Active Embedding Search via Noisy Paired Comparisons

Gregory H. Canal

Georgia Institute of Technology

In collaboration with:

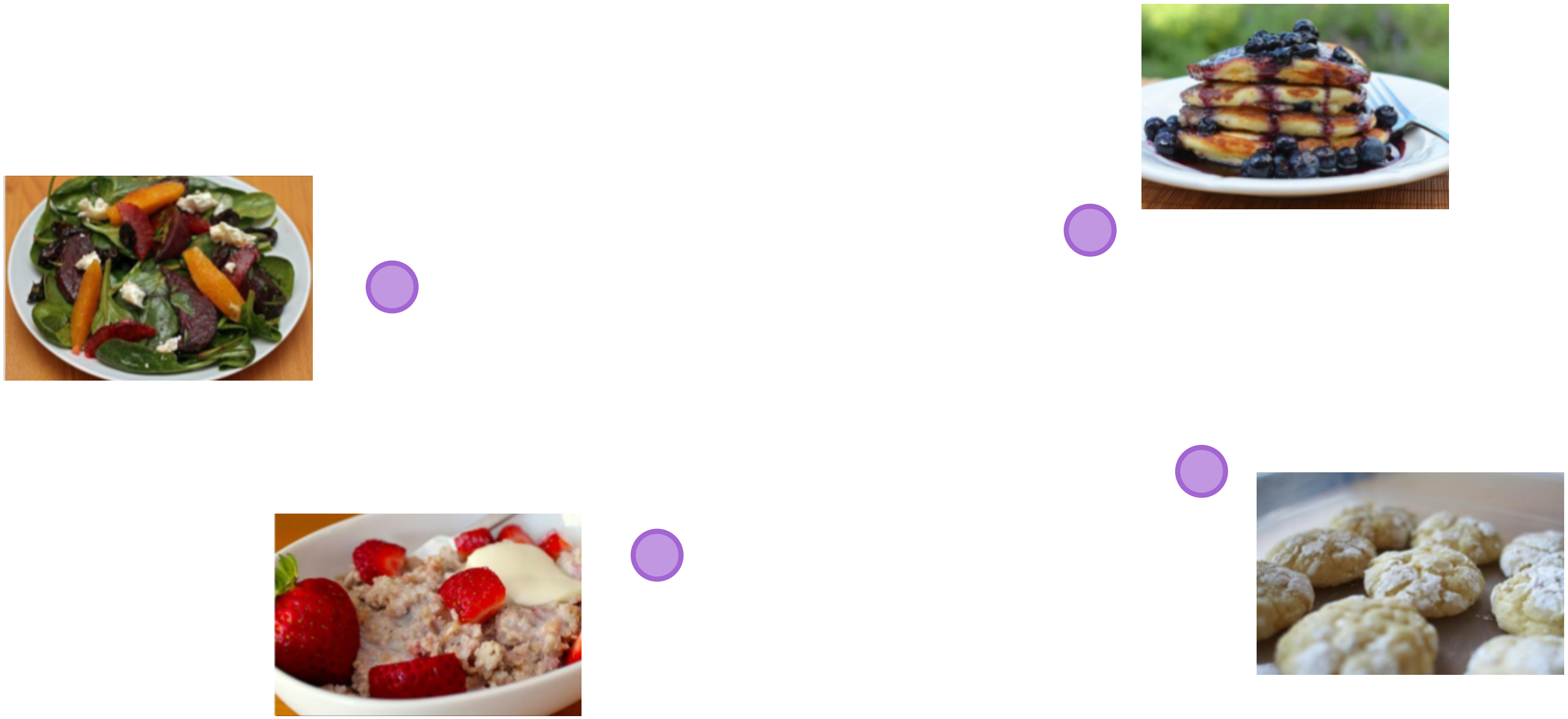
Andrew K. Massimino, Mark A. Davenport, Christopher J. Rozell

Center for Signal & Information Processing

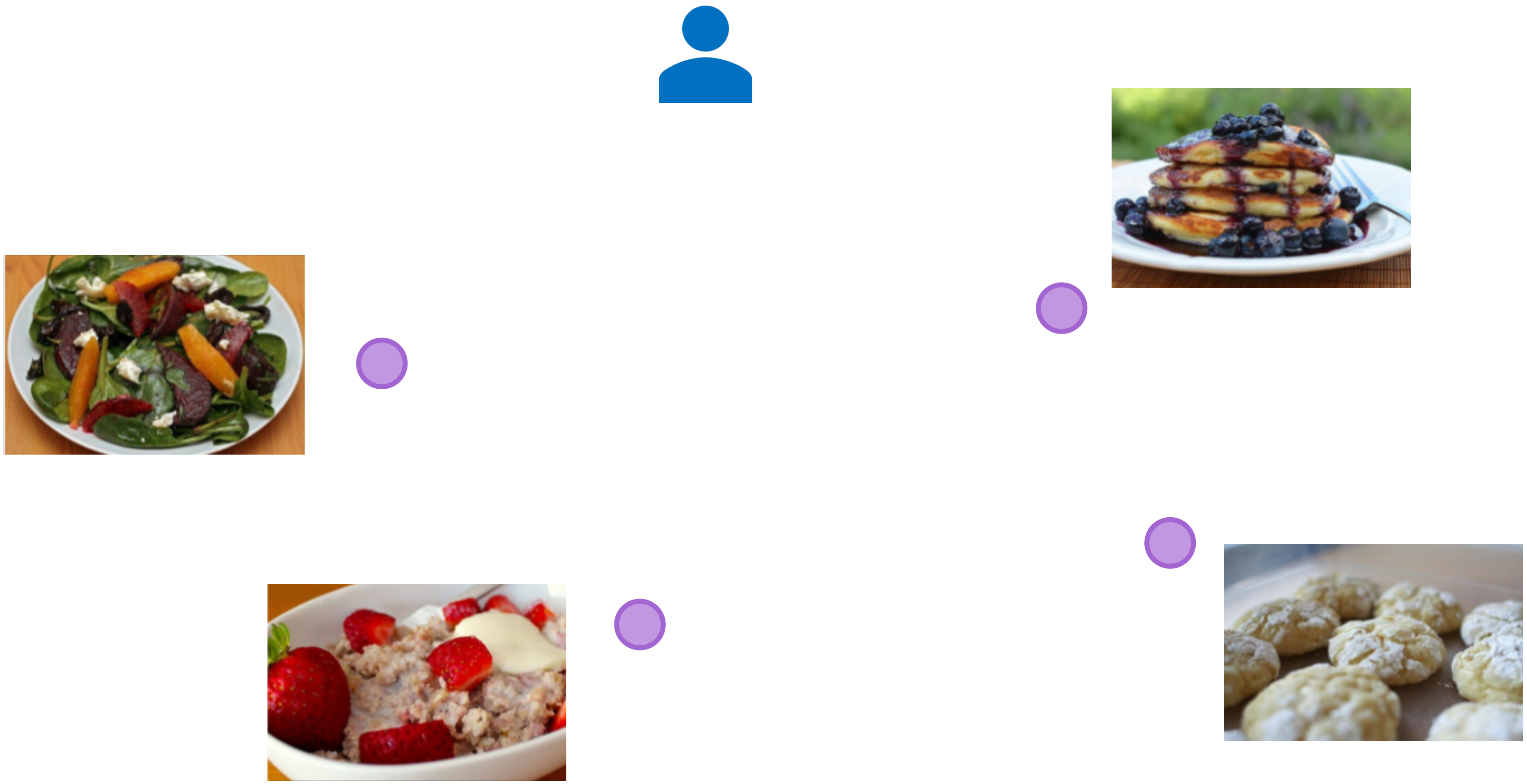
Sensory Information Processing Lab

Estimating preferences in similarity embedding

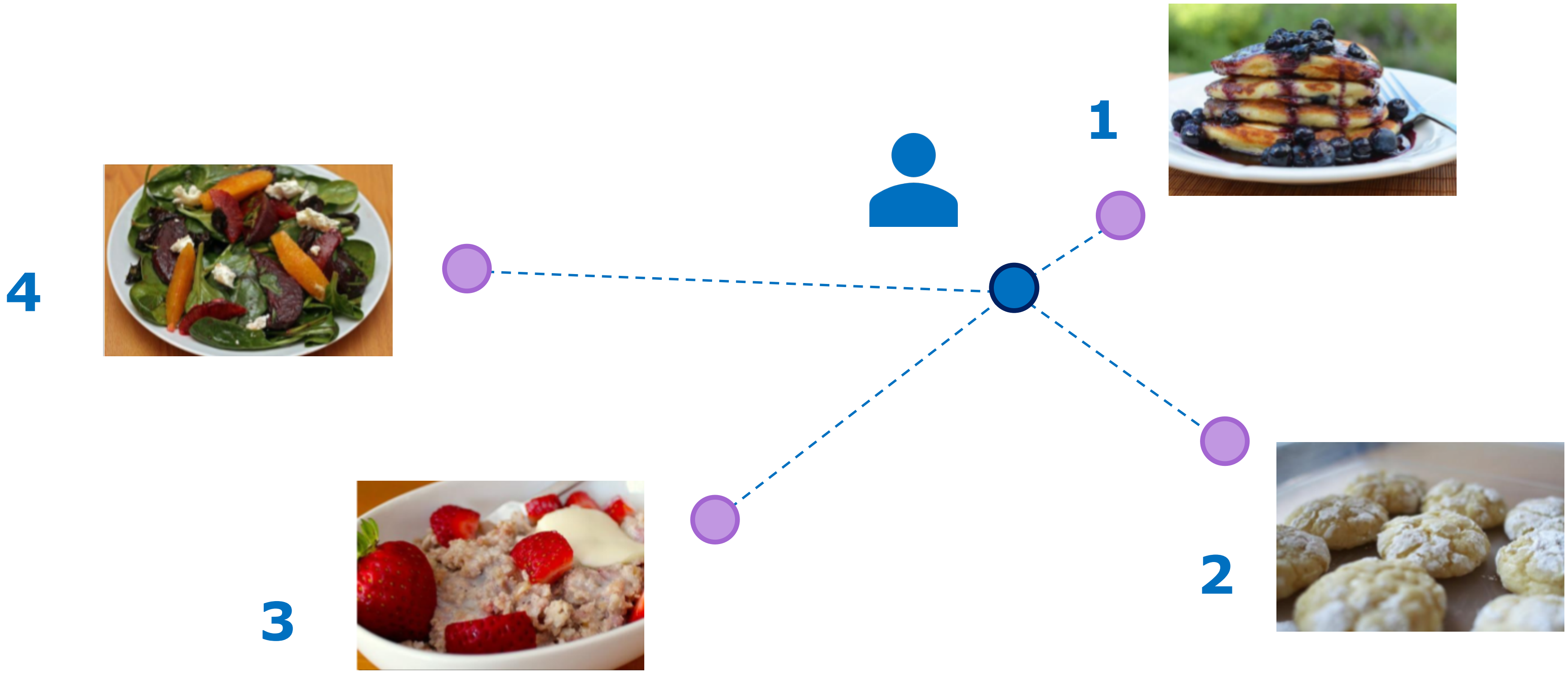
Estimating preferences in similarity embedding



