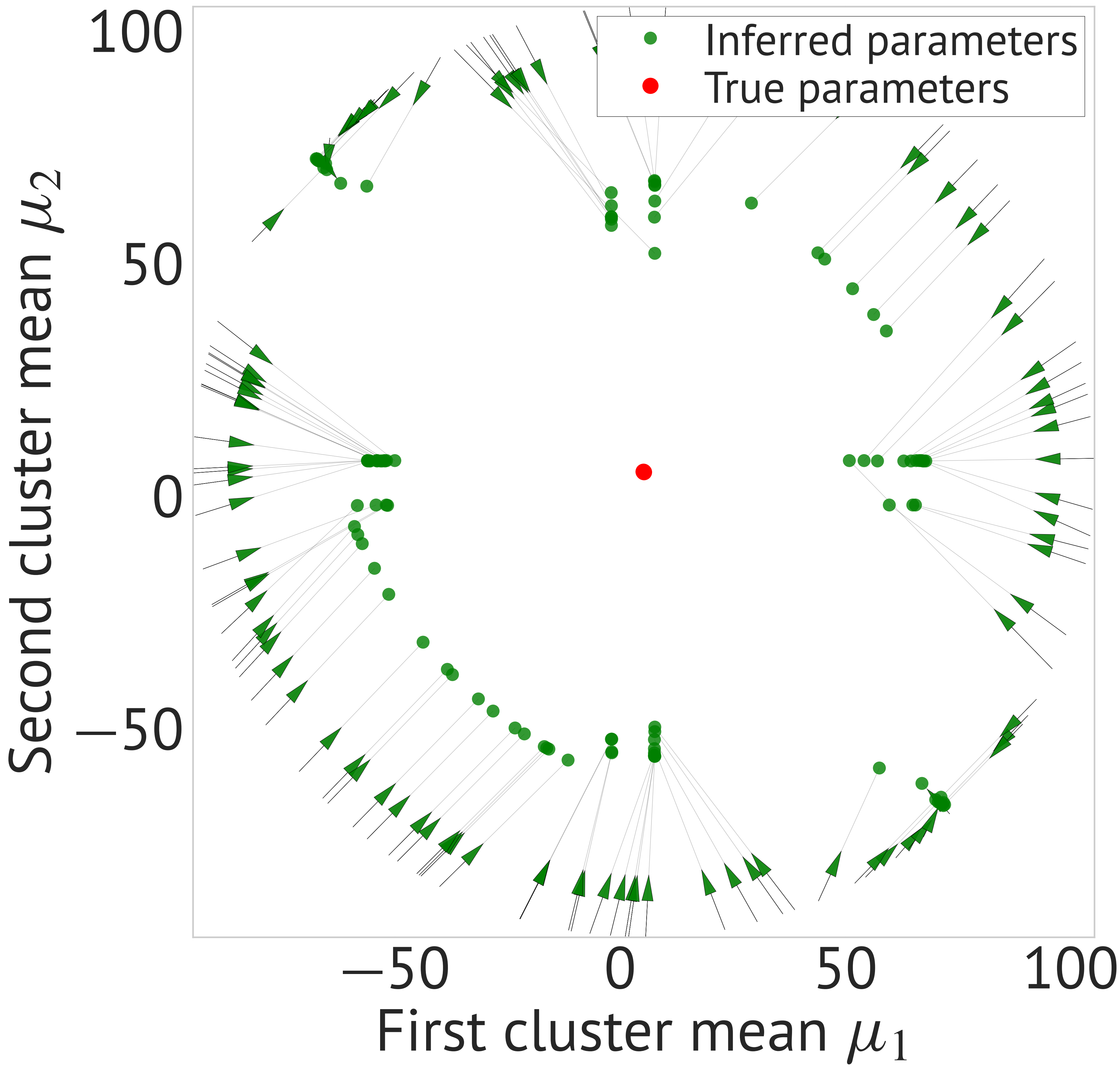
$$z_{ik} \sim \mathrm{Bernoulli}(\pi)$$

$$x_i \sim \mathrm{Normal}(\mathbf{z}_i^\top \mu, \sigma^2 = 1)$$

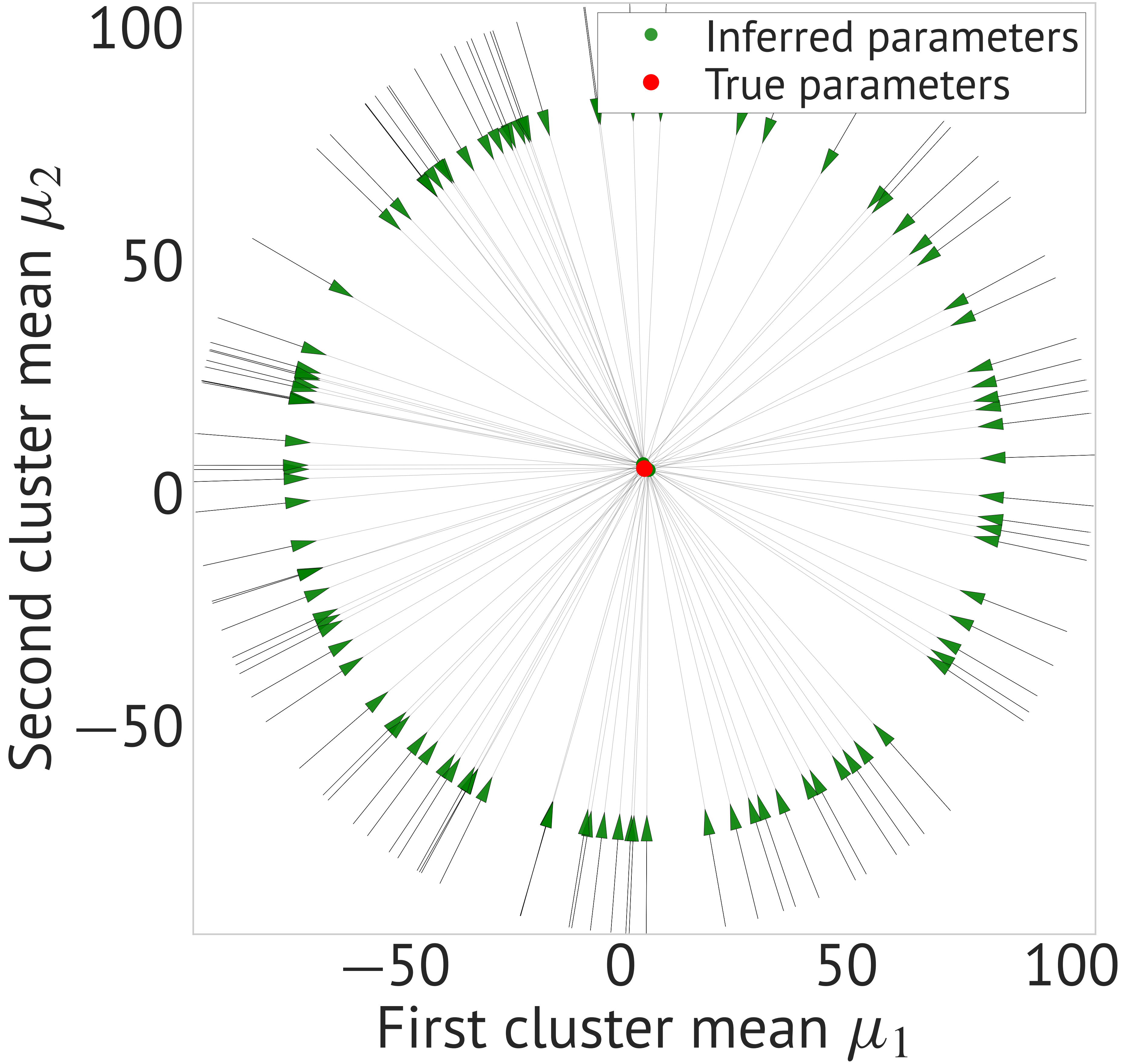
Goal: infer cluster