Data-Centric Approaches to Recommendation

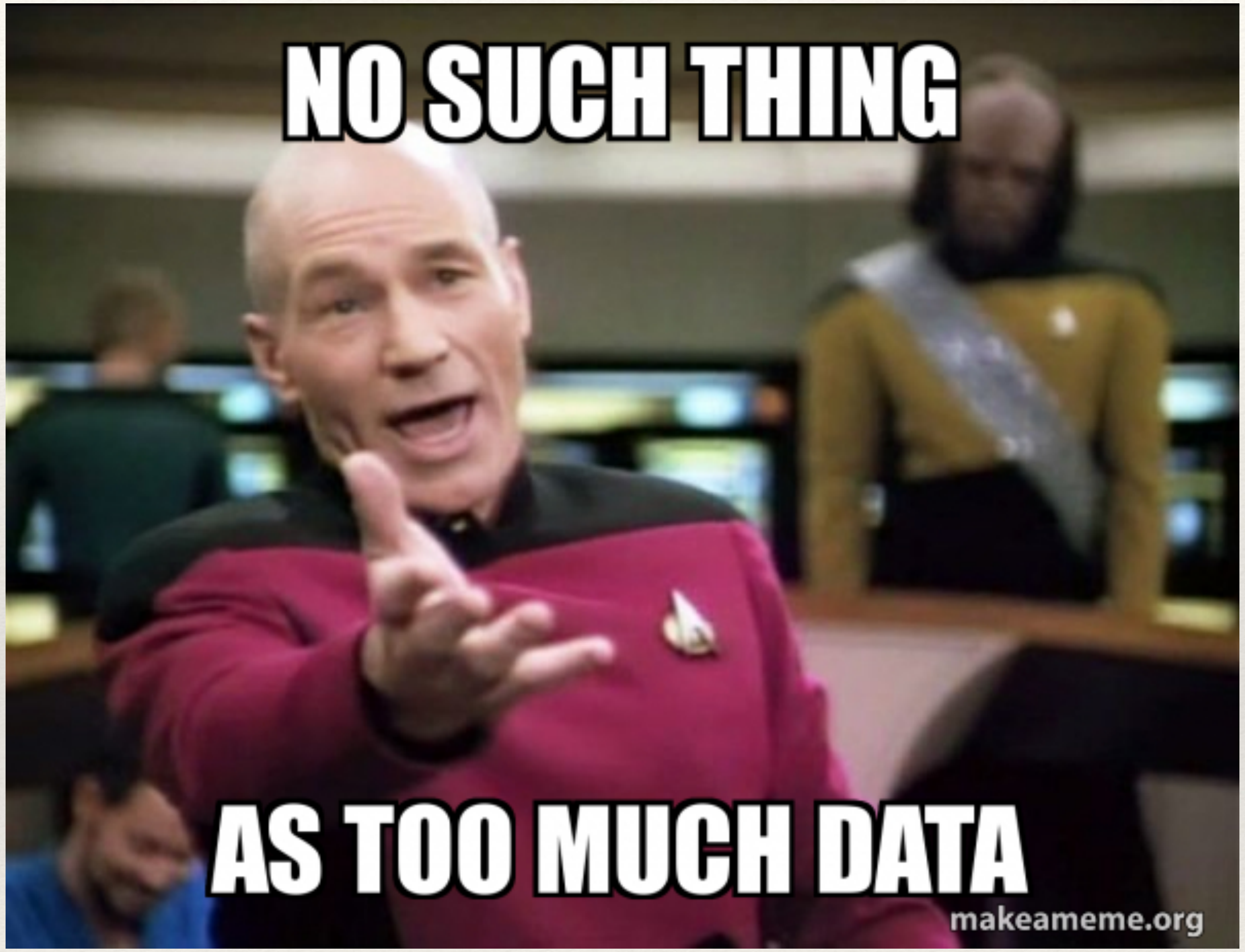
Question: Is **more data** what you need for **better recommendation**?