Estimating preferences in similarity embedding

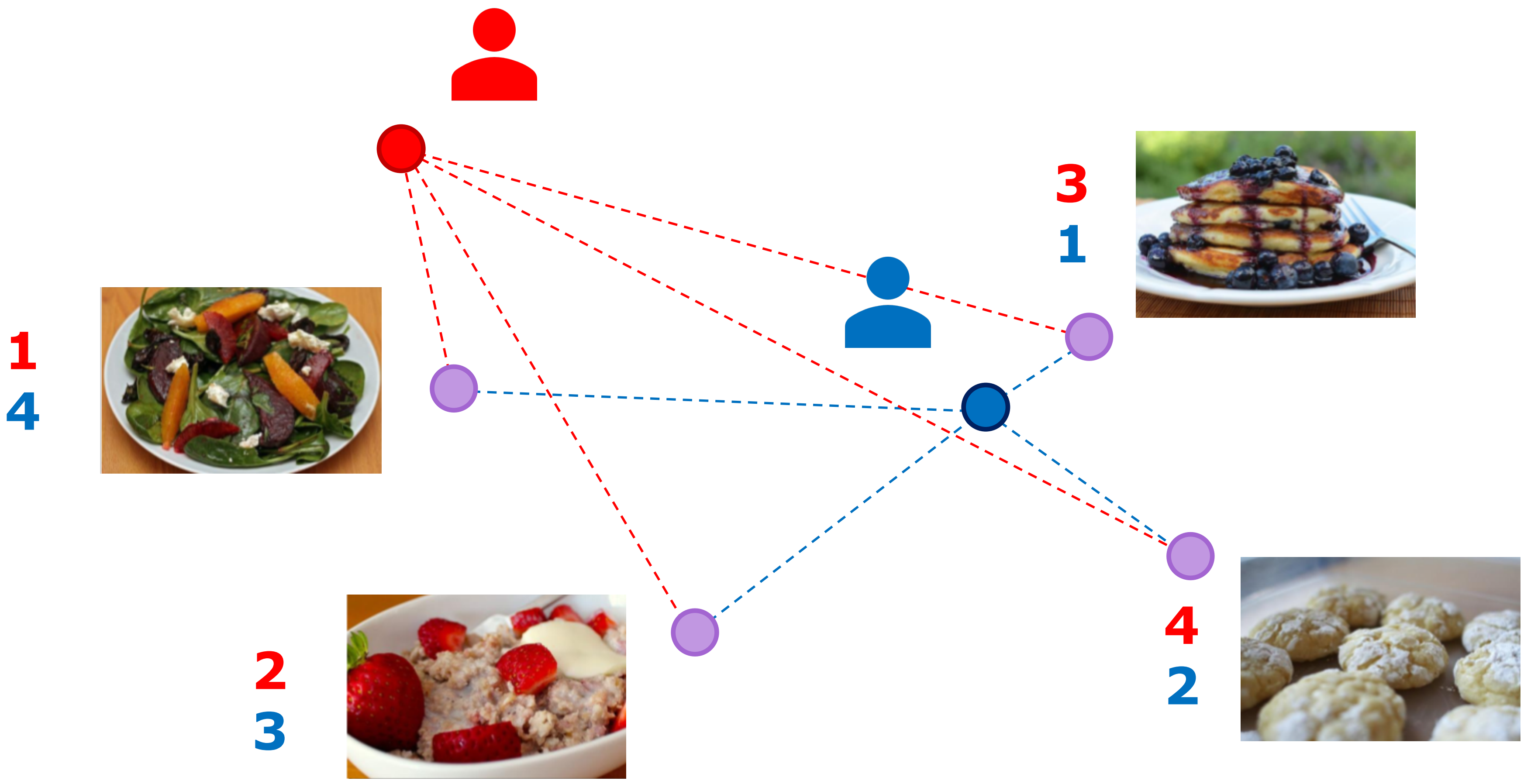


Estimating preferences in similarity embedding



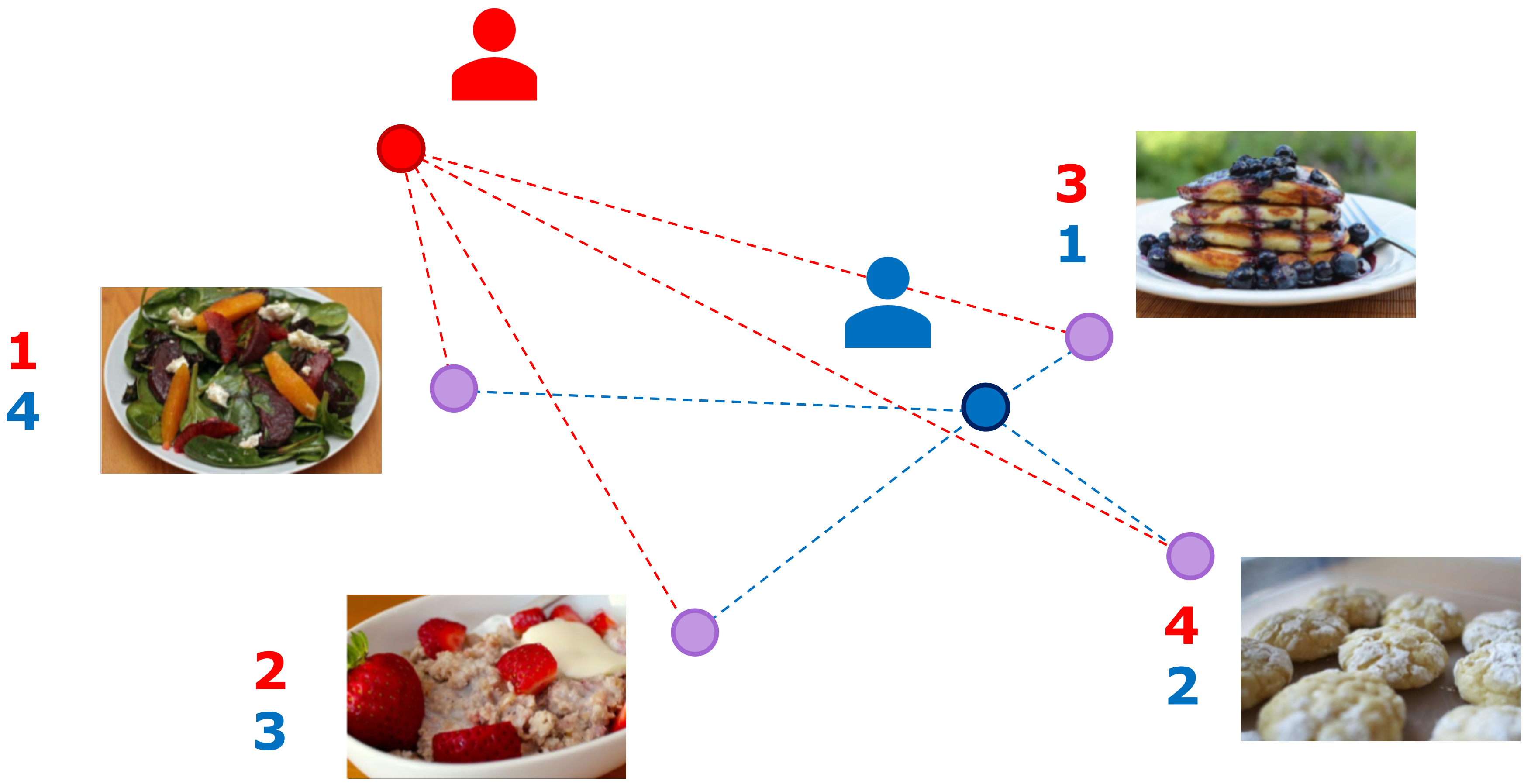
- Item preferences ranked by distance to user

Estimating preferences in similarity embedding



- Item preferences ranked by distance to user

Estimating preferences in similarity embedding

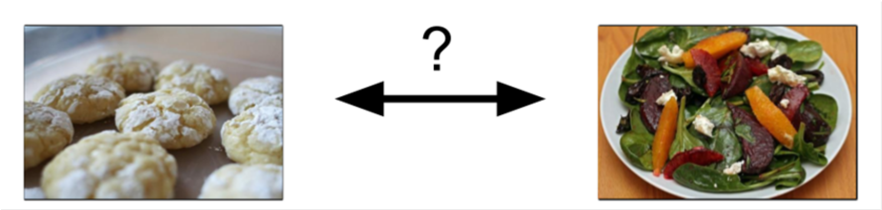


- Item preferences ranked by distance to user
- Continuous user point: hypothetical *ideal* item (not necessarily in dataset)

Method of paired comparisons

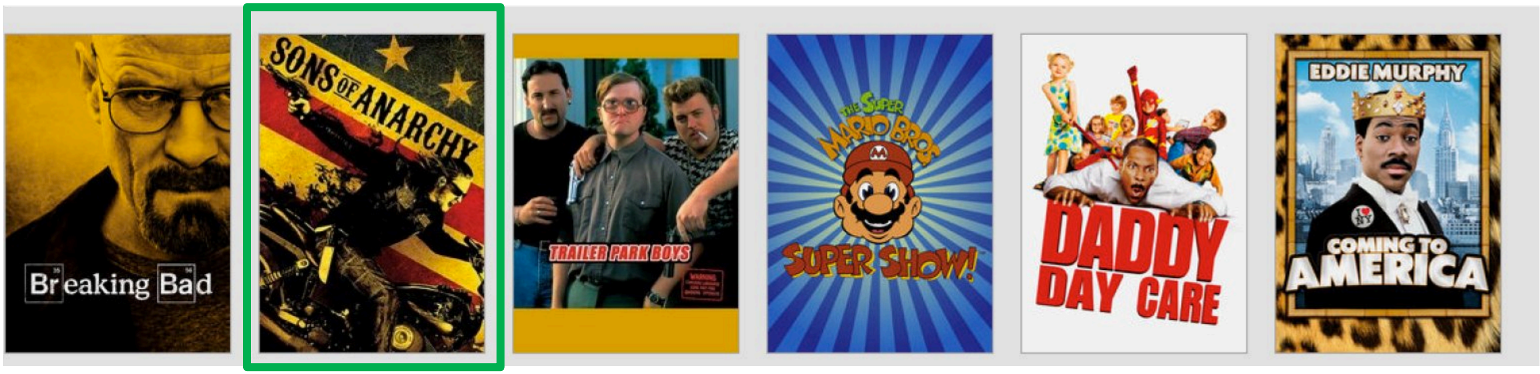
Learn preferences via *method of paired comparisons* (David, 1963)

"Which of these two foods do you prefer to eat?"



– Direct comparisons may be explicitly solicited

– Comparisons are *implicitly* solicited everywhere

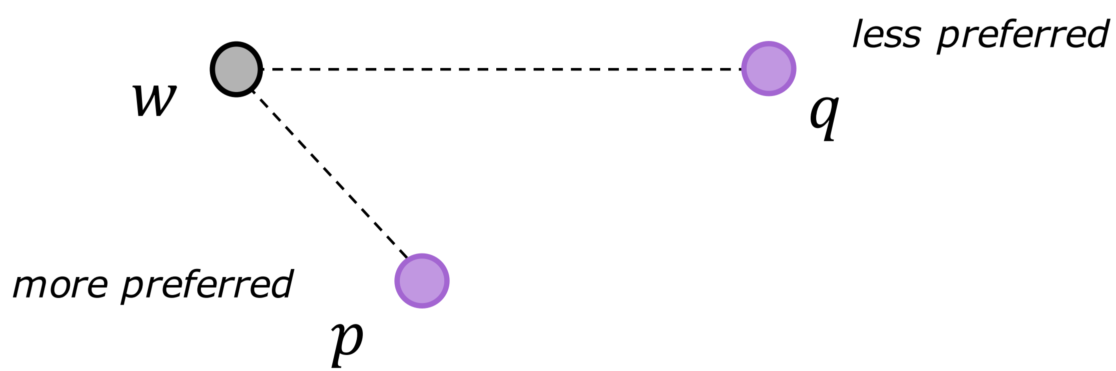


– In practice, responses are noisy, inconsistent

Ideal point model

- *Pairwise search*: estimate user vector $w \in \mathbb{R}^d$ based on paired comparisons between items
- *Ideal point model*: continuous point w encodes ideal item that is preferred over all other items (Coombs, 1950)

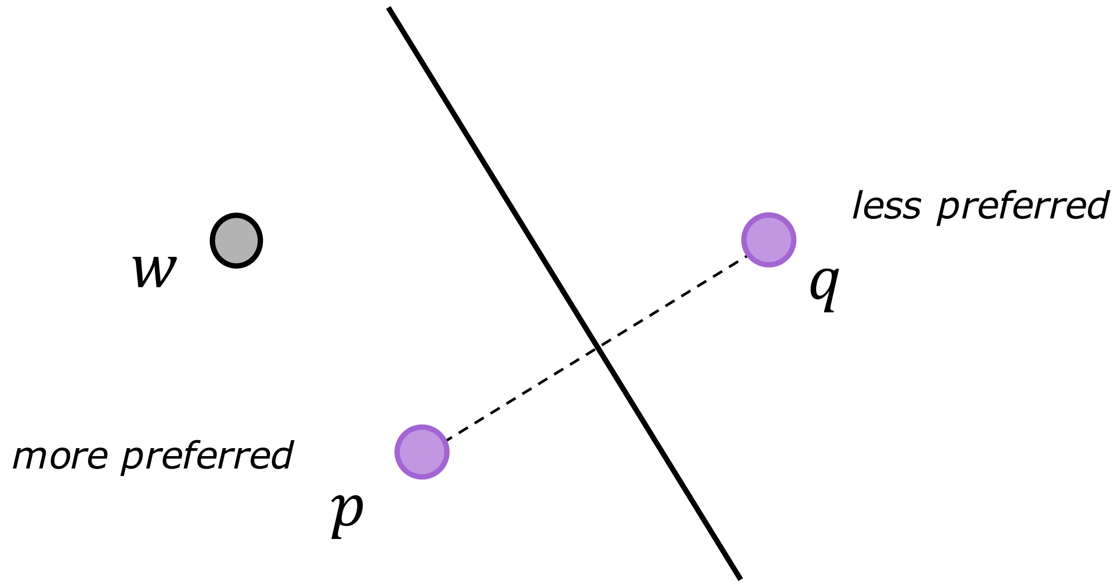
Paired comparison (p, q): user at w prefers item p over item q if and only if $\|w - p\| < \|w - q\|$