mean parameter



Variational Inference

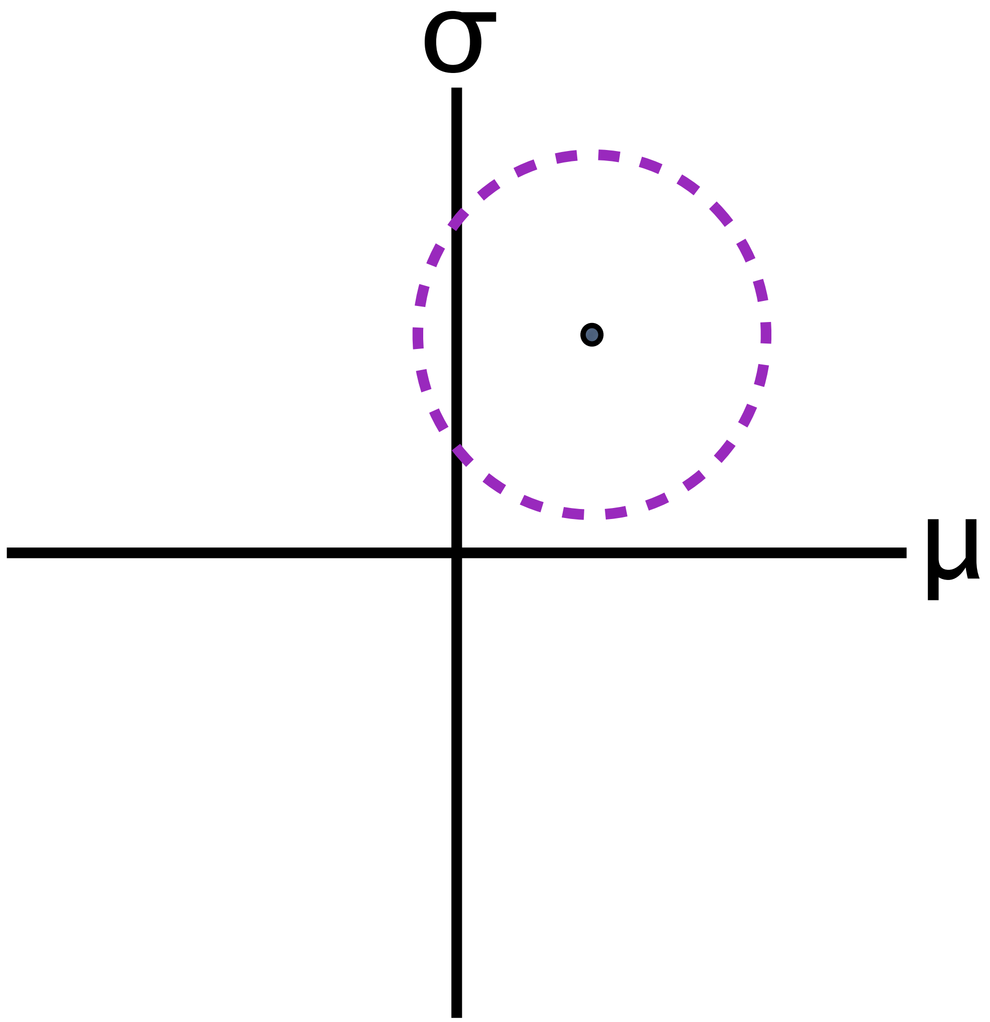


Proximity Variational Inference