Noveen Sachdeva

@noveens97

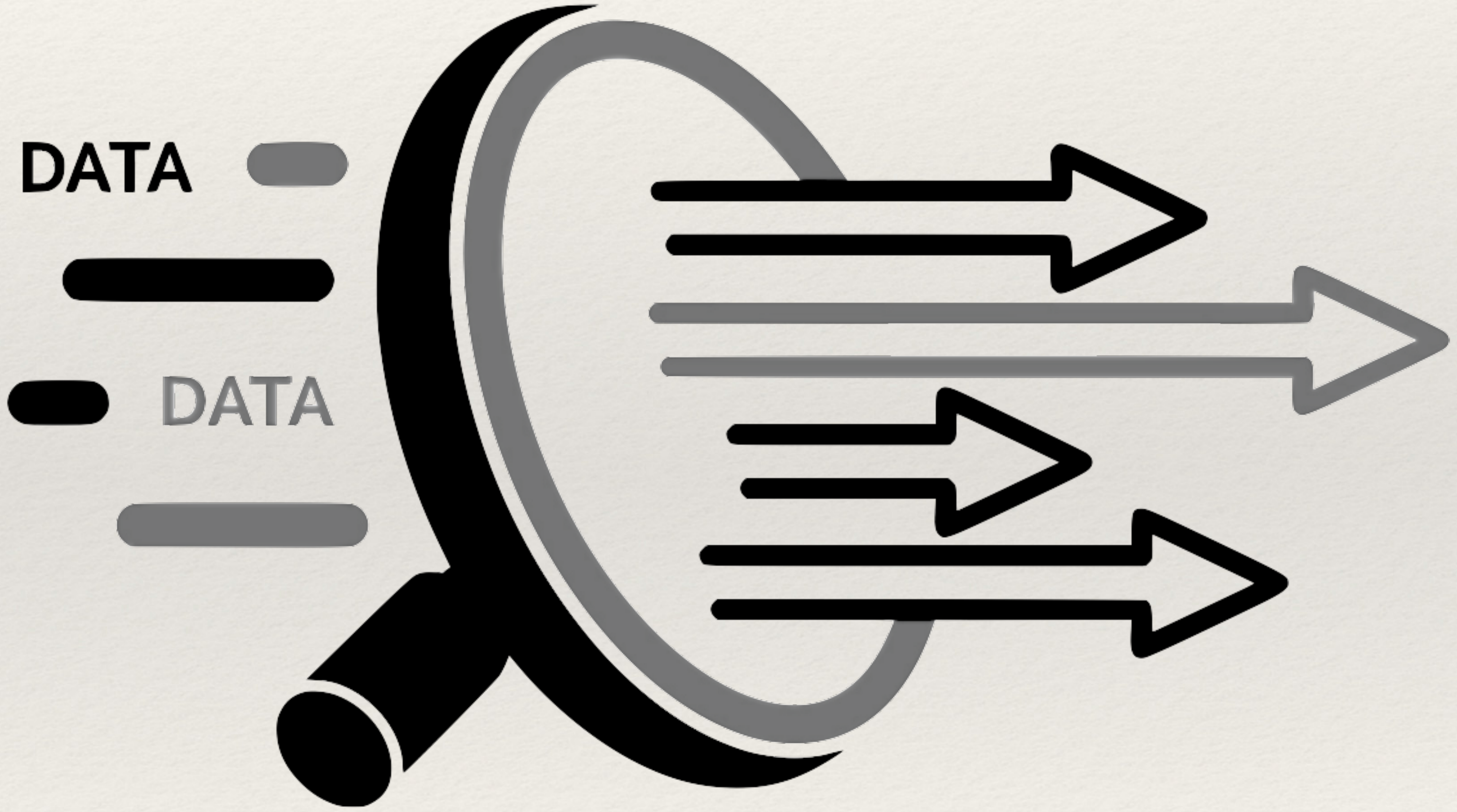
UC San Diego

Google Brain, July 2022


NO SUCH THING
AS TOO MUCH DATA
makeameme.org

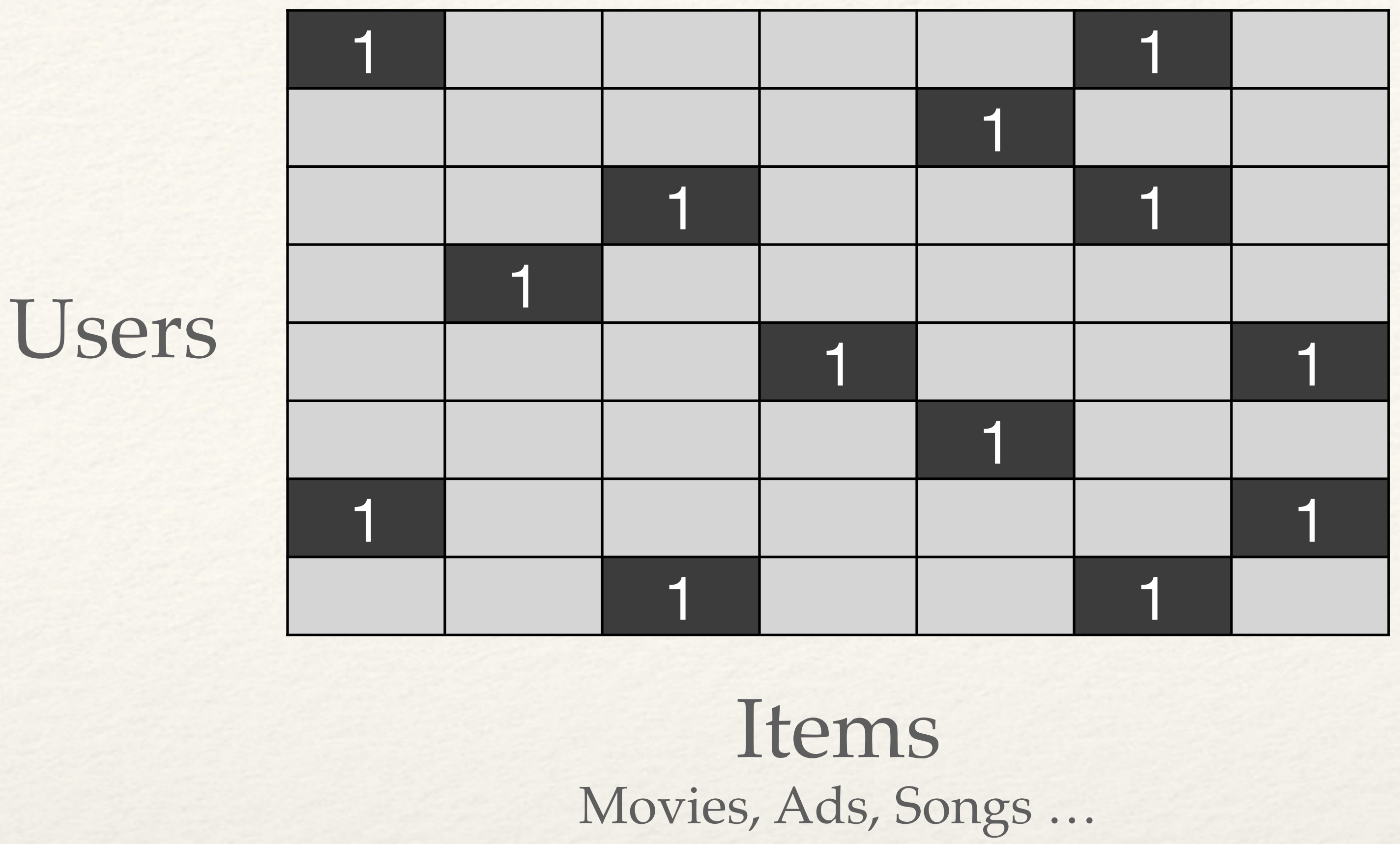
Talk Layout

- Primer, Premise & Scope
- SVP-CF & Data-Genie 🧞
- Infinite Recommendation Networks
- Dataset Distillation (Distill-CF)
- Future Directions