Ideal point model

- *Pairwise search*: estimate user vector $w \in \mathbb{R}^d$ based on paired comparisons between items
- *Ideal point model*: continuous point w encodes ideal item that is preferred over all other items (Coombs, 1950)

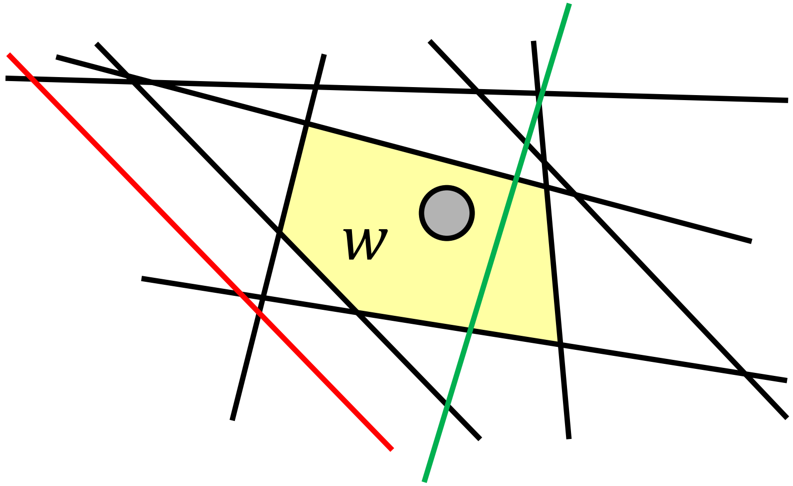
Paired comparison (p, q): user at w prefers item p over item q if and only if $\|w - p\| < \|w - q\|$



Prior work

How can paired comparisons (hyperplanes) be selected?

- Query as few pairs as possible
- Linear models (e.g., learning to rank, latent factors) unsuitable for nonlinear ideal point model (Wu et al., 2017; Qian et al., 2015)
- Feasible region tracking
  - Query pairs adaptively
  - Add slack variables to feasible region (Massimino & Davenport, 2018)
  - Repeat comparisons, take majority vote (Jamieson & Nowak, 2011)
  - Previous methods ***do not*** incorporate noise into pair selection

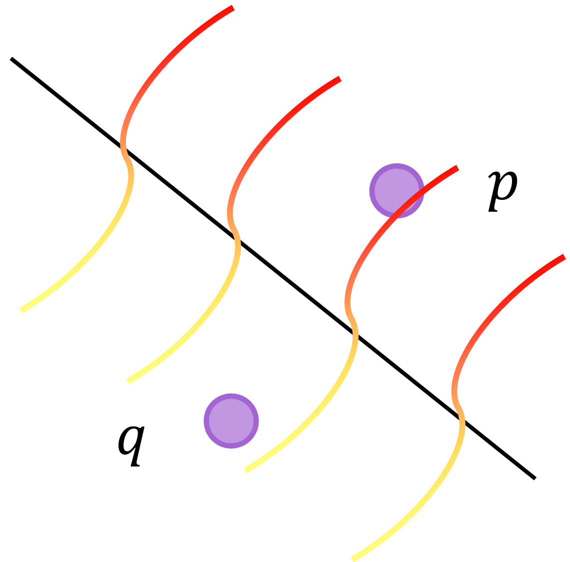


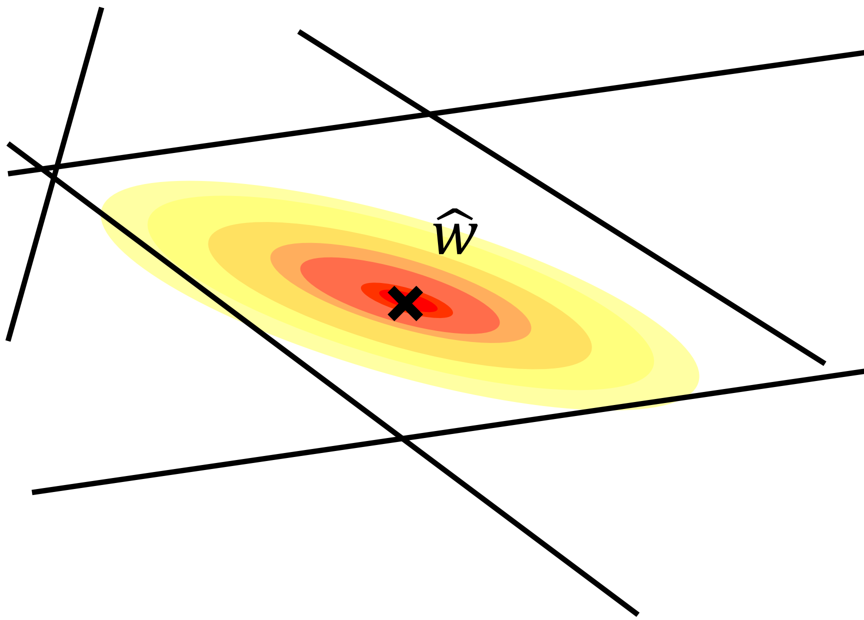
Modeling response noise

- $a_{pq} \in \mathbb{R}^d, b_{pq} \in \mathbb{R}$: weights, threshold of hyperplane bisecting p, q
- Model noise with logistic response probability

$$P(p \prec q) = \frac{1}{1 + e^{-k_{pq}(a_{pq}^T w - b_{pq})}}$$

  - k_{pq}: *noise constant*, represents signal-to-noise ratio
  - User estimated as posterior mean (MMSE estimator)





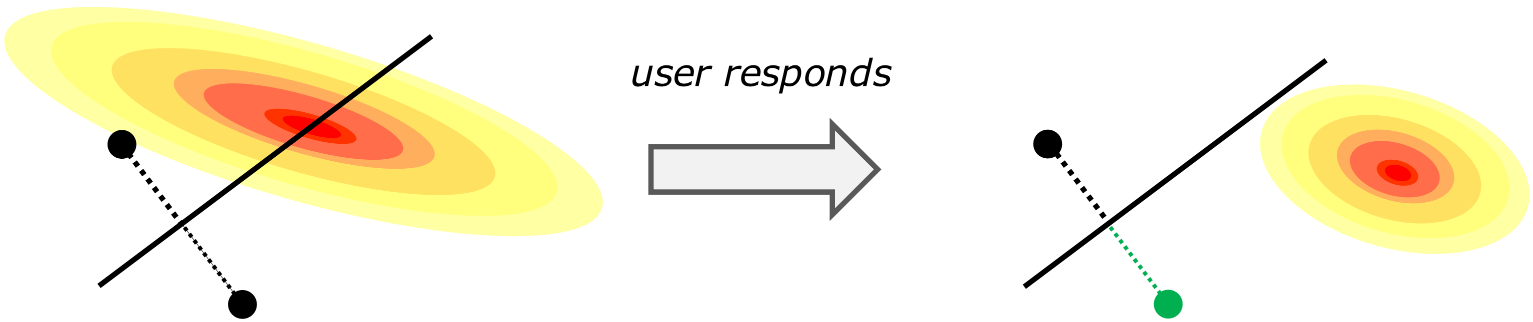
Our contribution

- ***Directly*** incorporate noise model into adaptive selection of pairs
- Strategy 1: InfoGain
- Strategy 2: EPMV
  - analytically tractable
- Strategy 3: MCMV
  - computationally tractable

Strategy 1: Maximize information gain (InfoGain)

- Y_i: binary response to i^th paired comparison
- $h_i(W)$: differential entropy of posterior
- *InfoGain*: choose queries that maximize expected decrease in posterior entropy i.e. *information gain*:

$$I\left(W; Y_i \middle| y^{i-1}\right) = h_{i-1}(W) - E_{Y_i}[h_i(W)|y^{i-1}]$$



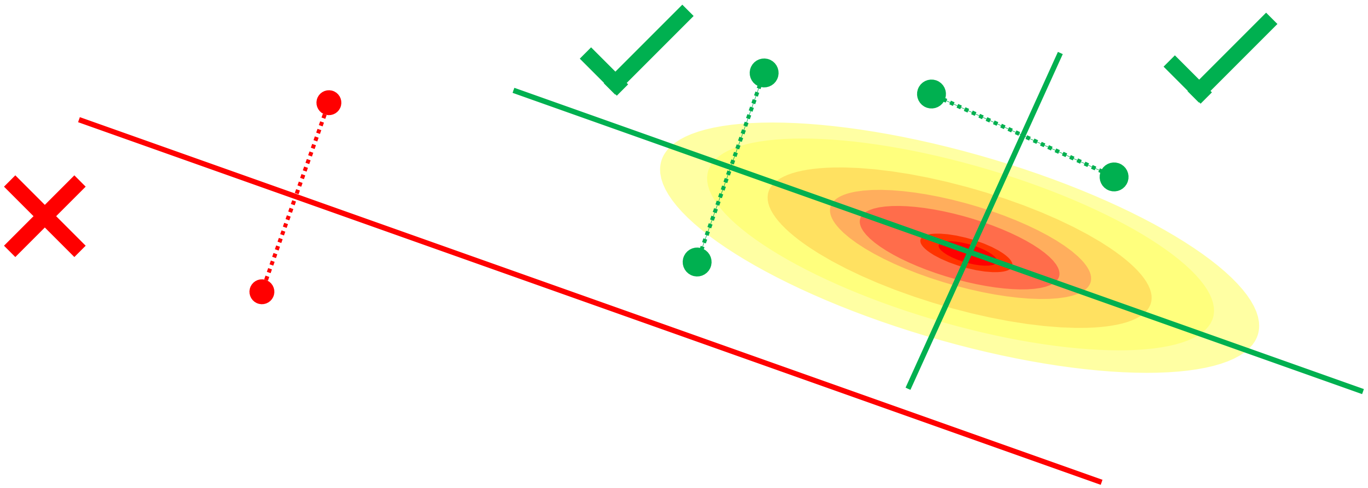
- No closed-form expression, estimate with samples from posterior
  - Computationally ***expensive***: scales in product of # of samples and # candidate pairs
- Difficult to analyze convergence

Information gain intuition

- Symmetry of mutual information:

$$I\left(W; Y_i \middle| y^{i-1}\right) = \boxed{H\left(Y_i \middle| y^{i-1}\right)} - H\left(Y_i \middle| W, y^{i-1}\right)$$

- First term promotes selection of comparisons where outcome is non-obvious, given previous responses
  - Maximized when comparison response is ***equiprobable***, i.e. probability of picking each pair item is 1/2



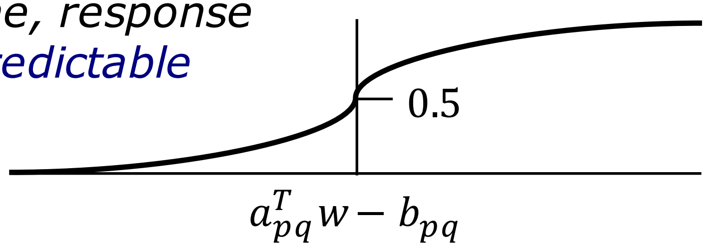
Information gain intuition

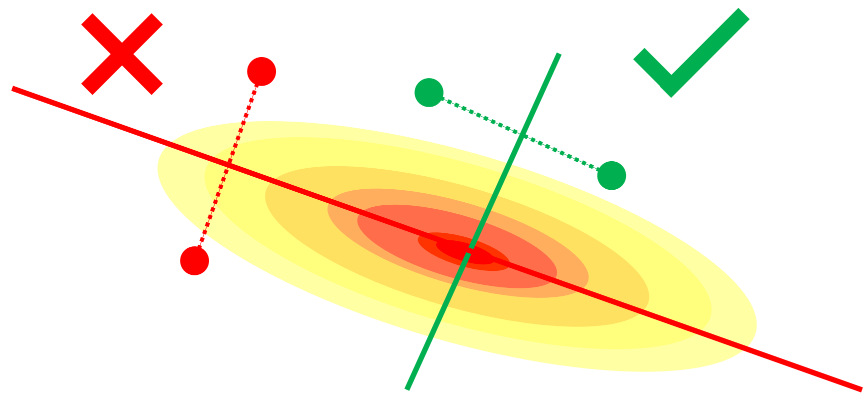
- Symmetry of mutual information:

$$I\left(W;Y_i\middle|y^{i-1}\right) = H(Y_i|y^{i-1}) - \boxed{H(Y_i|W,y^{i-1})}$$