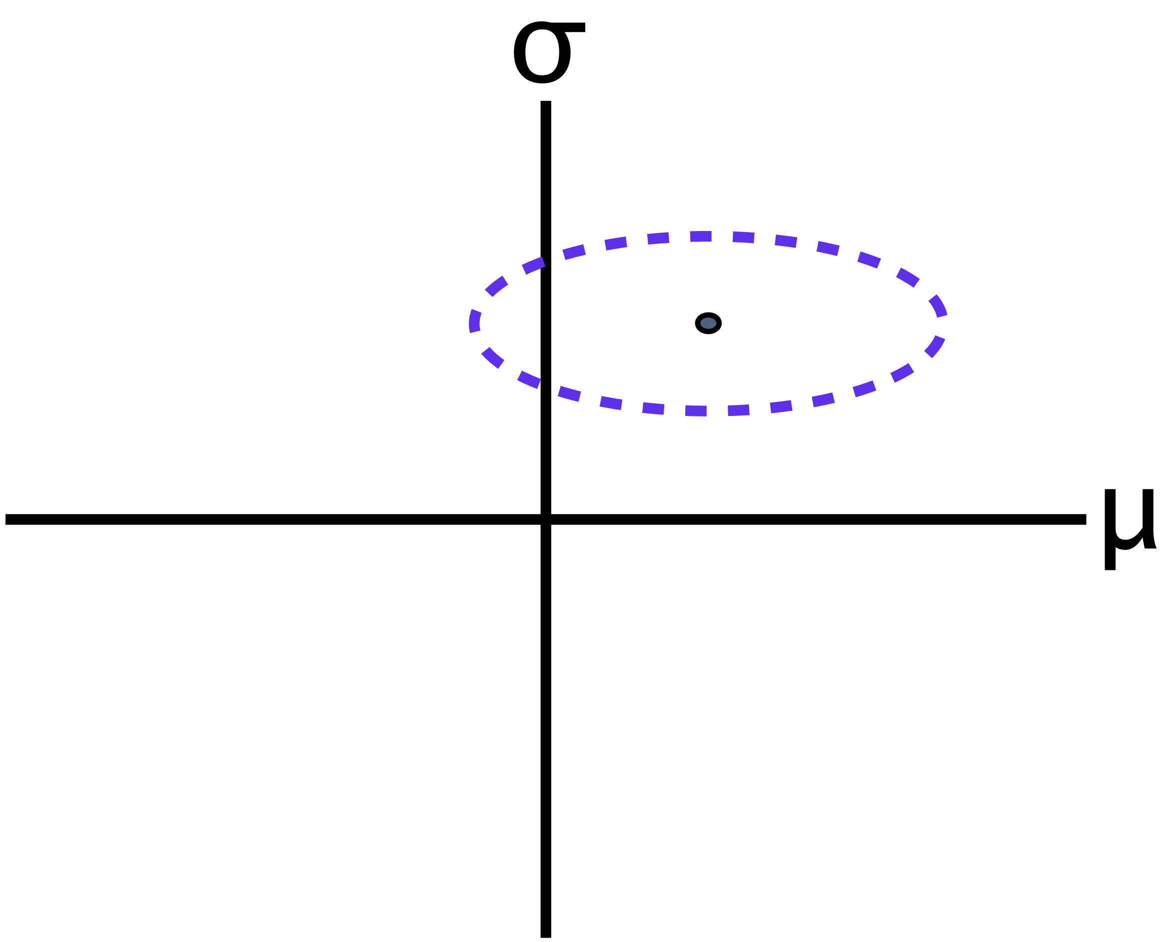
Gradient Ascent: Euclidean Proximity Constraint