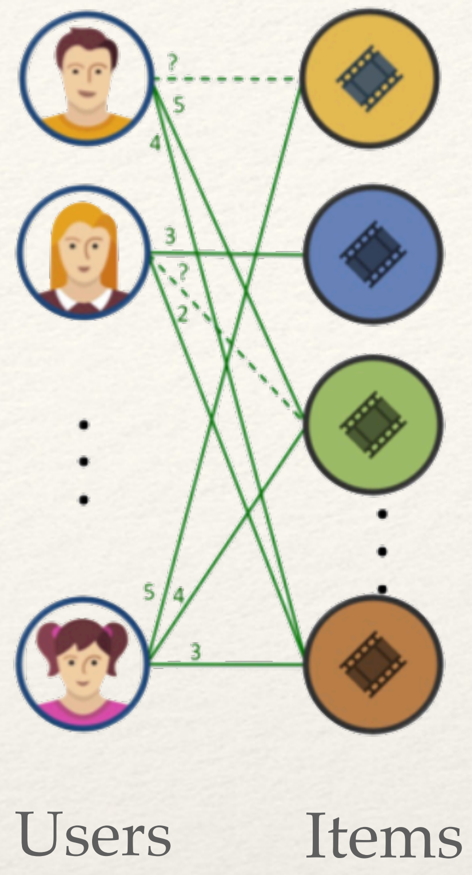


Primer

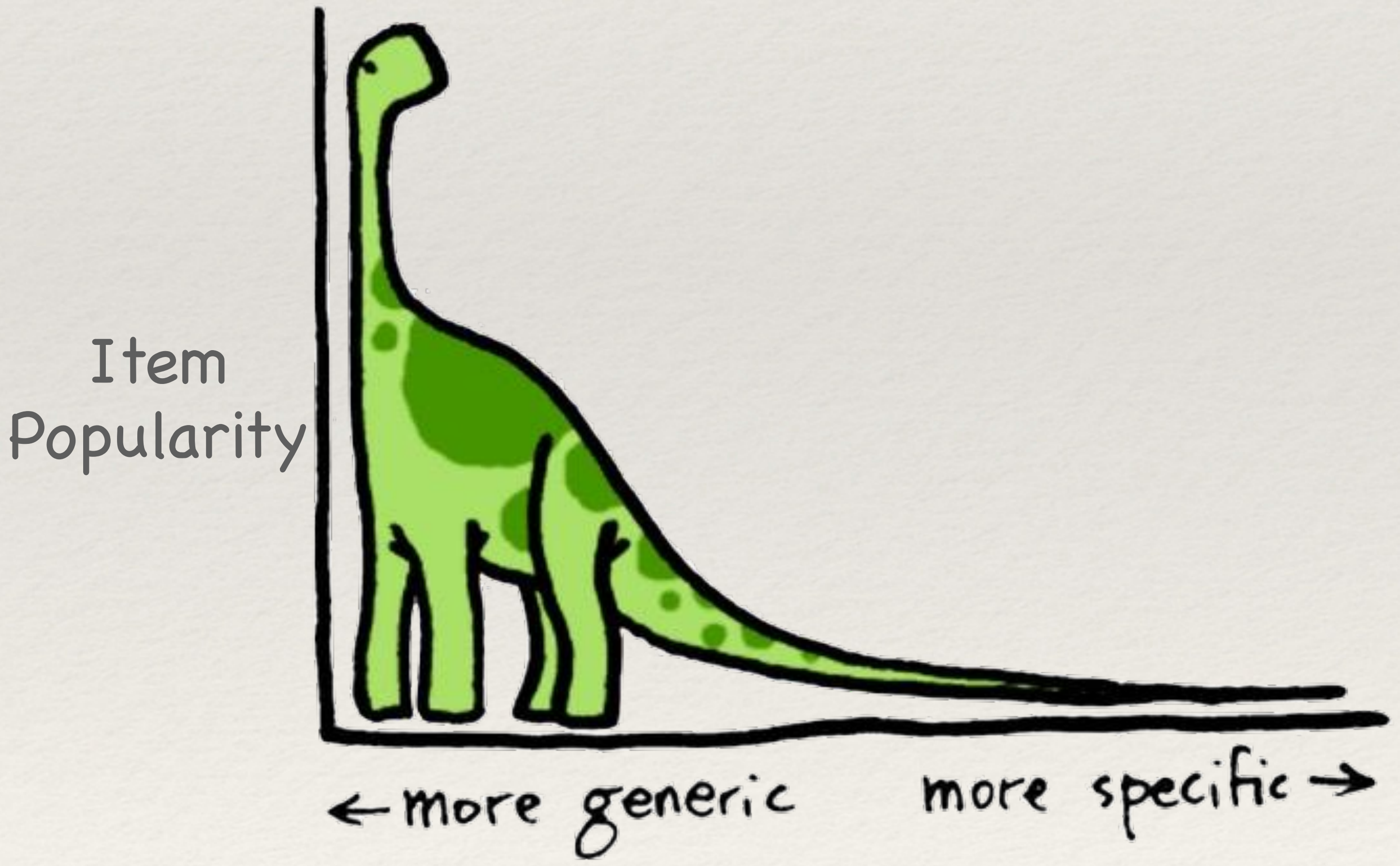
Recommender Systems

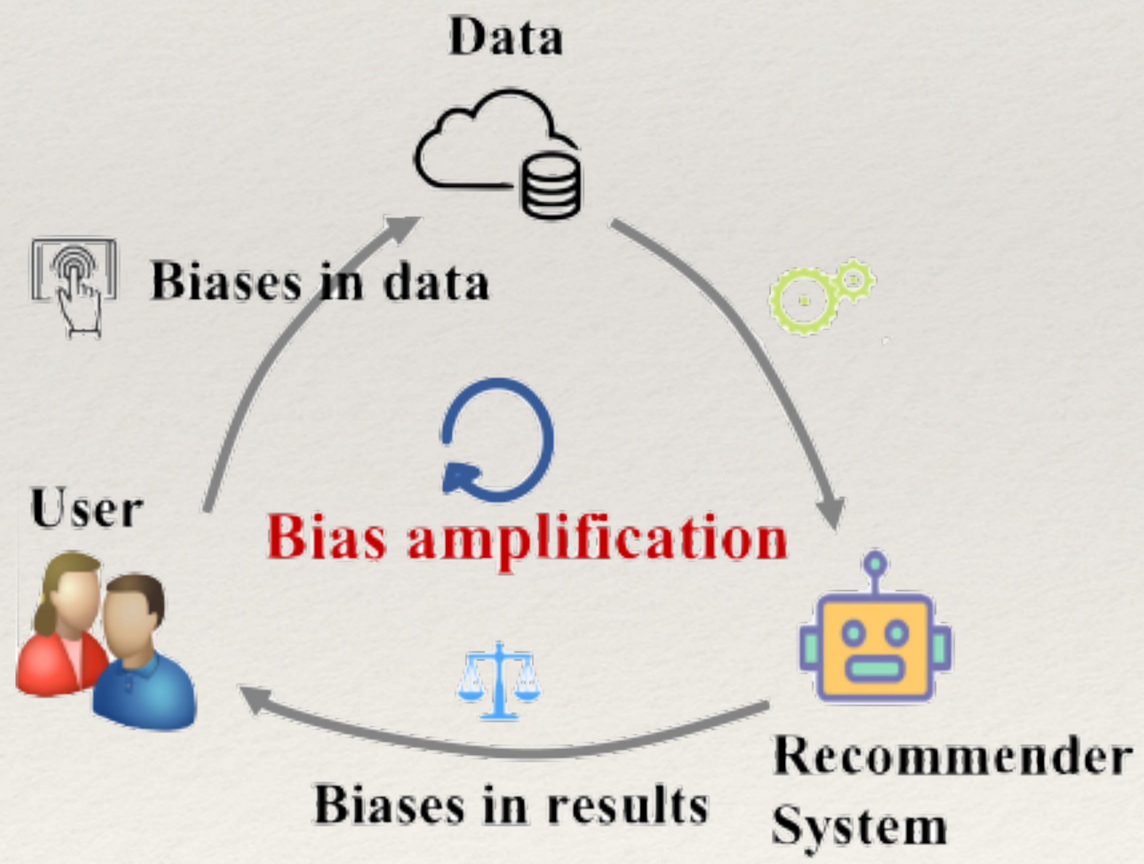


⇔



- Extremely sparse feedback
- Inherently bi-partite
- Long-tailed
- Missing-not-at-random



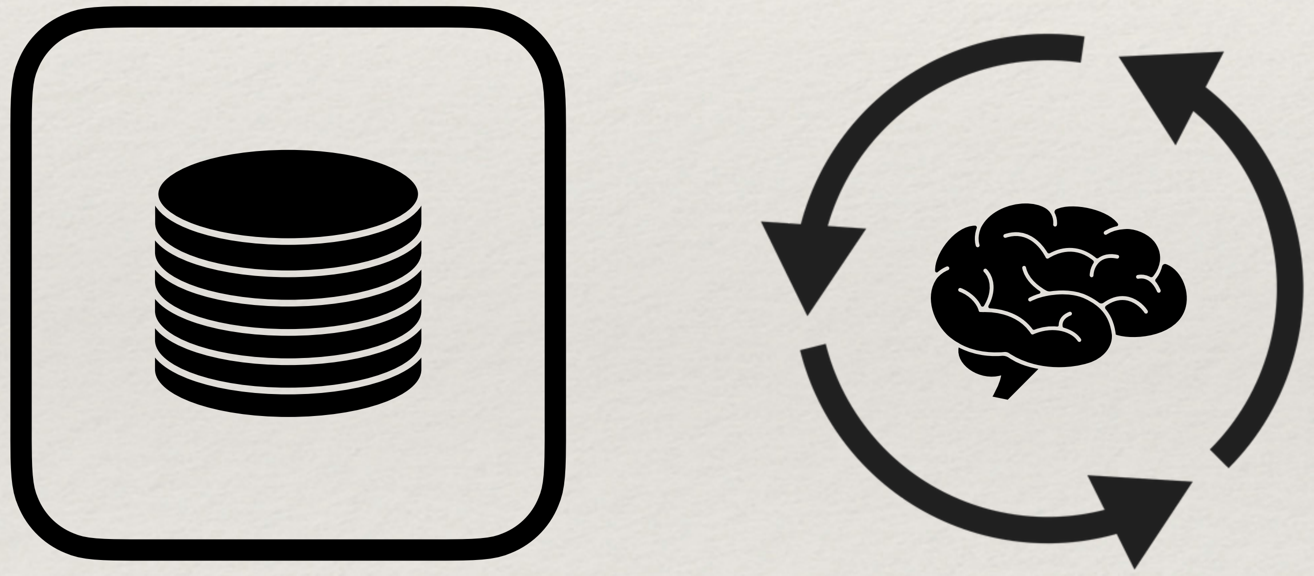


Premise

What is Data-Centric AI?