- Second term promotes selection of comparisons that would have predictable outcomes if w were known

When w close to hyperplane, response is unpredictable





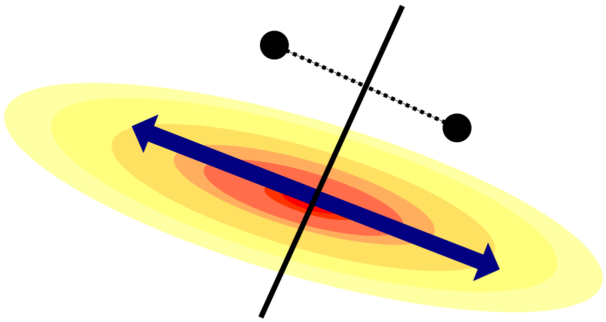
- Choose query where w is ***far*** from hyperplane in expectation
  - i.e. posterior variance orthogonal to hyperplane (*projected variance*) is ***large***

Strategy 2: Equiprobable, max-variance (EPMV)

$$I\left(W; Y_i \middle| y^{i-1}\right) = \boxed{H(Y_i | y^{i-1})} - \boxed{H(Y_i | W, y^{i-1})}$$

- *Equiprobable*: response is equally likely to be either item
  - Determines hyperplane threshold
- *Max-variance:* comparison cuts in direction of maximum projected variance
  - Determines hyperplane weights

$$P(p \prec q) = 1/2$$



EPMV theory

Proposition

For equiprobable comparison with hyperplane weights a_{pq}*,*

$$I\left(W; Y_i \middle| y^{i-1}\right) \geq L_1\left(a_{pq}^T \Sigma_{W|Y^{i-1}} a_{pq}\right)$$

where L_1 *is a monotonically increasing function.*

- EPMV approximates InfoGain

EPMV theory

Theorem

For the EPMV query scheme with each selected query satisfying $k_{pq}\|a_{pq}\| \geq k_{min} > 0$ *and stopping threshold* $\varepsilon > 0$*, consider the stopping time* $T_\varepsilon = \min\left\{i: \left|\Sigma_{W|y^i}\right|^{\frac{1}{d}} < \varepsilon\right\}$*. We have*

$$E[T_\varepsilon] = O(d \log\frac{1}{\varepsilon} + \frac{1}{\varepsilon k_{min}^2} d^2 \log\frac{1}{\varepsilon}).$$

Furthermore, for any query scheme $E[T_\varepsilon] = \Omega(d \log\frac{1}{\varepsilon})$*.*

- For large noise constants ($k_{min} \gg 0$), EPMV reduces the posterior volume at a nearly-optimal rate.

EPMV in practice

- Often, one selects pair from pool, rather than querying arbitrary hyperplanes
- Select pair that maximizes approximate EPMV utility function, for $\lambda > 0$

$$\arg\max_{p,q} \quad \overbrace{k_{pq}\sqrt{a_{pq}^T \Sigma_{W|Y^{i-1}} a_{pq}}}^{\text{Prefers max-variance queries}} - \lambda \overbrace{\left|\widehat{p}_1 - \frac{1}{2}\right|}^{\text{Prefers equiprobable queries}}$$

$$\widehat{p}_1 = P(Y_i = 1 | Y^{i-1}) = \mathbb{E}_{W|Y^{i-1}}\left[\frac{1}{1 + e^{-k_{pq}(a_{pq}^T W - b_{pq})}}\right]$$

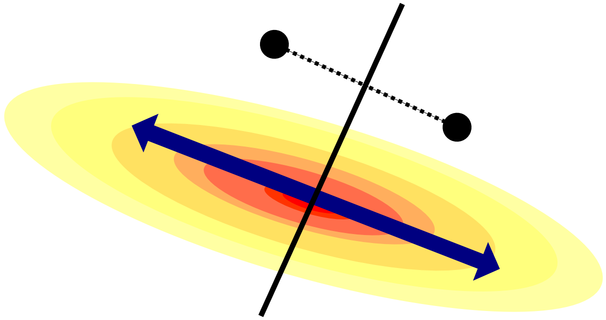
- Computationally ***expensive*** – same utility evaluation cost as InfoGain

Strategy 3: Mean-cut, max-variance (MCMV)

$$I\left(W;Y_i\middle|y^{i-1}\right) = \boxed{H(Y_i|y^{i-1})} - \boxed{H(Y_i|W,y^{i-1})}$$

- Computational bottleneck in EPMV is evaluating equiprobable property
  - Approximate equiprobable property with *mean-cut* property i.e. hyperplane passes through posterior mean

$$a_{pq}^T E\left[W\middle|Y^{i-1}\right] - b_{pq} = 0$$



MCMV theory

Proposition
For mean-cut comparisons with $a_{pq}^T \Sigma_{W|Y^{i-1}} a_{pq} \gg 0$,

$$|p(Y_i|y^{i-1}) - 1/2| \lessapprox 0.14$$

- ➢ MCMV approximates EPMV

Proposition
For mean-cut comparison with hyperplane weights a_{pq},

$$I\left(W; Y_i \middle| y^{i-1}\right) \geq L_2\left(a_{pq}^T \Sigma_{W|Y^{i-1}} a_{pq}\right)$$

where L_2 *is a monotonically increasing function.*

- ➢ MCMV approximates InfoGain

MCMV in practice

- Select pair that maximizes utility function, for $\lambda > 0$

$$\arg\max_{p,q} \quad \underbrace{k_{pq}\sqrt{a_{pq}^T \Sigma_{W|Y^{i-1}} a_{pq}}}_{\text{Prefers max-variance queries}} - \lambda \underbrace{\frac{|a_{pq}^T \mathbb{E}[W|Y^{i-1}] - b_{pq}|}{\|a_{pq}\|_2}}_{\text{Prefers mean-cut queries}}$$

- Computational cost is ***much*** cheaper than InfoGain and EPMV
  - Scales with *sum* of # number of posterior samples and # candidate pairs, rather than *product*

Methods overview

Method	**Advantages**	**Limitations**
InfoGain	Directly minimizes posterior volume	Computationally expensive Difficult to analyze
EPMV	Convergence guarantee	Computationally expensive
MCMV	Computationally cheap	No convergence guarantee (future work)

Simulated results

- Item embedding constructed from Yummly Food-10k dataset (Wilber et al., 2015; 2014)
  - 10,000 food items
  - ~1 million human comparisons between items



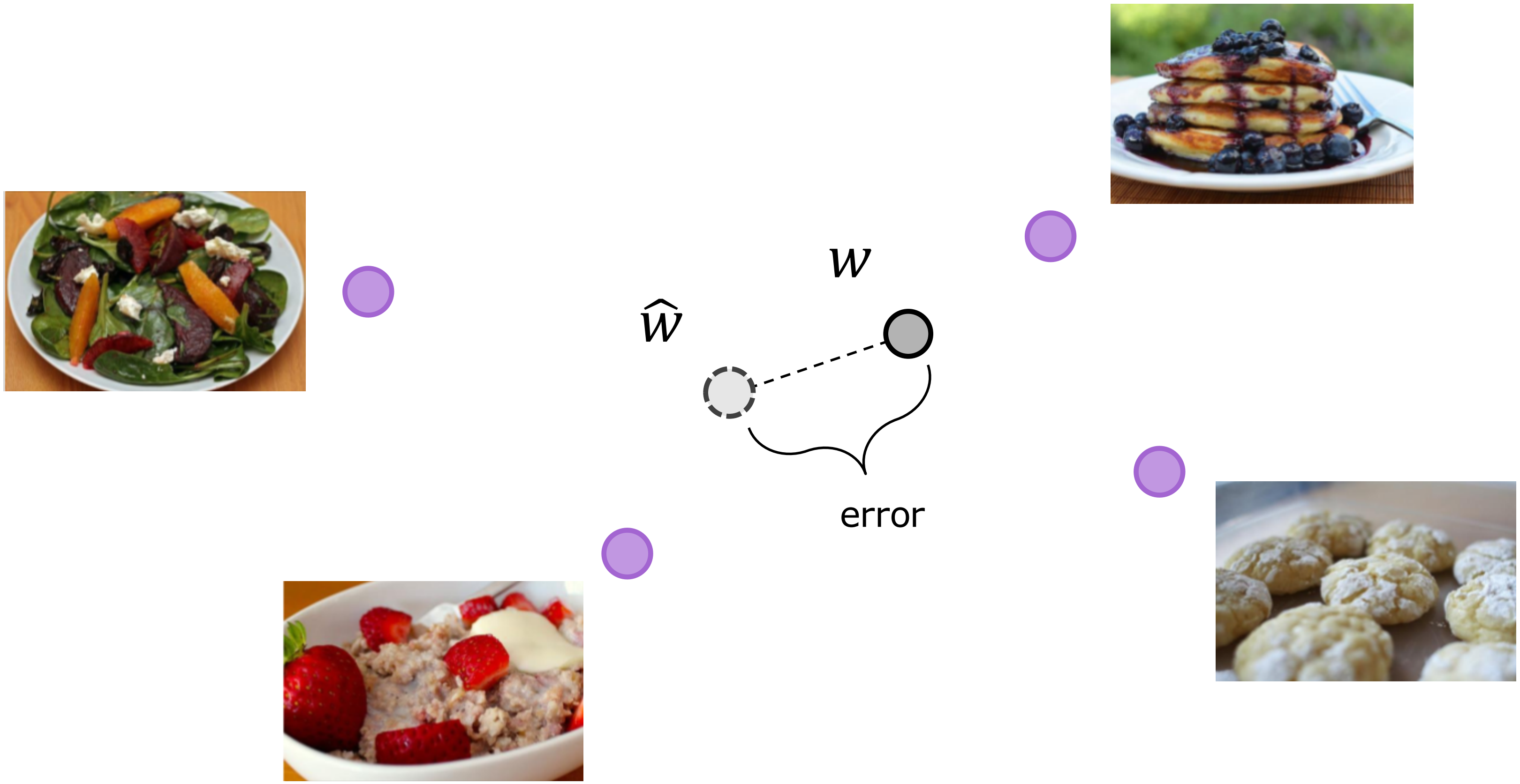

- Simulated pairwise search
  - Noise constant k_{pq} estimated from training comparisons
  - User preference point drawn uniformly from hypercube, $d = 4$

Simulated results – baseline methods

- *Random*
  - pairs selected uniformly at random
  - user estimated as posterior mean
- *GaussCloud* (Massimino & Davenport, 2018)
  - pairs chosen to approximate Gaussian point cloud around estimate, shrinks over multiple stages
  - user estimated by approximately solving non-convex program
- *ActRank* (Jamieson & Nowak, 2011)
  - pairs selected that intersect feasible region of preference points
  - query repeated multiple times, majority vote taken
  - user estimated as Chebyshev center*