$$U(\boldsymbol{\lambda}_{t+1}) = \mathcal{L}(\boldsymbol{\lambda}_t)$$



Proximity Variational Inference

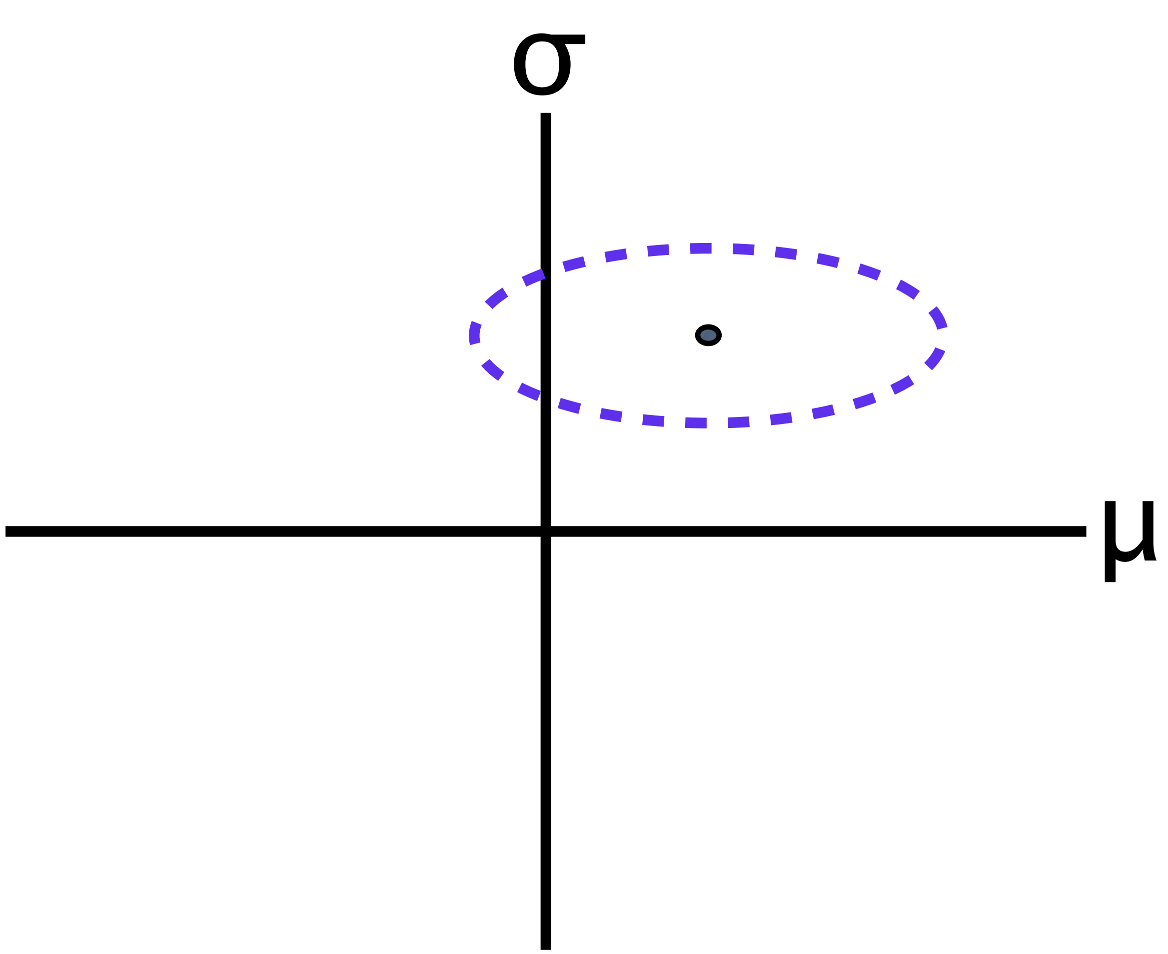
$$U(\boldsymbol{\lambda}_{t+1}) = \mathcal{L}(\boldsymbol{\lambda}_t) + \nabla\mathcal{L}(\boldsymbol{\lambda}_t)^\top(\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t) - \frac{1}{2\rho}(\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t)^\top(\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t)$$