Model-Centric AI

Data | **Model**

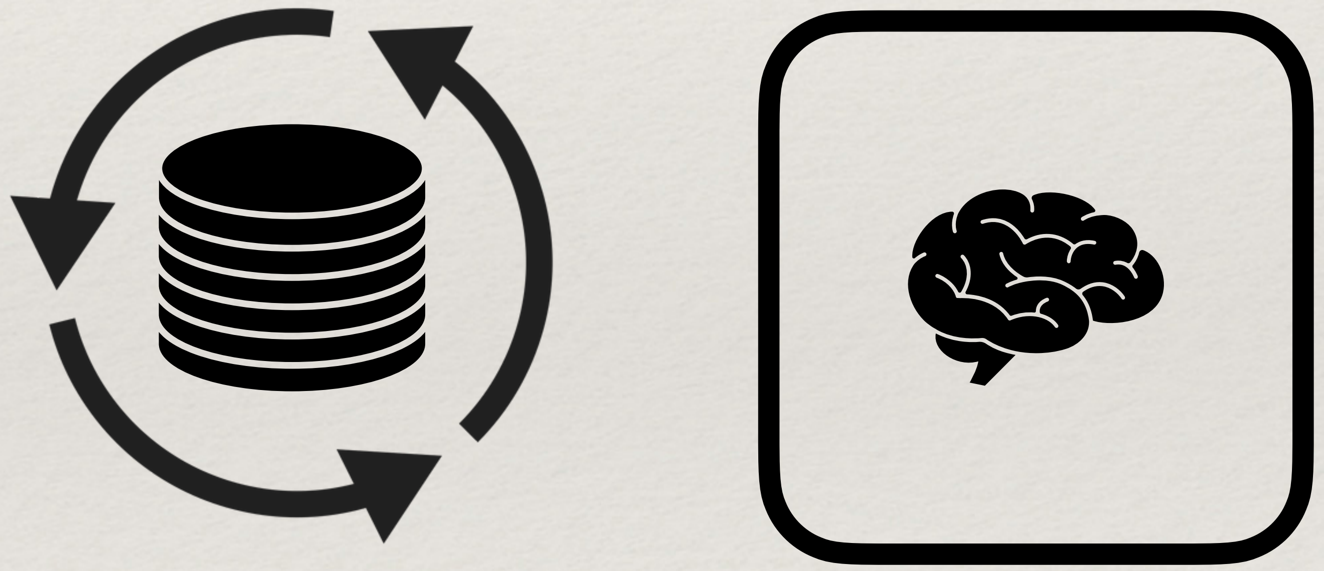


Freeze | *Improve*

- Well studied
- Expensive

Data-Centric AI

Data | **Model**

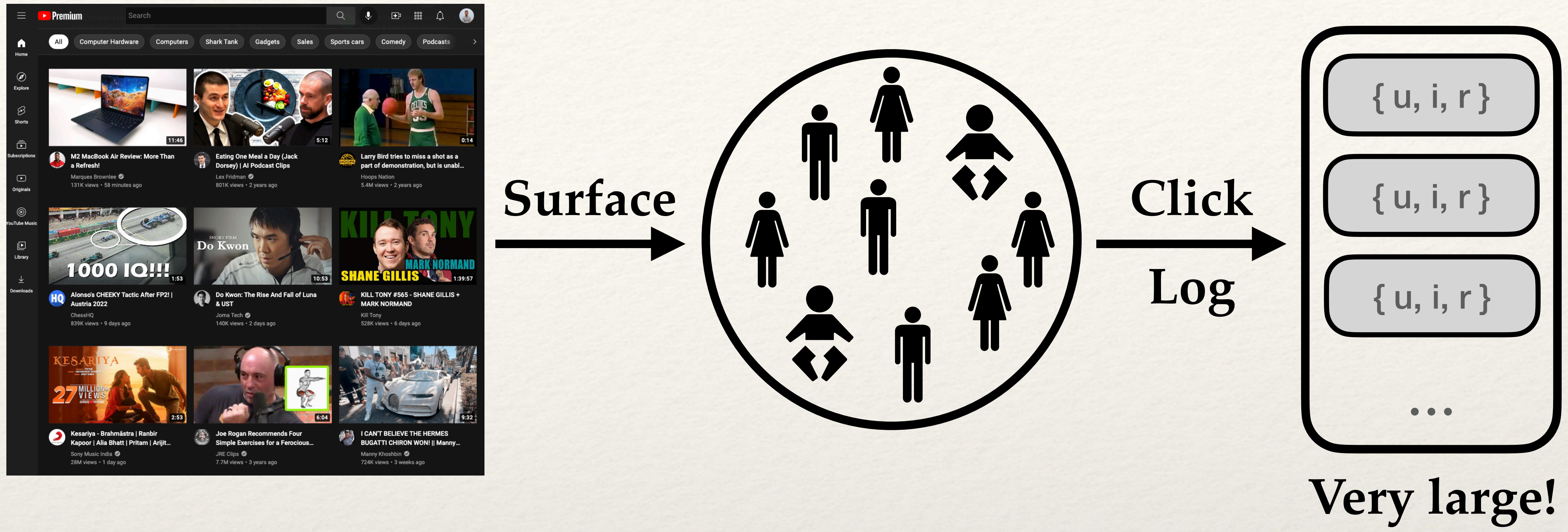


Improve | *Freeze*

- Under-studied
- Scalable

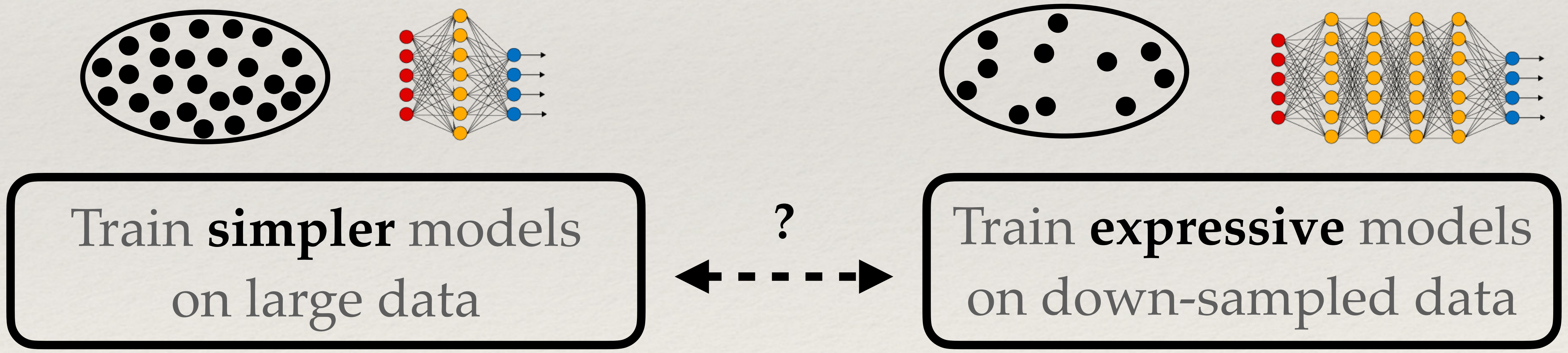
Premise

Why Data-Centric Recommender Systems?



- Unsupervised → large quantities of user-feedback

- Scaling-up systems by scaling-down data
  - Shift focus from data quantity → data "quality"
  - Dimension in performance : resources tradeoff
  - Savings in time, human-effort & environment degradation

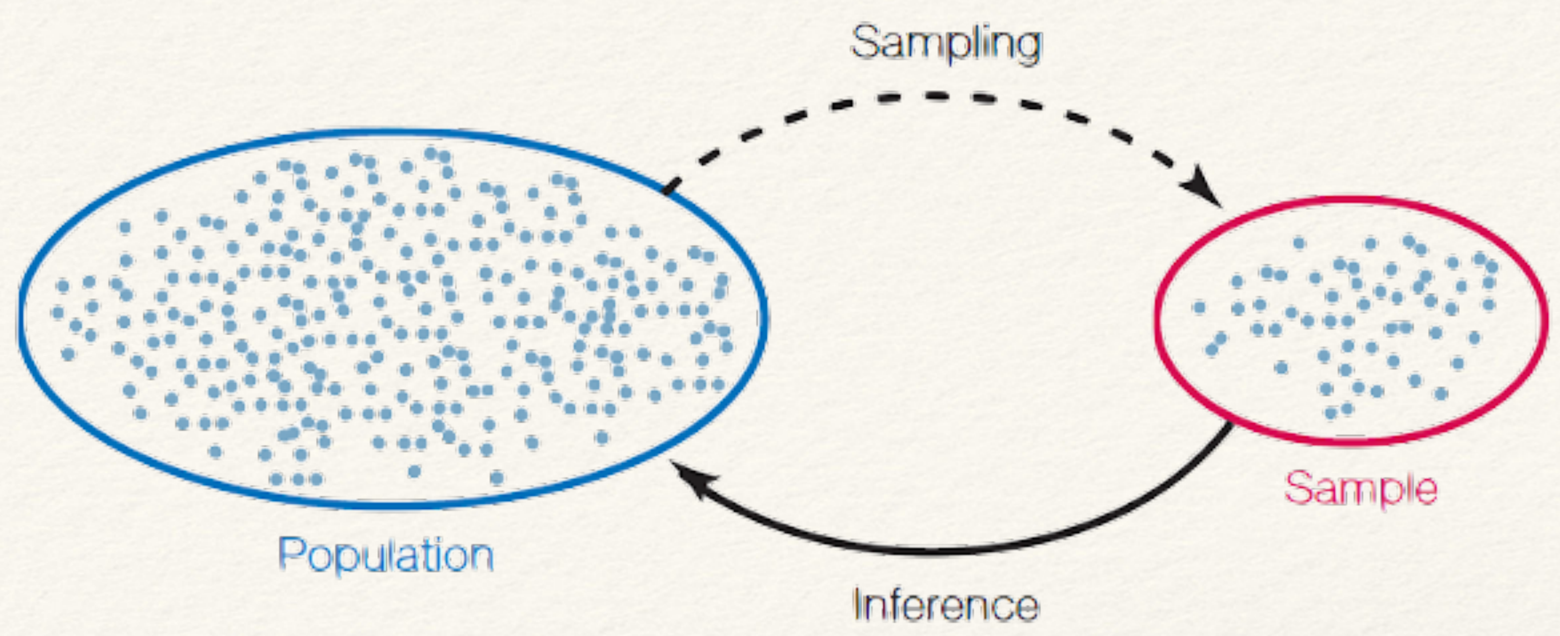


E.g. Linear modeling, Matrix Factorization, Item-item CF, etc.

E.g. Higher-order modeling, User-user CF, etc.

Scope

Scaling-up Systems by Scaling-down Data



Data down-sampling

Bias-sensitive

Generate a data sample which can guarantee **similar performance** of the same downstream model when trained on the full-dataset vs. data summary

Variance-sensitive

Generate a data sample which can accurately retain the **relative ordering** of different learning algorithms when trained on the full-dataset vs. data summary

Data Summary$_1$

$V(R_1) = 0.24$
$V(R_2) = 0.2$
$V(R_3) = 0.26$
$V(R_4) = 0.22$

?