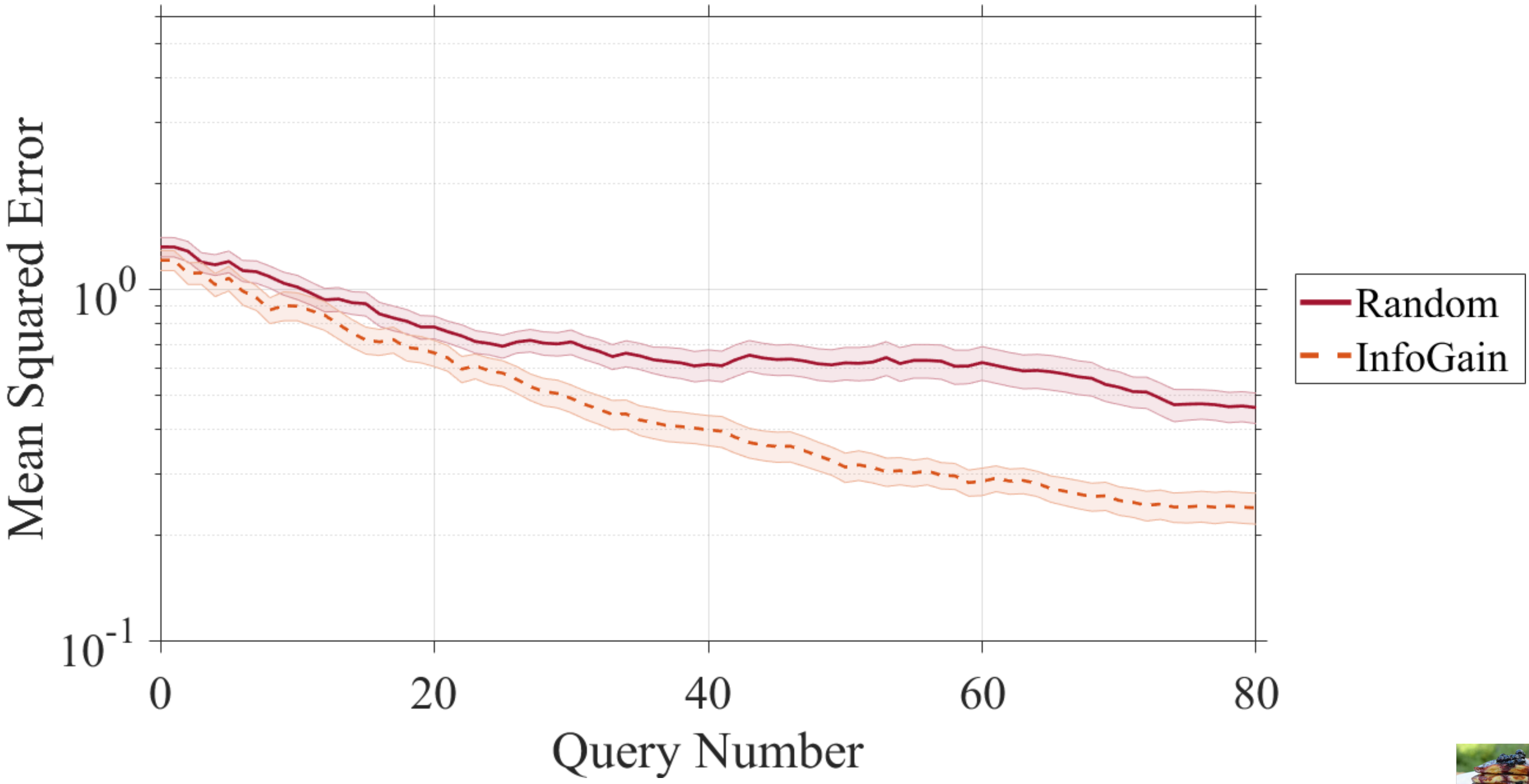
* our addition

Simulated results - MSE

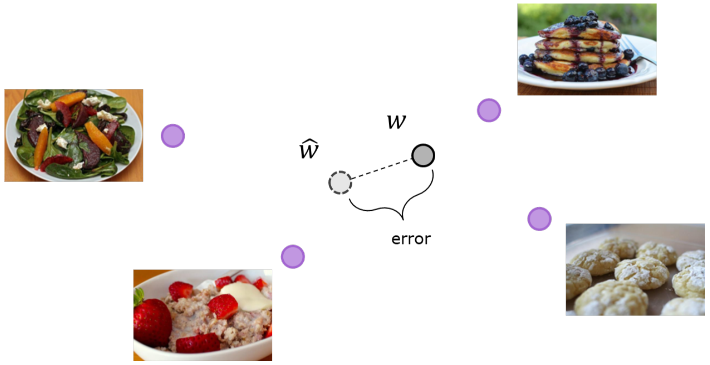


- Mean-squared error (MSE) measures accuracy in estimating user preference point

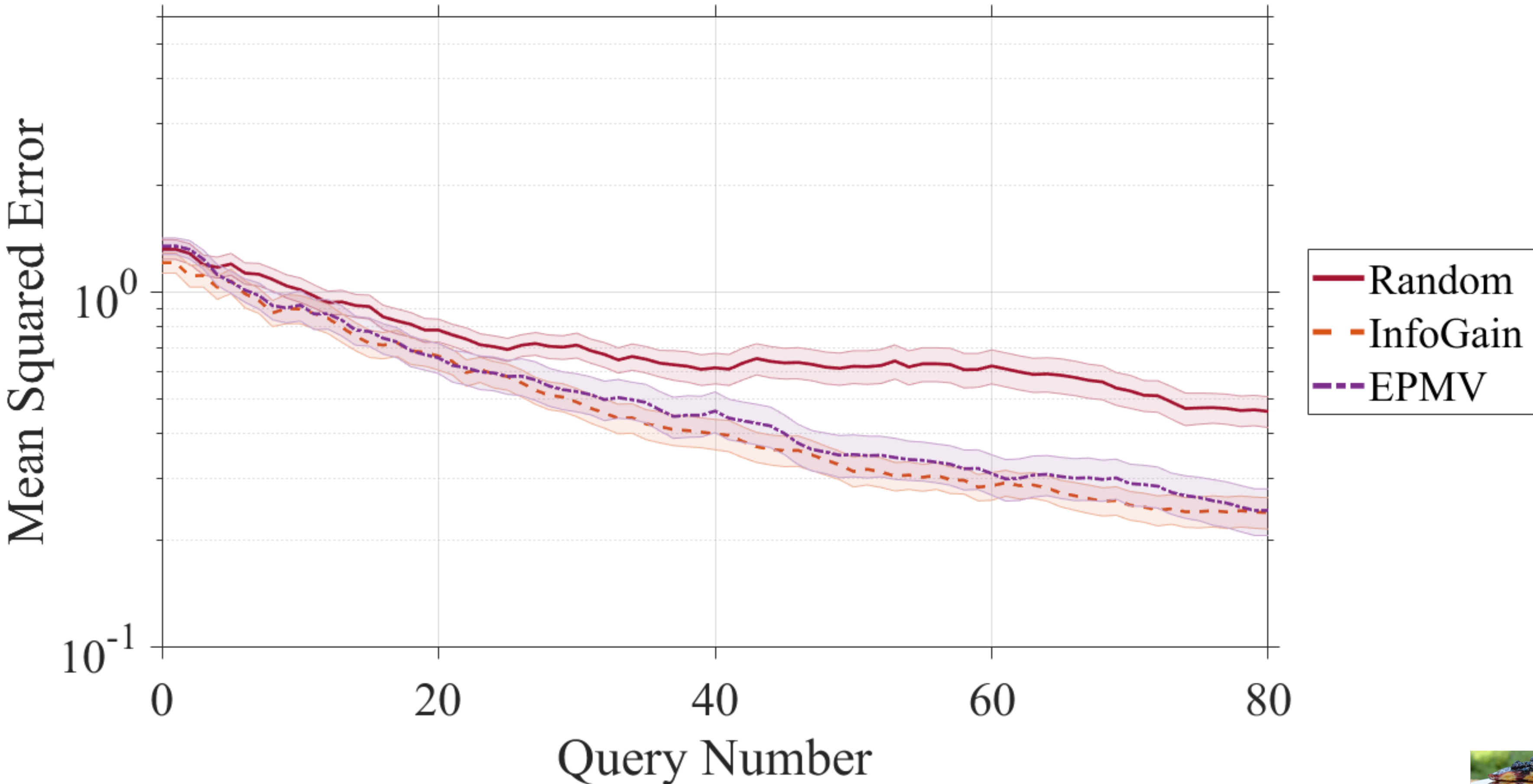
Simulated results - MSE



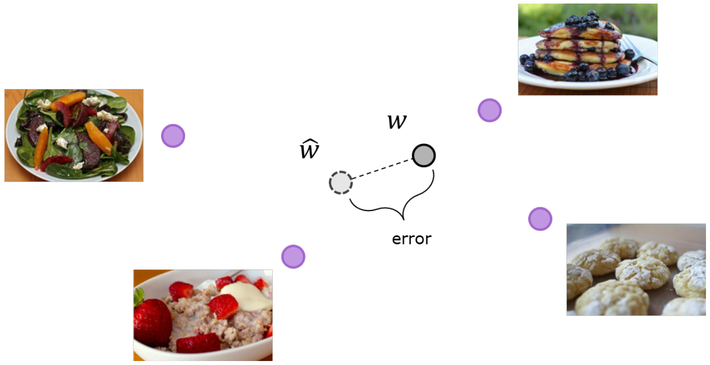
- Mean-squared error (MSE) measures accuracy in estimating user preference point