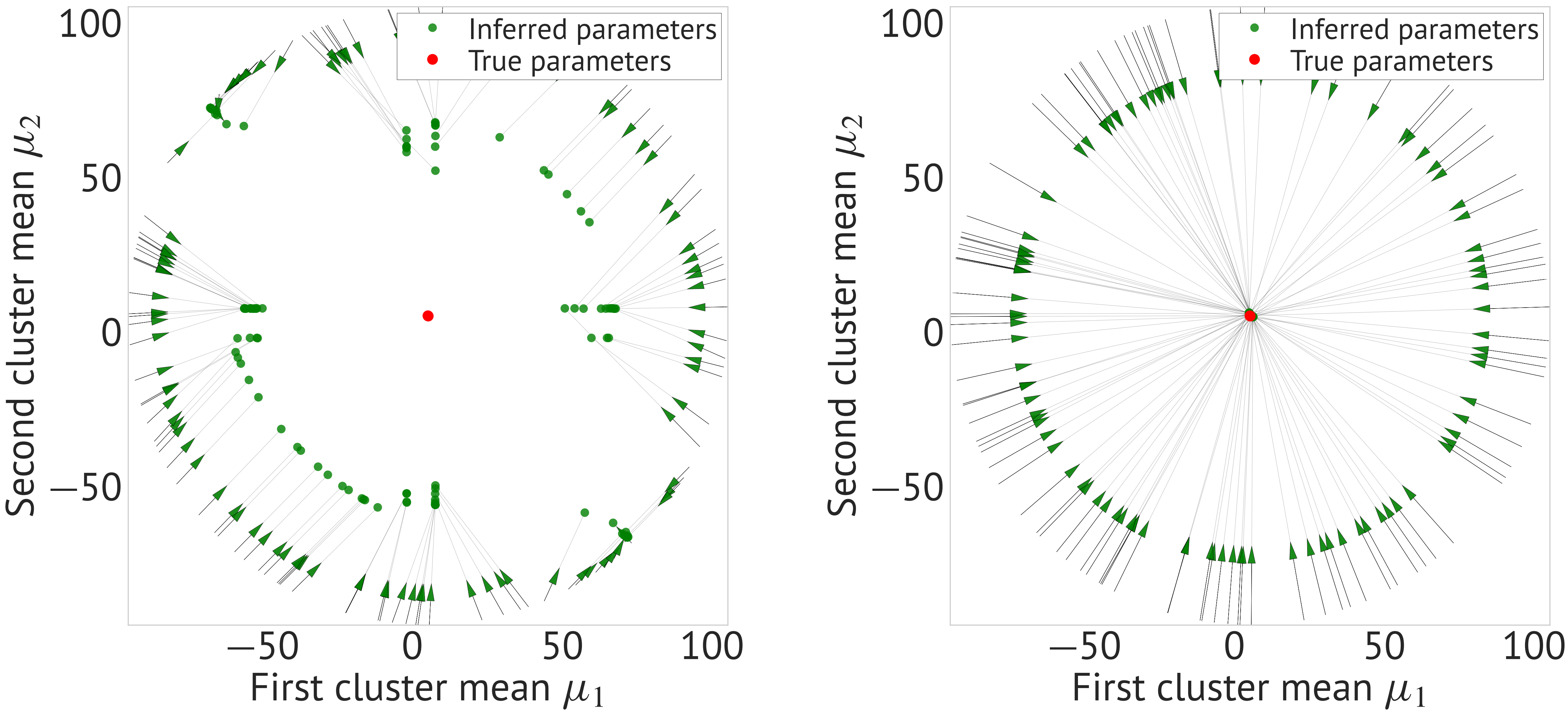


Proximity Variational Inference

Examples of proximity statistics $f(\boldsymbol{\lambda})$:

- Mean/Variance
- Entropy
- KL divergence
- Many more!





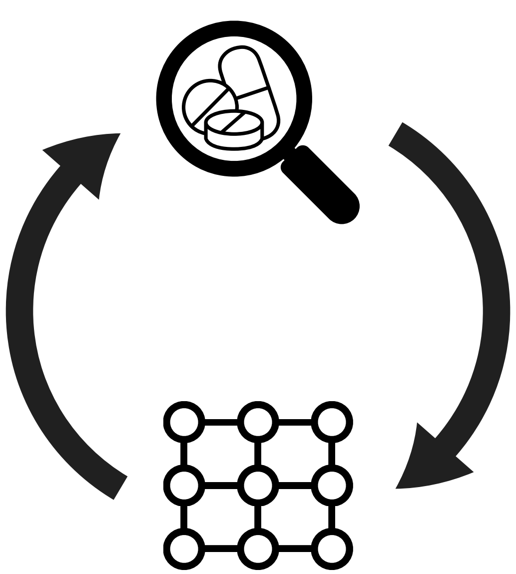
Inferred parameters
True parameters
100
50
0
−50
Second cluster mean μ_2
−50
0
50
100
First cluster mean μ_1
Inferred parameters
True parameters
100
50
0
−50
Second cluster mean μ_2
−50
0
50
100
First cluster mean μ_1

Sigmoid Belief Network

Inference Method	ELBO	Likelihood
Variational Inference	−121.4	−113.7
Deterministic Annealing	−116.8	−108.8
PVI, Entropy Constraint	**−113.3**	**−106.7**
PVI, Mean/Variance Constraint	−114.9	−107.4