Full Dataset

$V(R_1) = 0.2$
$V(R_2) = 0.22$
$V(R_3) = 0.24$
$V(R_4) = 0.26$

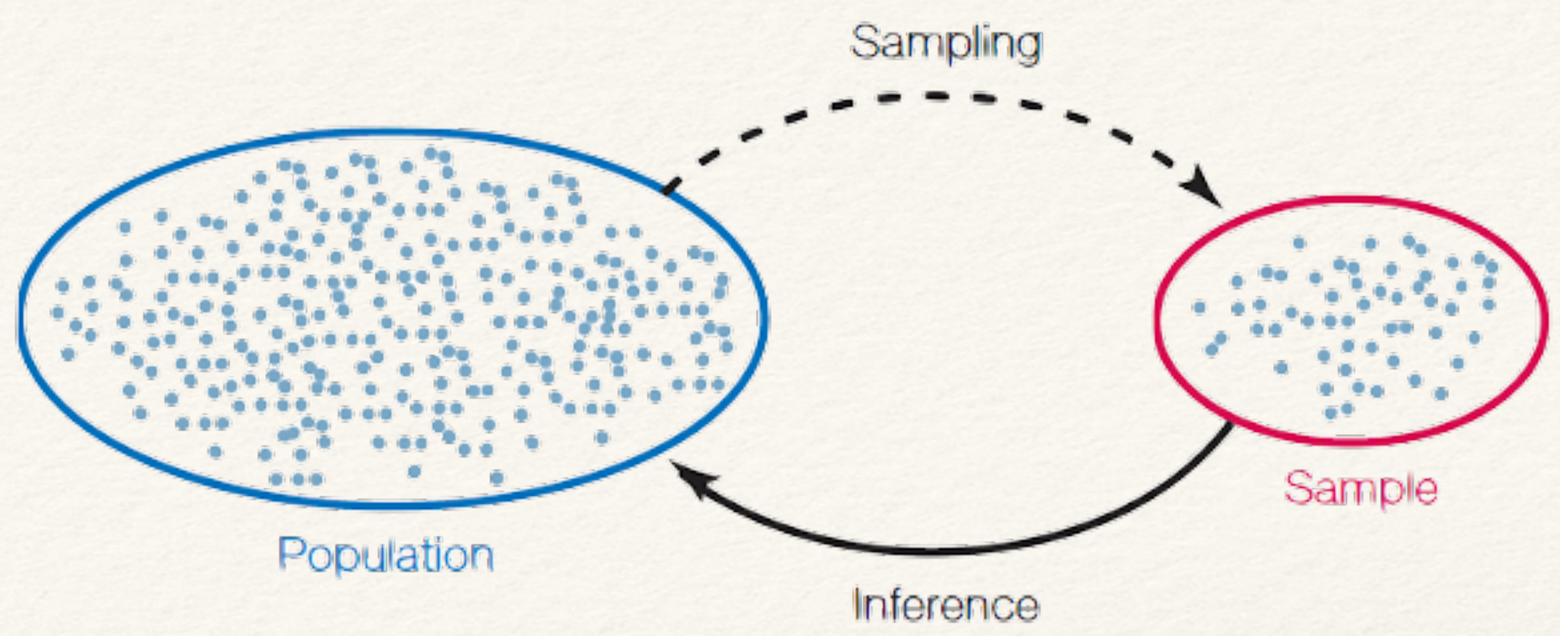
?

Data Summary$_2$

$V(R_1) = 0.3$
$V(R_2) = 0.32$
$V(R_3) = 0.34$
$V(R_4) = 0.36$

Scope

Scaling-up Systems by Scaling-down Data



Data down-sampling

Bias-sensitive

Generate a data sample which can guarantee **similar performance** of the same downstream model when trained on the full-dataset vs. data summary

- Direct deployment of models trained on data summary
- Faster research iterations
- Need modeling assumptions (at least for RecSys)

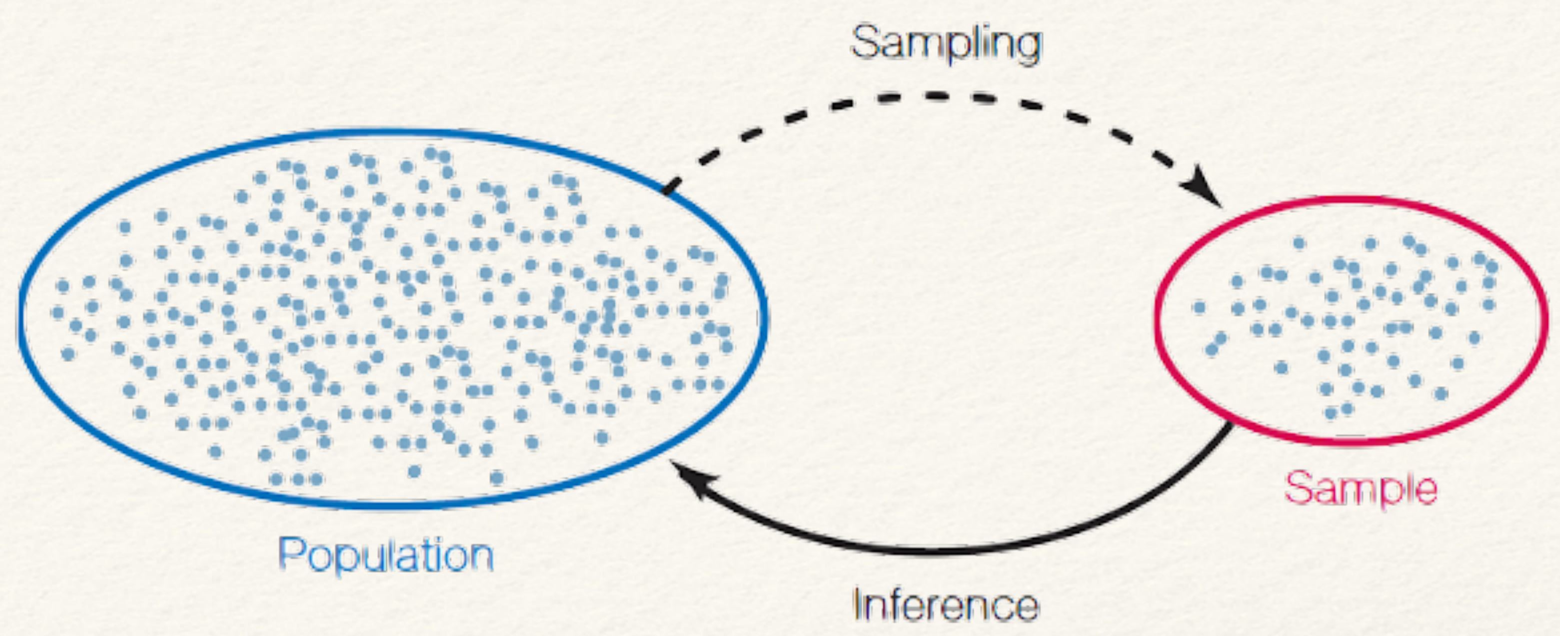
Variance-sensitive

Generate a data sample which can accurately retain the **relative ordering** of different learning algorithms when trained on the full-dataset vs. data summary

- Model search e.g. NAS, hyper-parameter optimization
- Offline model-to-model comparison
- No modeling assumptions

Scope

Scaling-up Systems by Scaling-down Data



Data down-sampling

Sampling

Pick the most informative subset of data-points

- Heuristics
  - Random, Head-user, Random-walks, Centrality…
- Coreset construction
  - Combinatorial optimization
- Expressivity limited by the collected data

Synthesis

Generate a set of fake and informative data-points

- Typically, treat the to-be-synthesized data as parameters, and learn them through gradient descent
- In addition to being useful, the synthesized data is fake — easy to share, release …
- Expressivity limited by the optimization procedure