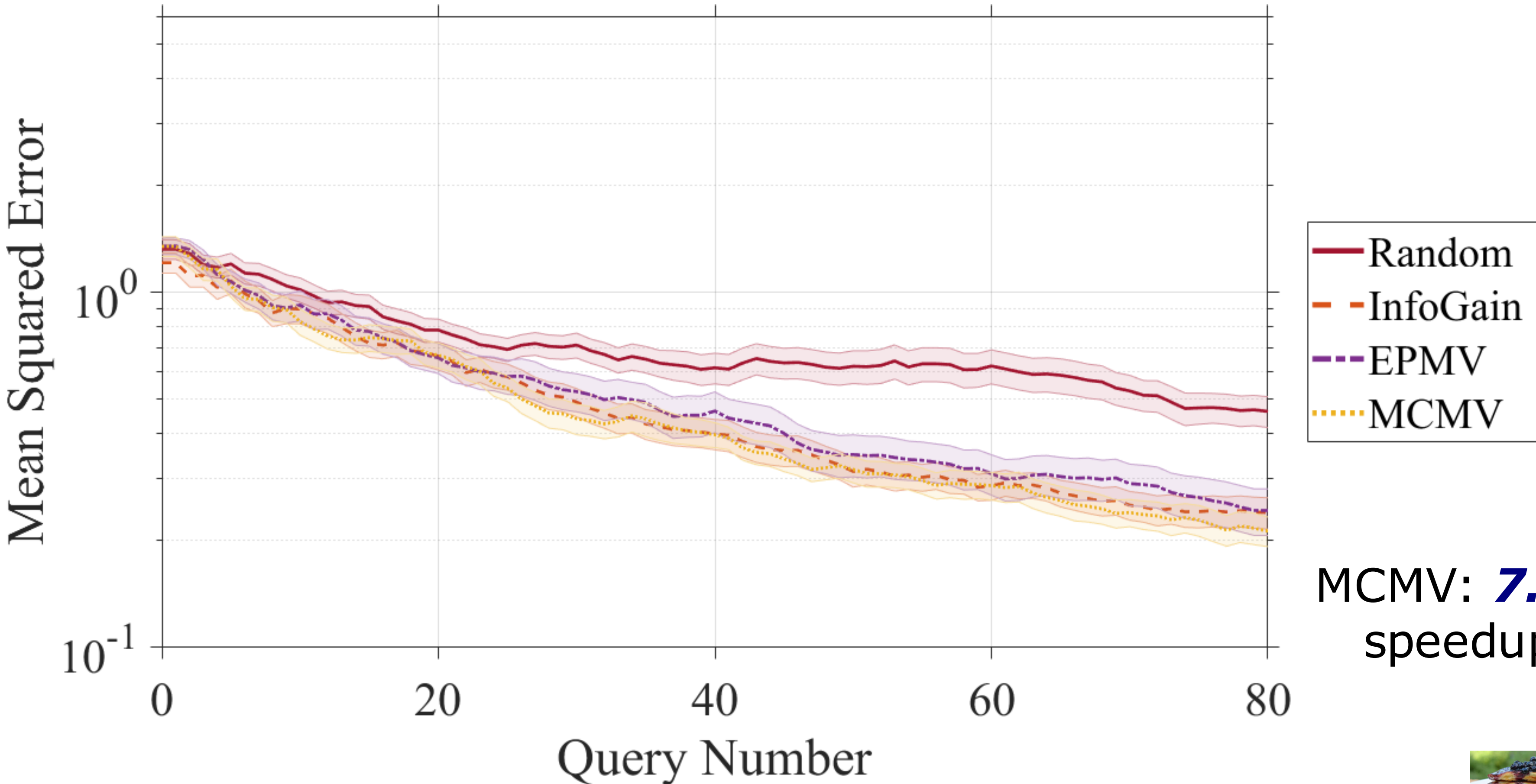
Simulated results - MSE



- Mean-squared error (MSE) measures accuracy in estimating user preference point

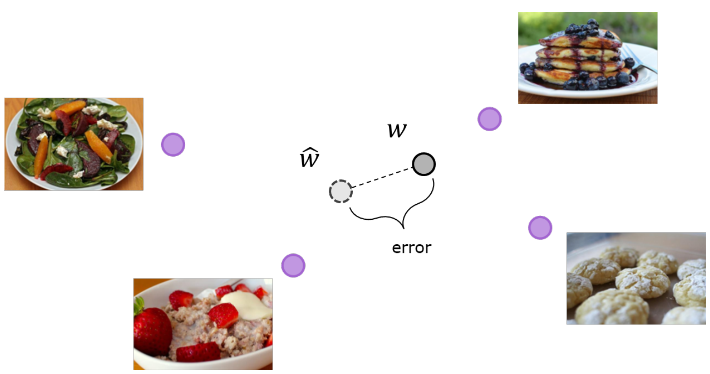


Simulated results - MSE

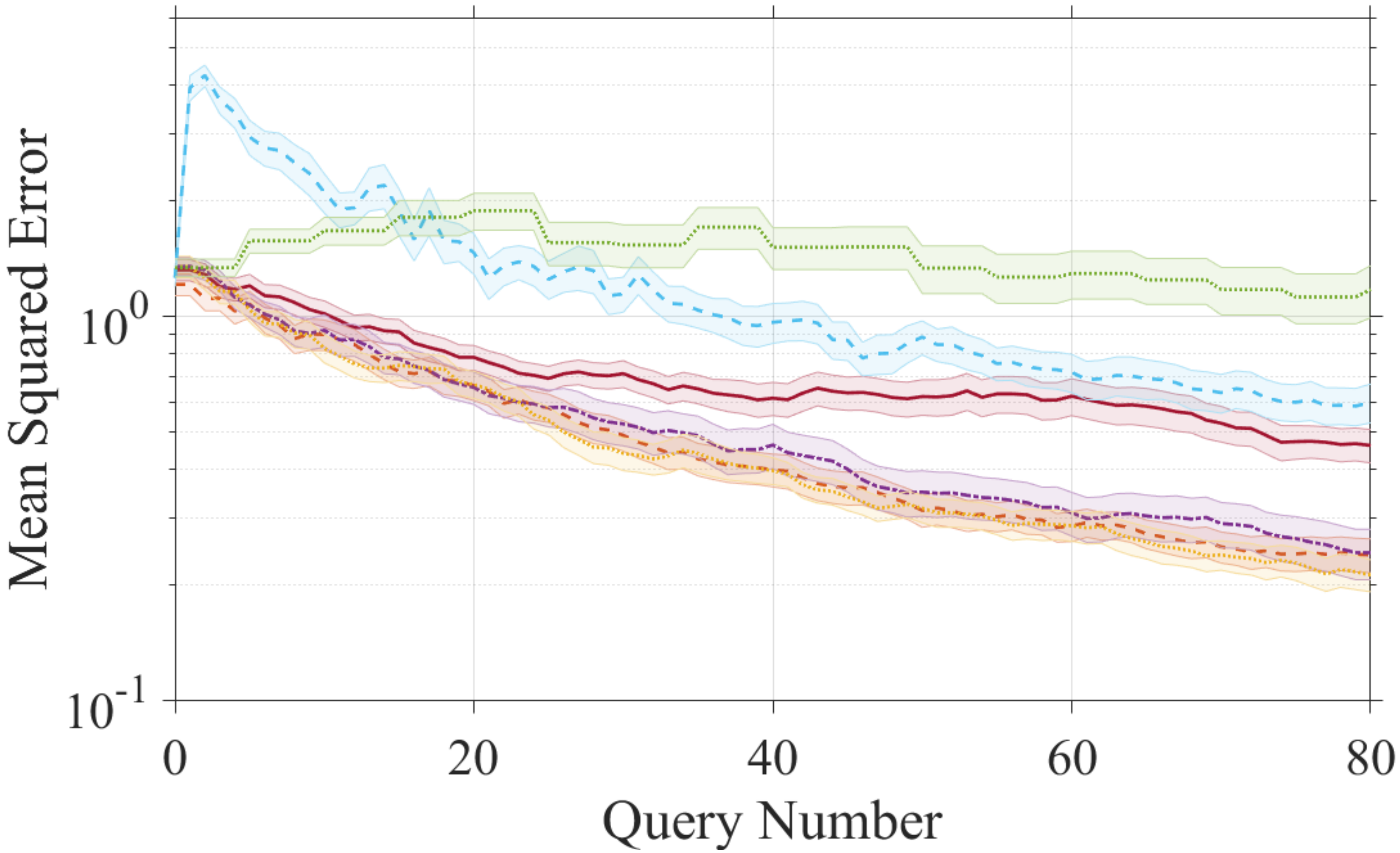


MCMV: ***7.5x*** speedup

- Mean-squared error (MSE) measures accuracy in estimating user preference point

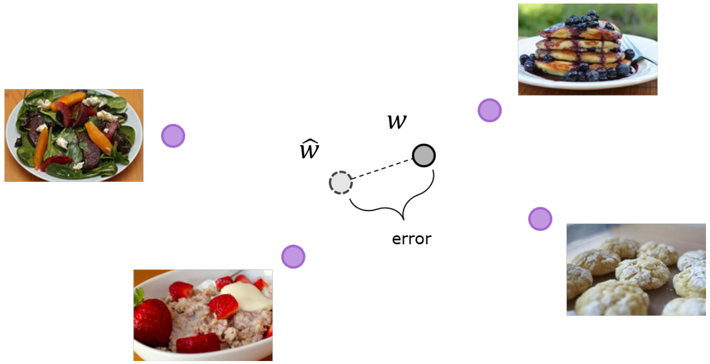


Simulated results - MSE

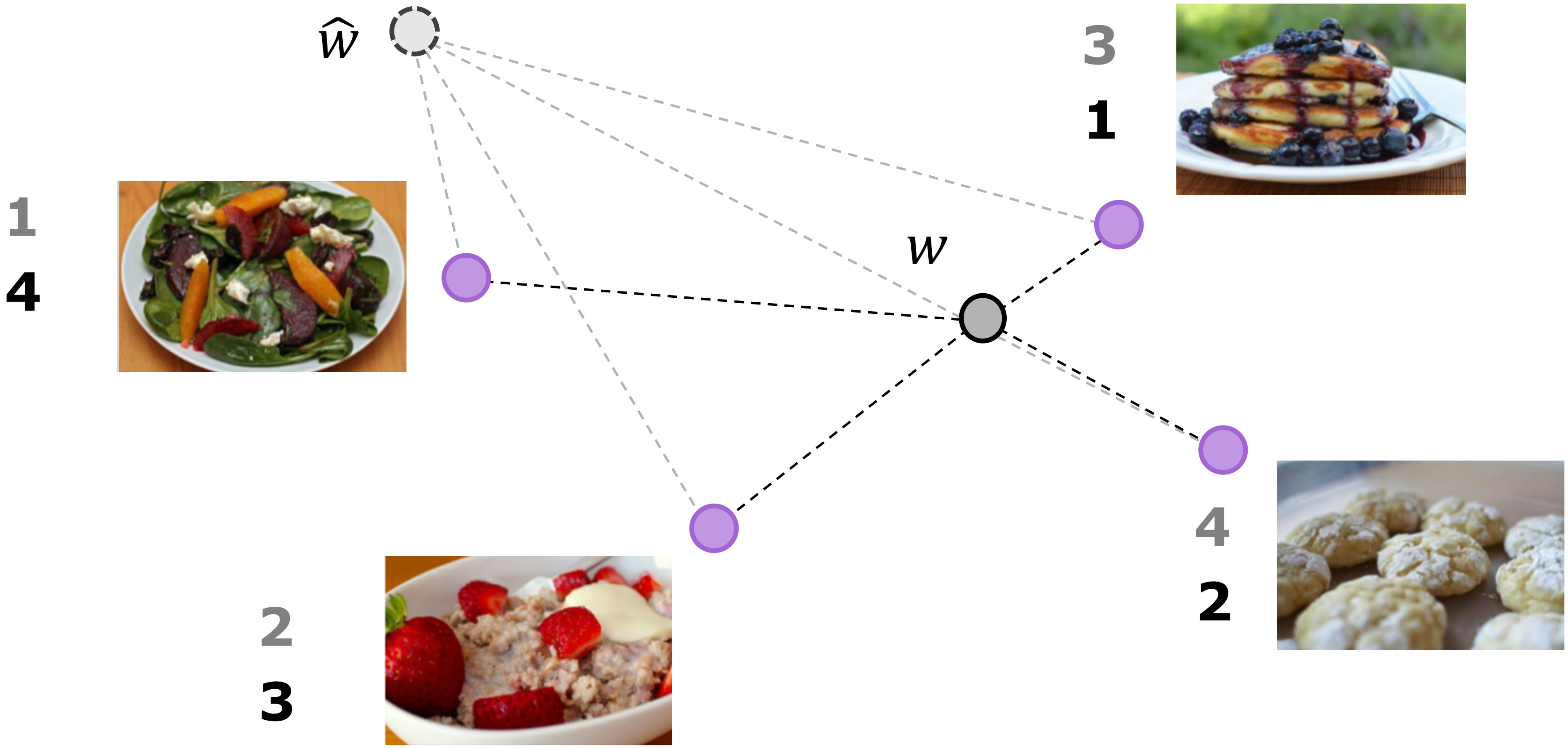


MCMV: ***7.5x*** speedup

- Mean-squared error (MSE) measures accuracy in estimating user preference point

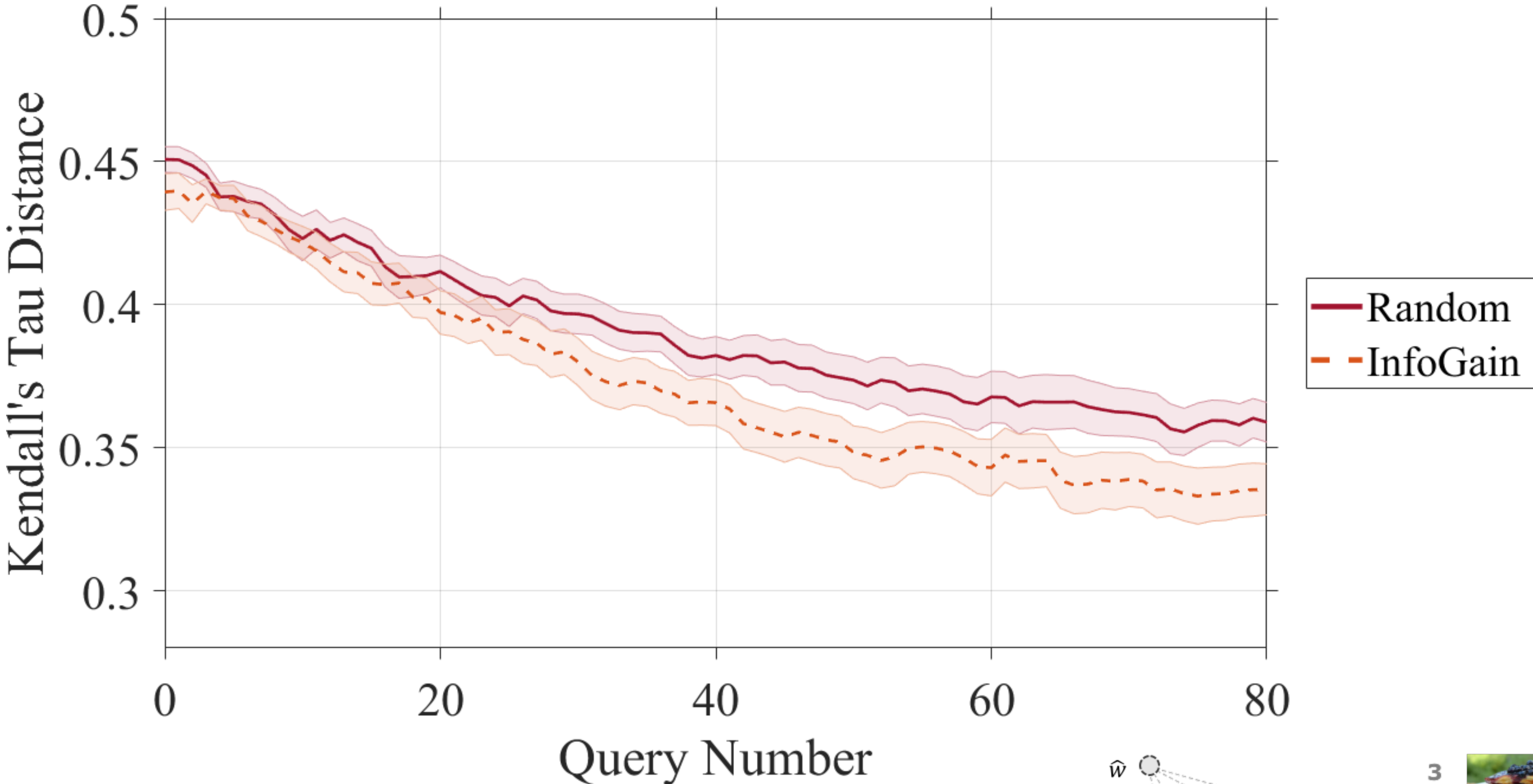


Simulated results – Kendall's Tau distance

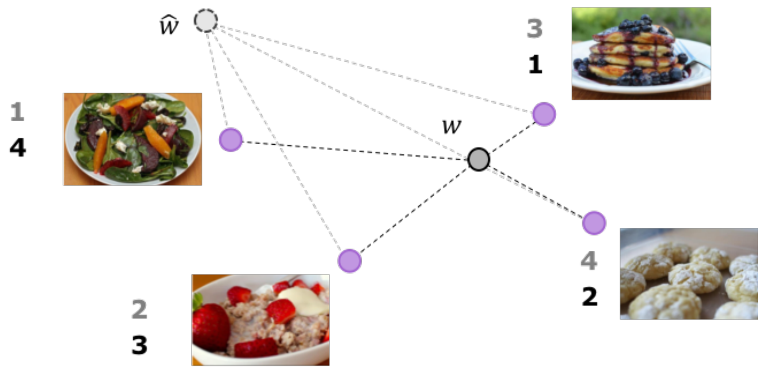


- Normalized Kendall's Tau distance for preference ranking of 15 randomly selected items