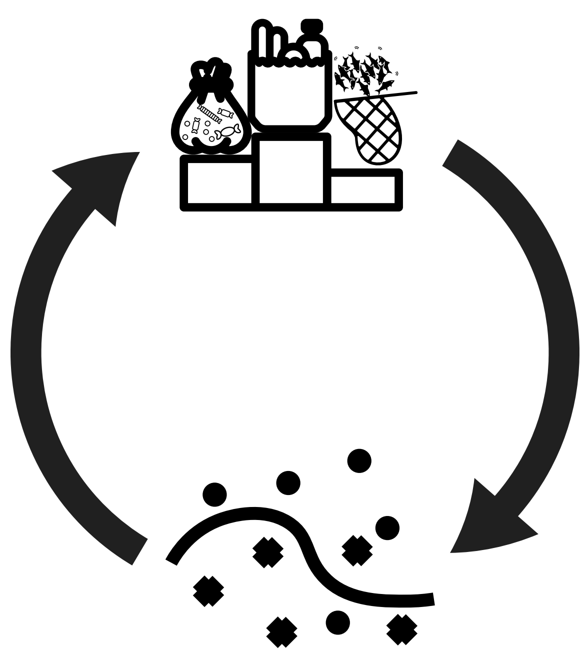
Ising Model

Inference Method	Free Energy
Variational Inference	−2.144
PVI, Entropy Constraint	**−2.158**



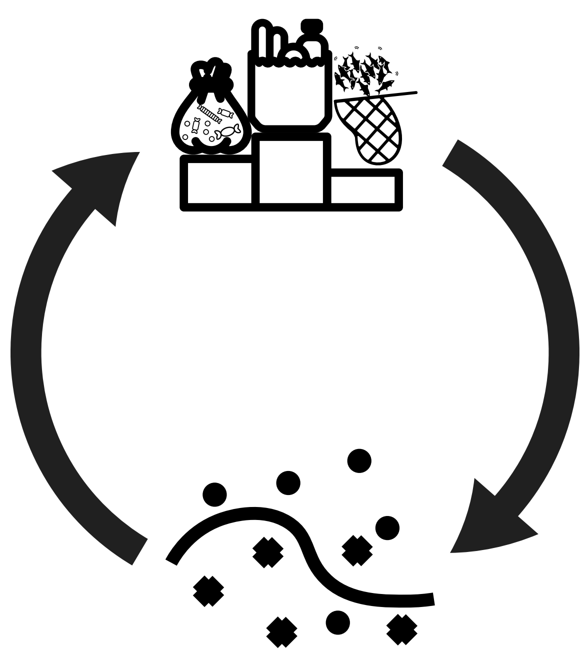
RankFromSets:
Meal Recommendation

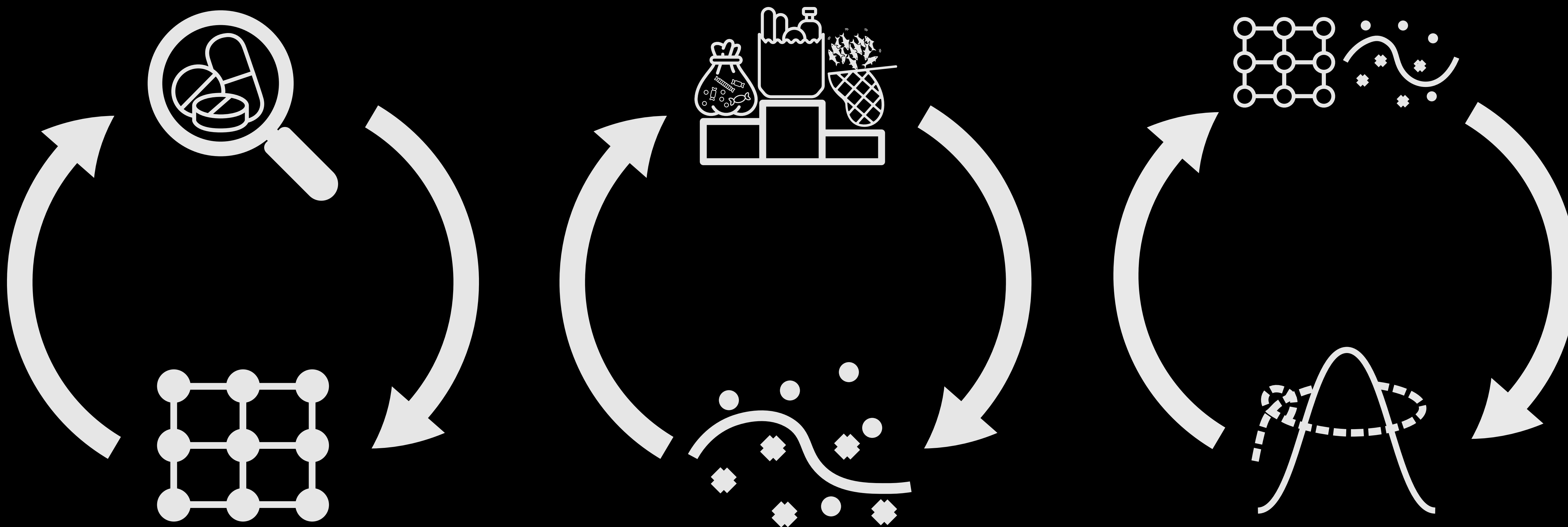
Model	Sampled Recall (%)
RFS	58
RFS, Entropy Constraint	**62**