SVP-CF

Selection-via-proxy for collaborative filtering data

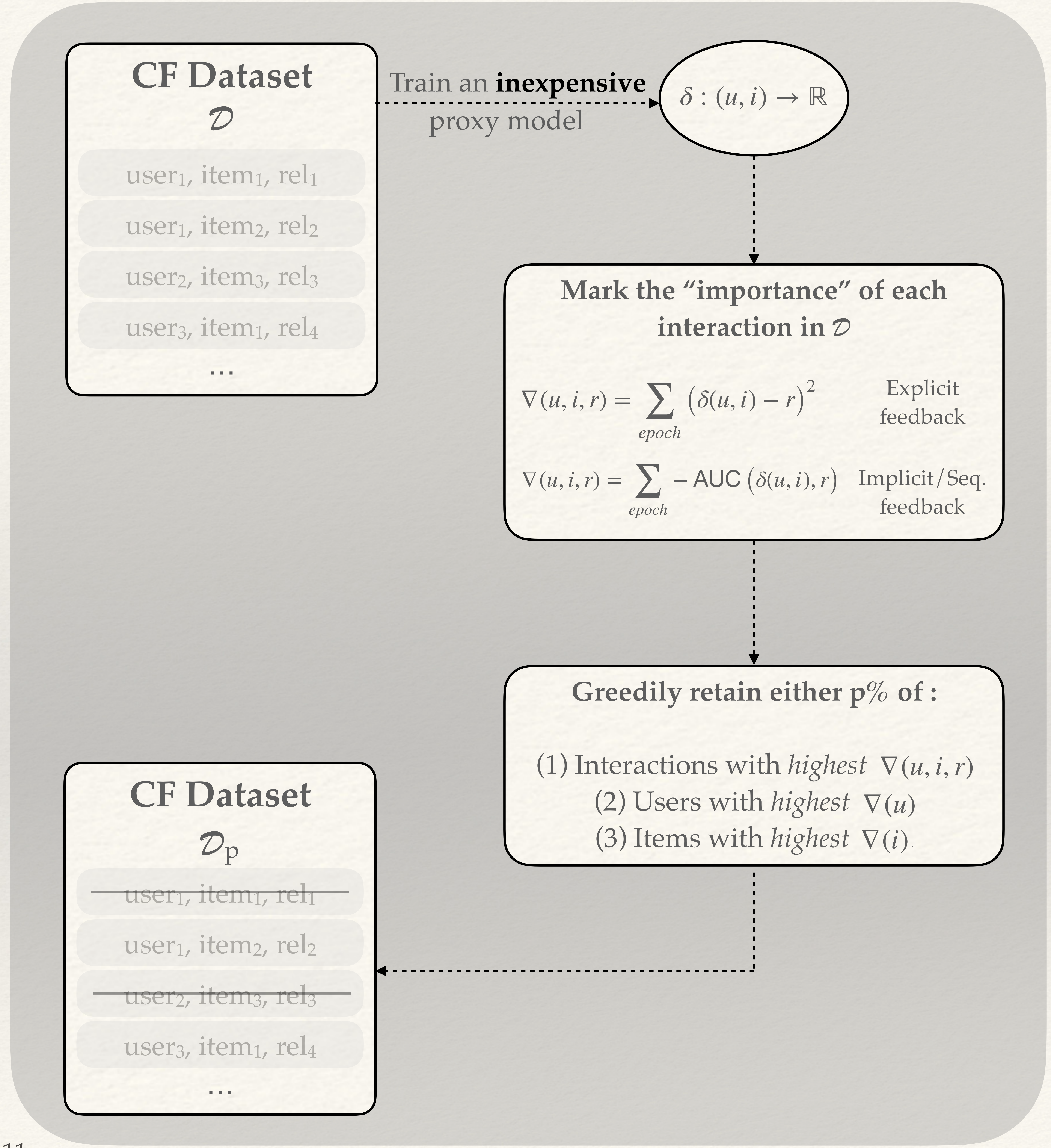
Premise: **Easy** parts of a dataset are most likely **easy** for all recommendation algorithms. Hence, removing such data is unlikely to change the relative ordering of algorithms.

SVP- CF

Selection-via-proxy for collaborative filtering data

Robust framework:

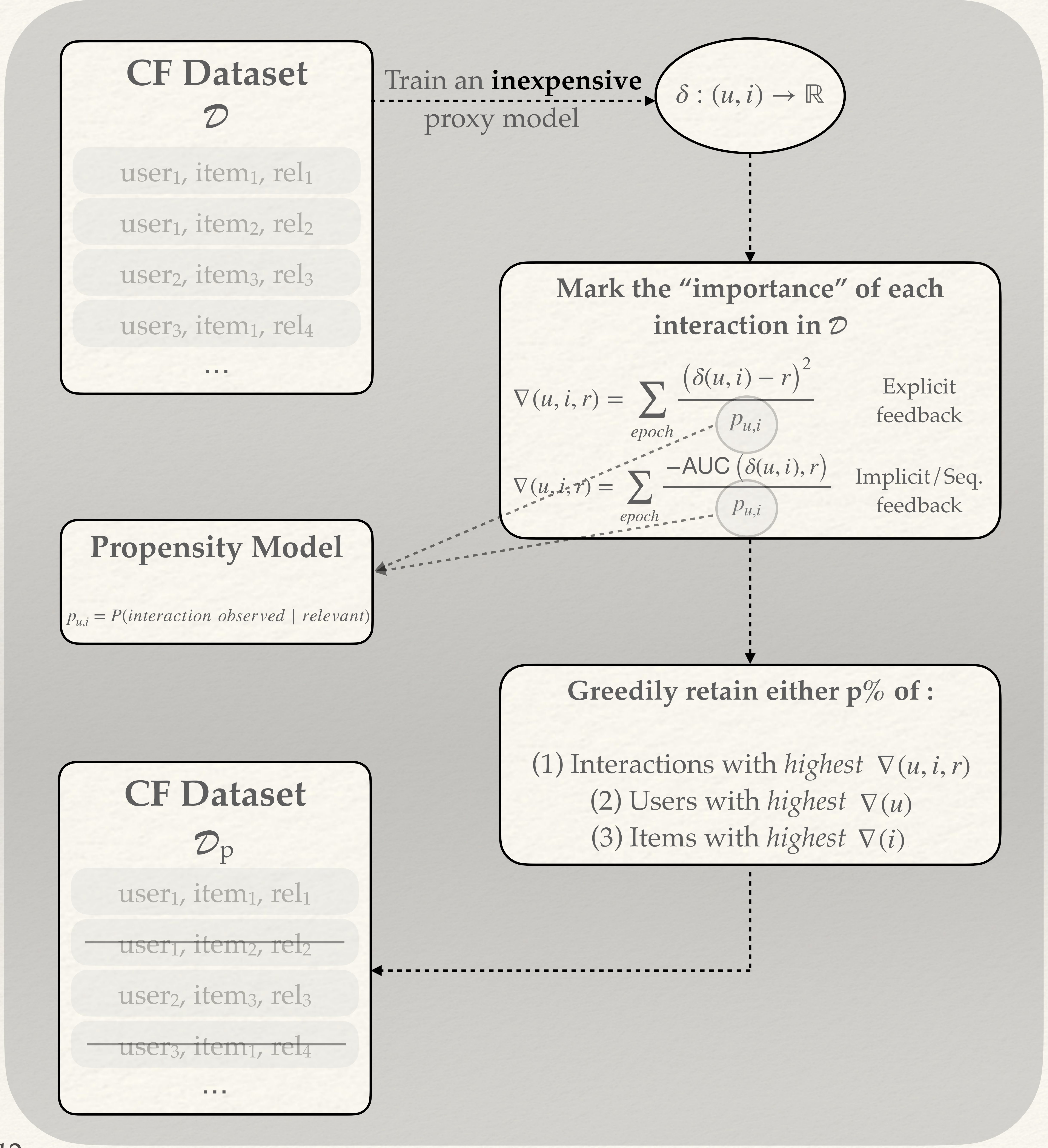
- Uses a proxy model to tag the **importance** of each interaction
- Efficiently handle multiple recommendation scenarios *e.g.* explicit, implicit, sequential, etc.
- Sample across varieties of data modalities: interactions, users, items, or even combinations of them



SVP- CF- Prop

Handling the missing-not-at-random characteristics

- Re-weigh the importance scores in SVP-CF using the probability of a user-item interaction going missing (propensity).
- Implicitly also handles the long-tail and data sparsity issues in user-item interaction data.