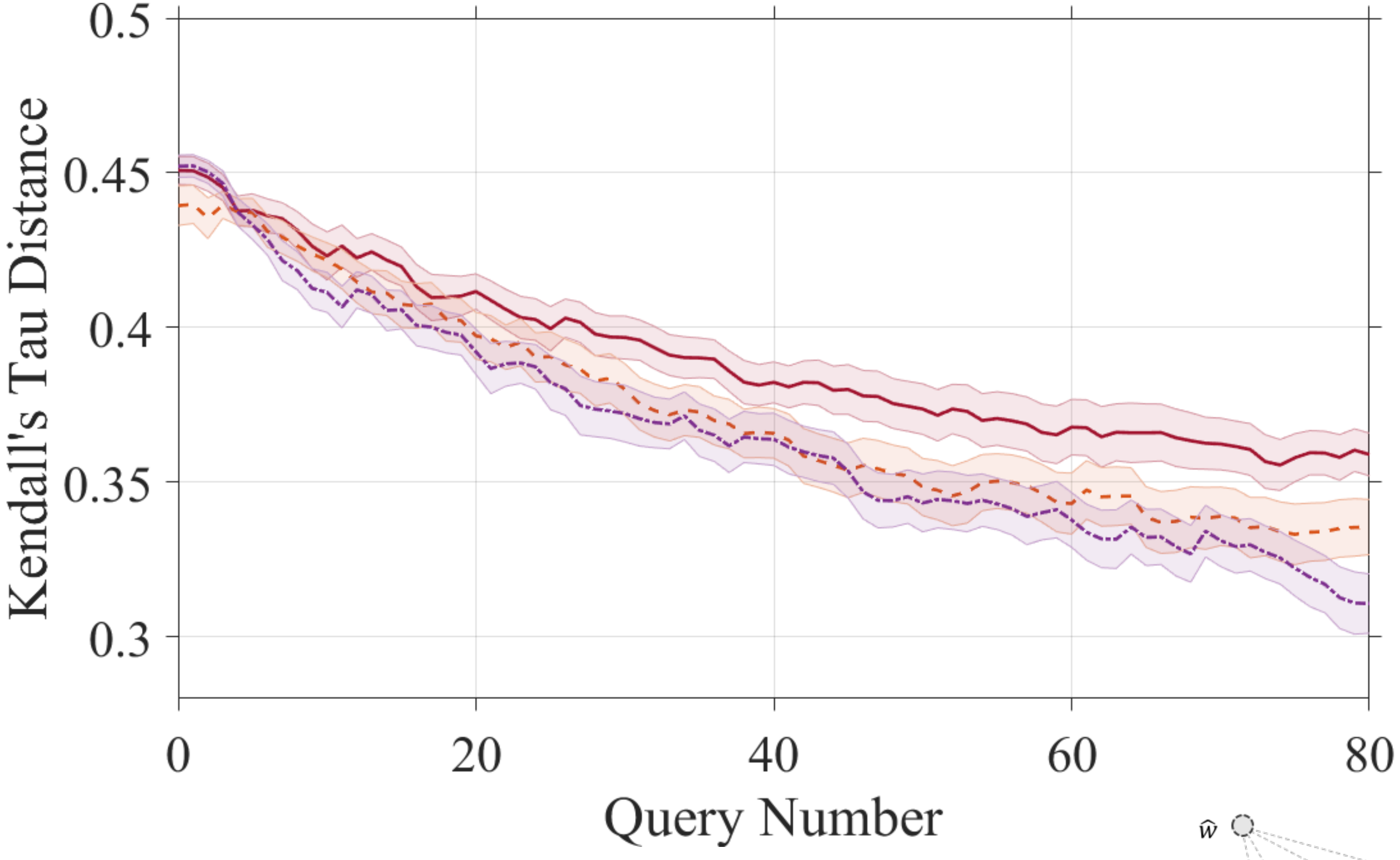
Simulated results – Kendall's Tau distance



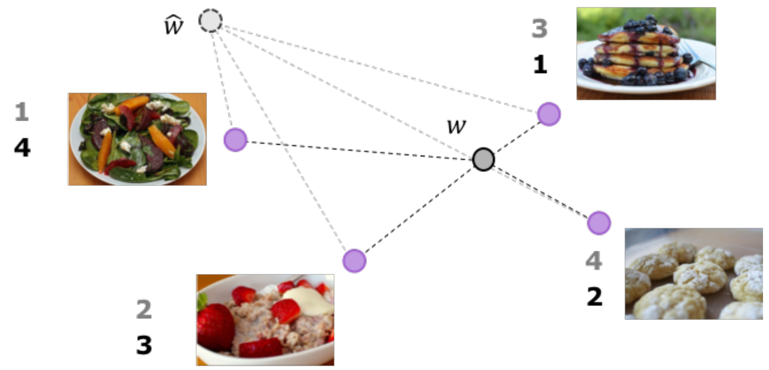
- Normalized Kendall's Tau distance for preference ranking of 15 randomly selected items



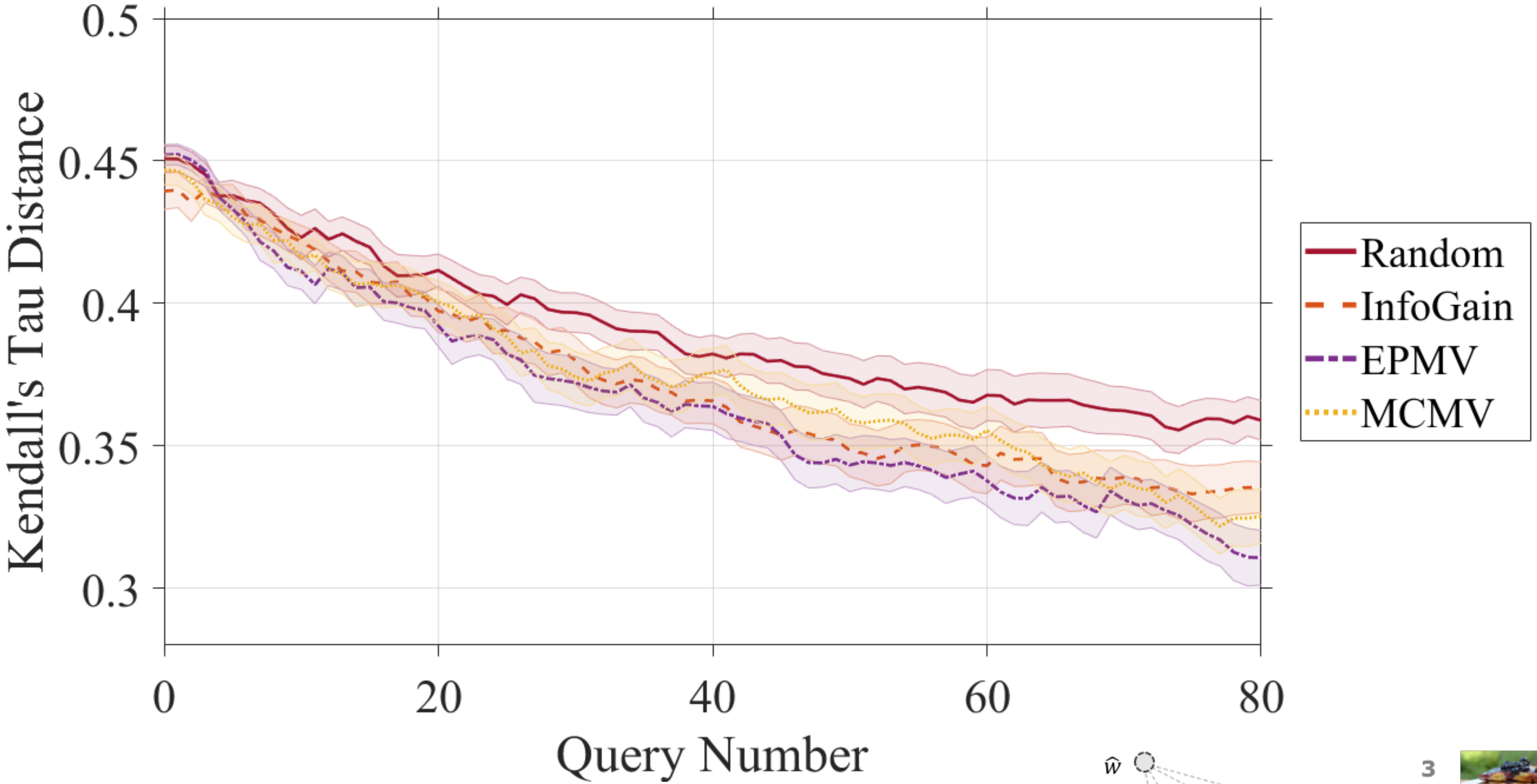
Simulated results – Kendall's Tau distance



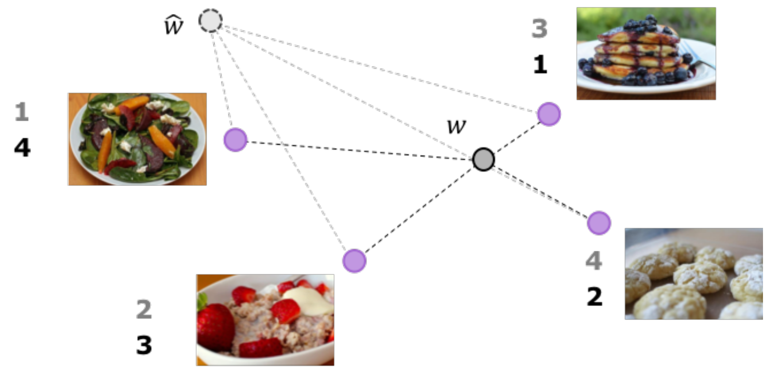
- Normalized Kendall's Tau distance for preference ranking of 15 randomly selected items