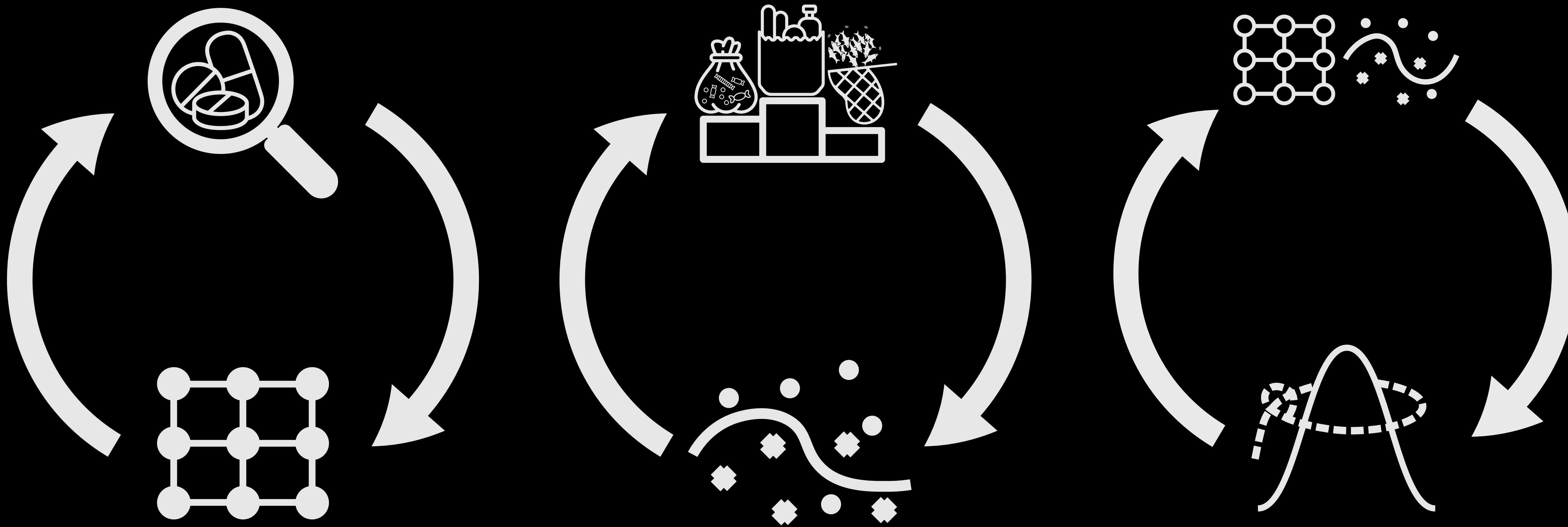


RankFromSets: ArXiv Recommendation

Model	Recall @ 10 (%)	Recall @ 100 (%)
RFS	0.32	2.54
RFS, Entropy Constraint	**0.44**	2.54







- J. Altosaar, R. Ranganath, and K. Cranmer. Hierarchical variational models for statistical physics. *Machine Learning and the Physical Sciences Workshop, Neural Information Processing Systems*. 2019

- J. Altosaar, R. Ranganath, W. Tansey. RankFromSets: Scalable Set Recommendation with Optimal Recall. *American Statistical Association SDSS 2020.*

- Altosaar, Jaan, Rajesh Ranganath, and David Blei (2018). “Proximity Variational Inference”. *Artificial Intelligence and Statistics.*

- *Design Credits:* Sergey Demushkin, Creaticca Creative Agency, Made by Made, Gan Khoon Lay, myiconfinder, Susannanova, Walmart, Raisin Bran, PNGFuel, Alibaba Group, Keith Haring