Data-Genie

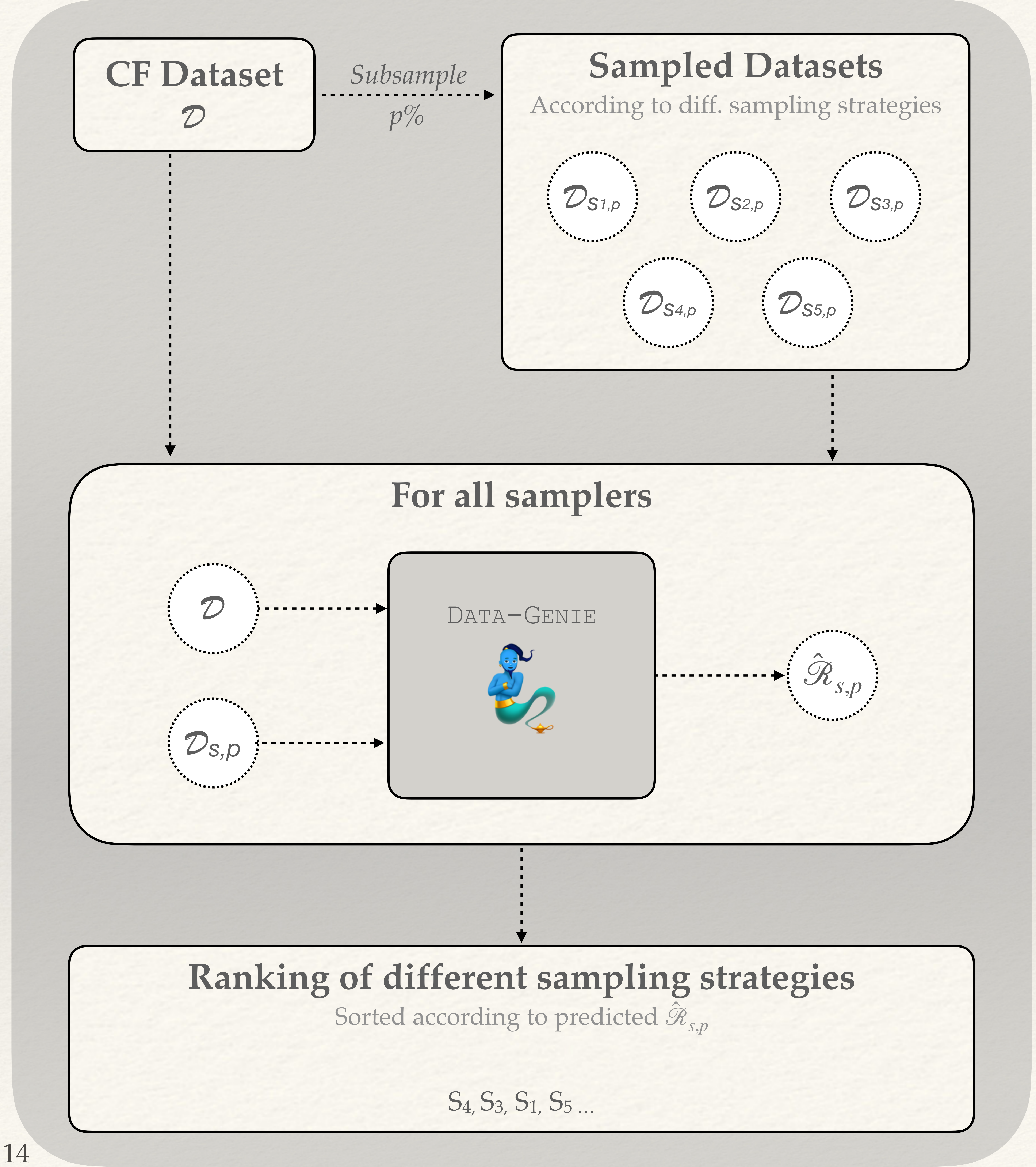
Which sampler is best for me?

Premise: Can we build an oracle-model which given (1) a dataset, (2) list of sampling strategies, and (3) a sampling budget, can **automatically predict** which sampling scheme would be the best?

🧞 DATA-GENIE

Which Sampler is best for me?

- Dynamically predicts the **performance** of a sampling strategy for any given CF-dataset.
- A trained DATA-GENIE model can transfer to **any** dataset, and can predict the utility of **any** sampling strategy.



DATA-GENIE

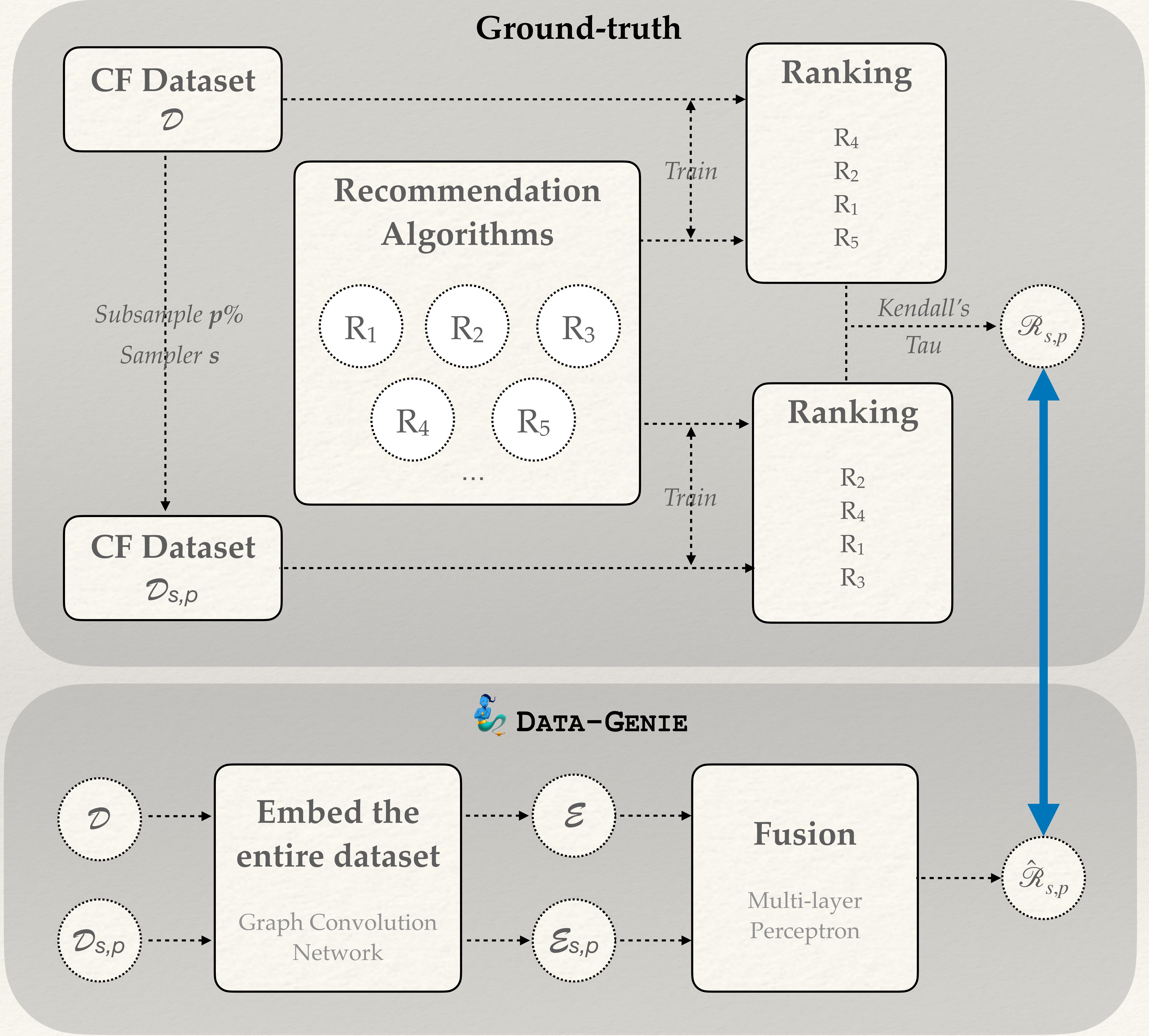
How is it trained?

- Circumvents the time-consuming process of training and benchmarking various algorithms.
- DATA-GENIE-regression:

$$\arg\min \sum_{\mathcal{D},\, s,\, p} \left(\mathcal{R}_{s,p} - \hat{\mathcal{R}}_{s,p}\right)^2$$

- DATA-GENIE-ranking:

$$\arg\min \sum_{\mathcal{D},\, p} \sum_{\mathcal{R}_{s_i,p} > \mathcal{R}_{s_j,p}} -\ln \sigma\left(\hat{\mathcal{R}}_{s_i,p} - \hat{\mathcal{R}}_{s_j,p}\right)$$



Experiments

Setup

	Sampling strategy
Interaction sampling	Random
	Stratified
	Temporal
	SVP-CF *w/* MF
	SVP-CF *w/* Bias-only
	SVP-CF-Prop *w/* MF
	SVP-CF-Prop *w/* Bias-only
User sampling	Random
	Head
	SVP-CF *w/* MF
	SVP-CF *w/* Bias-only
	SVP-CF-Prop *w/* MF
	SVP-CF-Prop *w/* Bias-only
Graph	Centrality
	Random-walk
	Forest-fire

Table 1: Sampling strategies used in our experiments

- 16 different sampling strategies
- 6 collaborative filtering datasets
- Explicit/Implicit/Sequential feedback for each CF-dataset
- 7 recommendation algorithms in our benchmarking suite
- A total of **400*k*** recommendation models trained! (~9 months of compute time!)