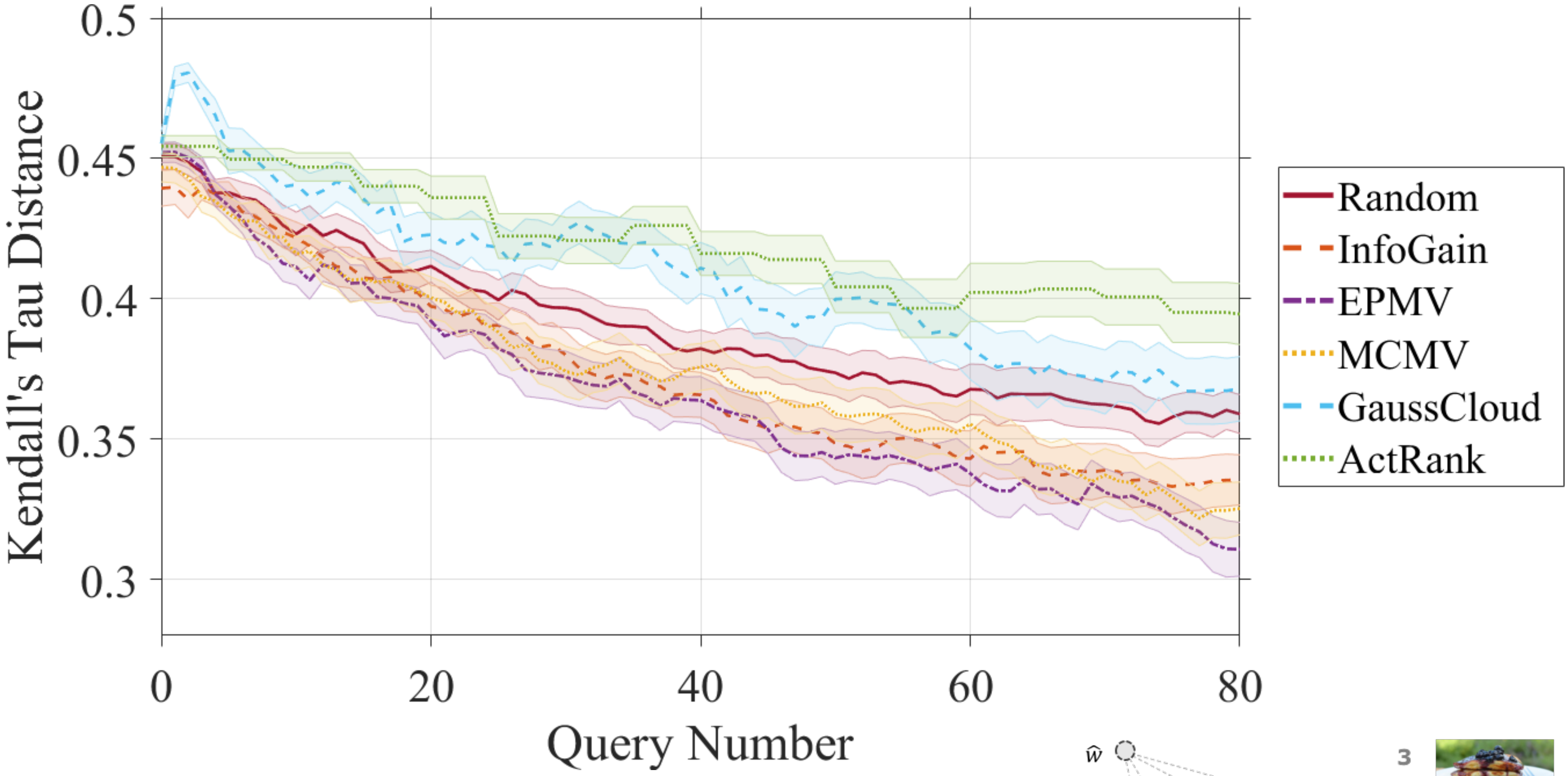
Simulated results – Kendall's Tau distance



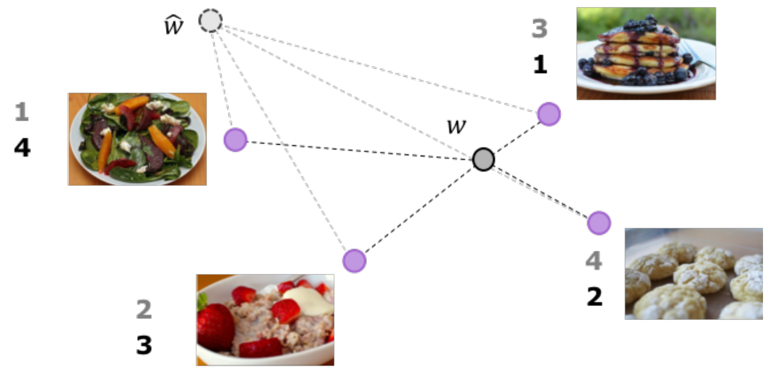
- Normalized Kendall's Tau distance for preference ranking of 15 randomly selected items



Simulated results – Kendall's Tau distance



- Normalized Kendall's Tau distance for preference ranking of 15 randomly selected items



Takeaways

- First effort to directly model noise in active pairwise preference learning for ideal point model
  - InfoGain
  - Equiprobable max-variance (EPMV)
  - Mean-cut max-variance (MCMV)
- Preliminary support for robustness to noise mismatch
- Potential applications
  - Advertising, online shopping
  - Parameter settings
  - Product customization, recipe generation
  - Database search (medical records, faces)

Code available at: https://github.com/siplab-gt/pairsearch

http://siplab.gatech.edu

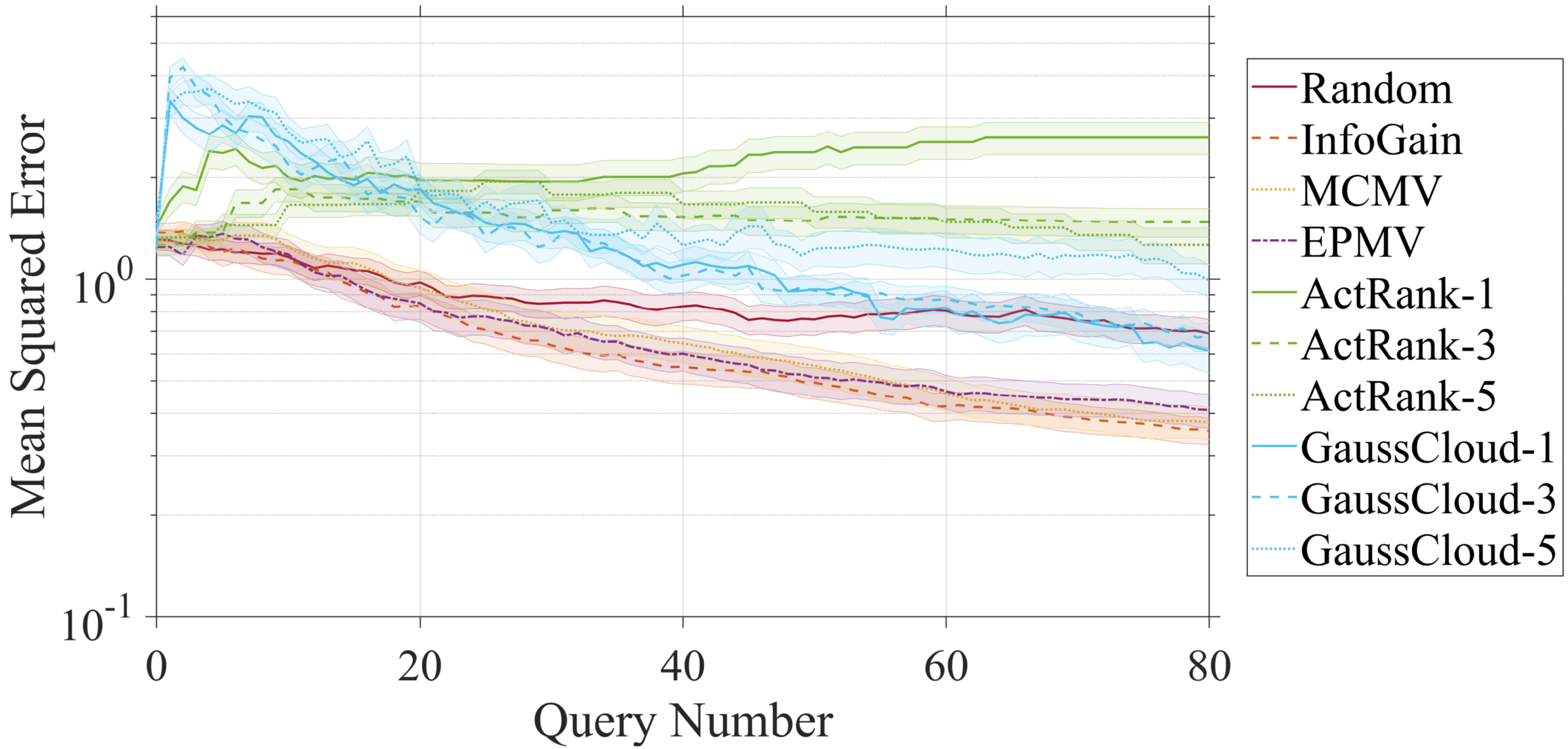
gregory.canal@gatech.edu

@GregHCanal

POSTER #260

TODAY, 6:30 – 9:00 PM, Pacific Ballroom

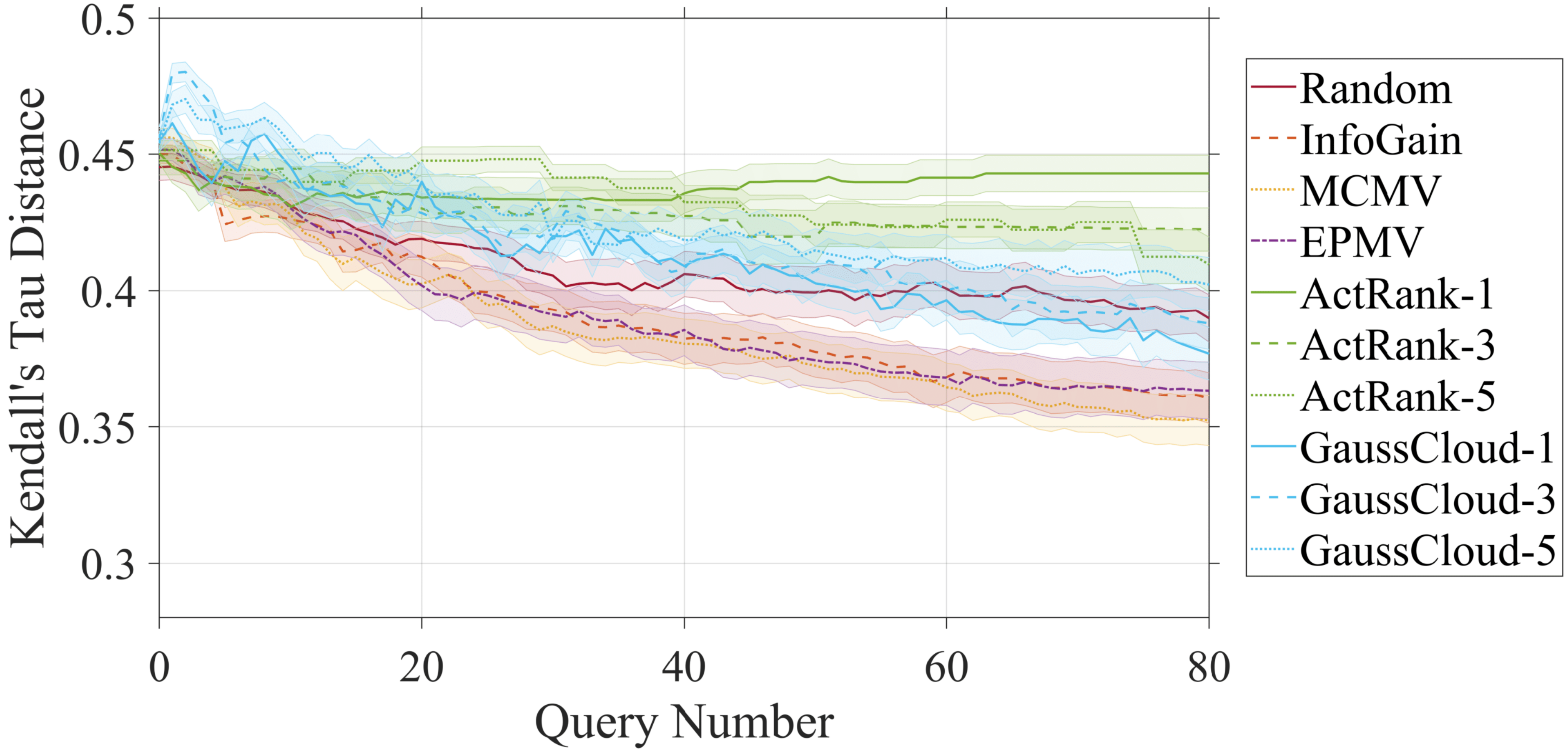
Simulated mismatched noise - MSE



$$y_i = \text{sign}(k_{pq}(a_i^T w - b_i) + Z)$$

$$Z \sim \mathcal{N}(0, 1)$$

Simulated mismatched noise – Kendall's Tau



$$y_i = \text{sign}(k_{pq}(a_i^T w - b_i) + Z)$$

$$Z \sim \mathcal{N}(0, 1)$$