Experiments

Major Results

	Sampling strategy	*Average* **Kendall's Tau**
Interaction sampling	Random	0.407
	Stratified	0.343
	Temporal	0.405
	SVP-CF *w/* MF	0.484
	SVP-CF *w/* Bias-only	0.468
	SVP-CF-Prop *w/* MF	0.43
	SVP-CF-Prop *w/* Bias-only	0.458
User sampling	Random	0.431
	Head	0.19
	SVP-CF *w/* MF	0.344
	SVP-CF *w/* Bias-only	0.343
	SVP-CF-Prop *w/* MF	0.429
	SVP-CF-Prop *w/* Bias-only	0.445
Graph	Centrality	0.266
	Random-walk	0.396
	Forest-fire	0.382

Table 2: Average Kendall's Tau of various sampling strategies

- Widely used practice of making dense data subsets (e.g. Head-user, centrality) seem to be the worst ideas of all sampling strategies.
- SVP-CF significantly outperforms other samplers in retaining the ranking of different recommendation algorithms.

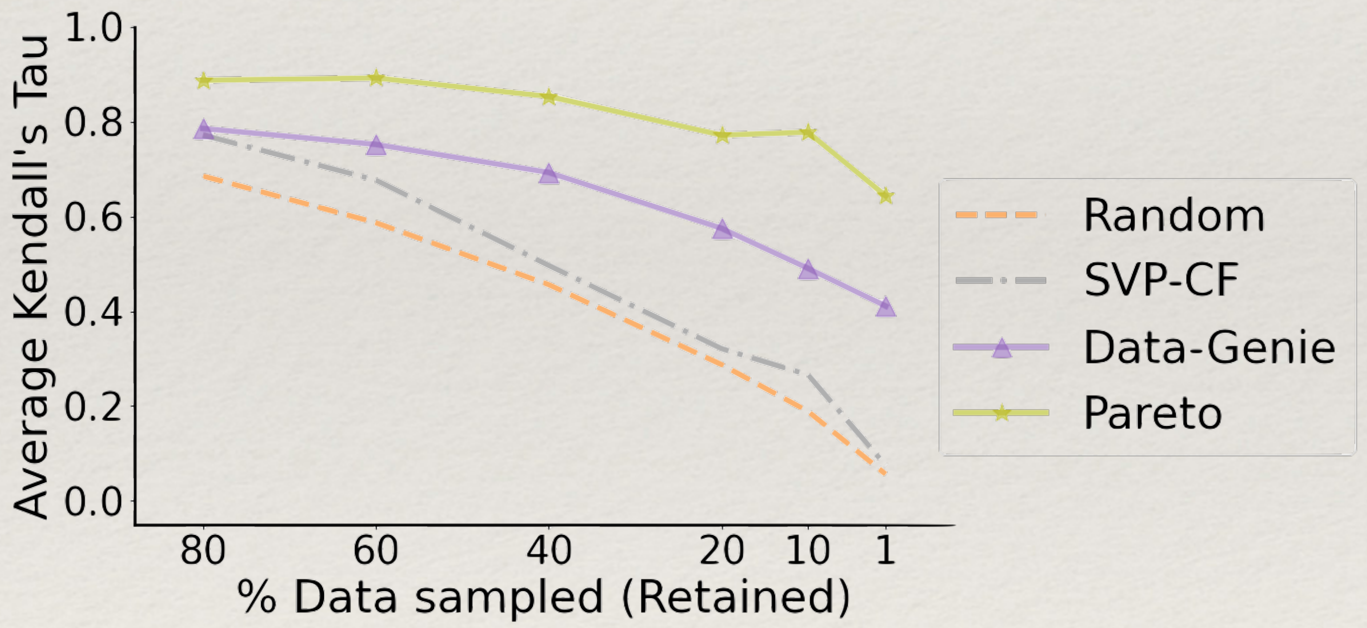


Figure 3: Does Data-Genie improve sampling performance with extreme sampling?

- Using SVP-CF, we can efficiently gauge the ranking of different algorithms with adequate confidence on **40-50%** data sub-samples, leading in an ~**2x** time speedup.
- Data-Genie enjoys the same level of performance with only **10%** of the original data, equating to ~**5.8x** time speedup!

∞-AE

Infinite-width AutoEncoder for Recommendation

Premise: Does stretching the bottleneck layer of an autoencoder till ∞ help in better recommendation?

∞-AE

Primer: Neural Tangent Kernel

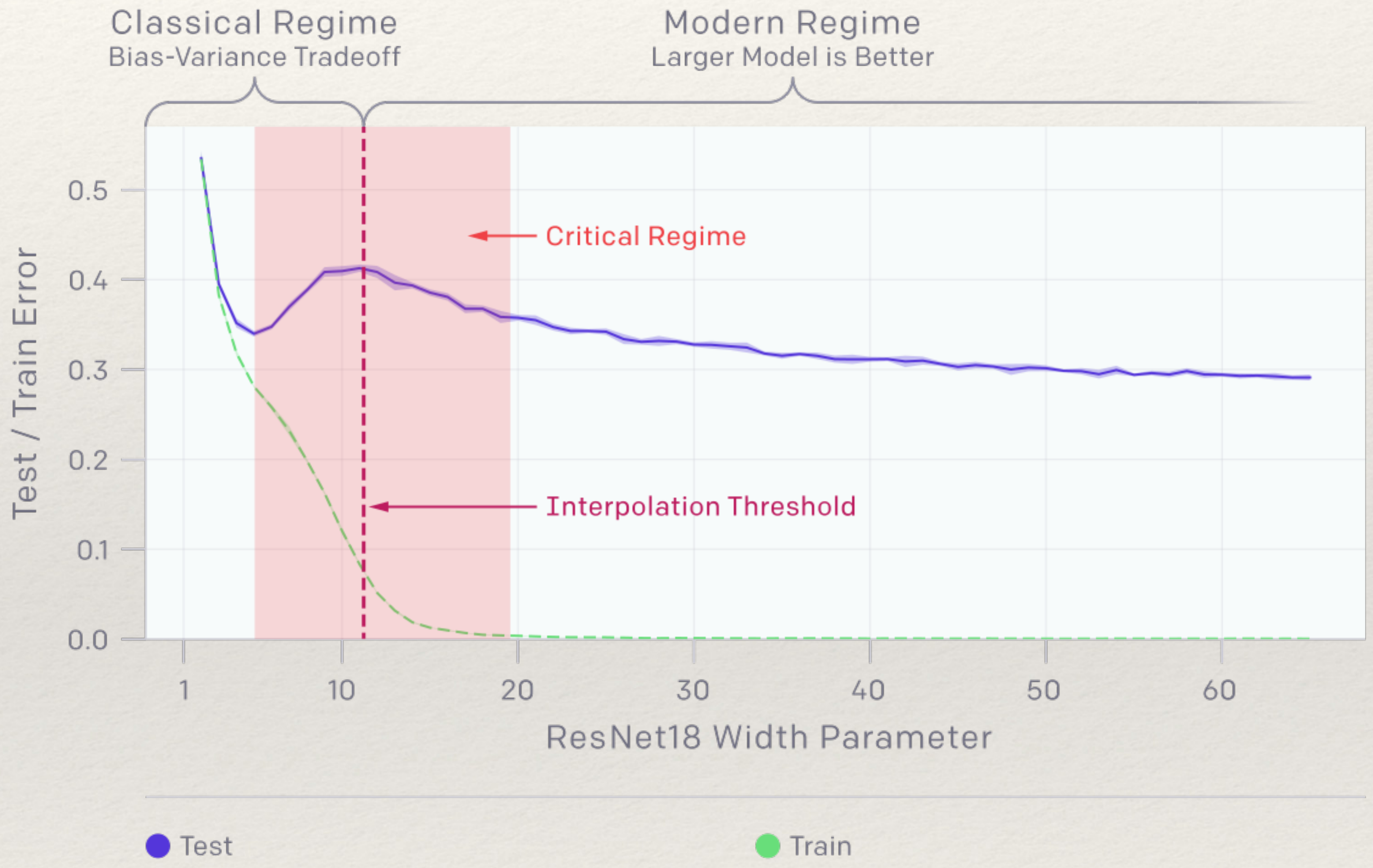
- **Infinite-width Correspondence:** Performing Kernelized Ridge Regression with the Neural Tangent Kernel (NTK) emulates the training of an infinite-width NN for an infinite number of SGD steps.

- For a given neural network architecture $f_\theta : \mathbb{R}^d \mapsto \mathbb{R}$, its corresponding NTK $\mathbb{K} : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ is given by:

$$\mathbb{K}(x, x') = \mathop{\mathbb{E}}_{\theta \sim W} \left[\left\langle \frac{\partial f_\theta(x)}{\partial \theta}, \frac{\partial f_\theta(x')}{\partial \theta} \right\rangle \right]$$

- Learning follows a double-descent phenomenon

- Finite-width counterparts empirically outperform NTK for standard image classification tasks



Credit: https://openai.com/blog/deep-double-descent/

∞-AE

Methodology

- X_u is the bag-of-items representation for user u i.e. all the items that u interacted with, and we aim to reconstruct it along with missing user preferences

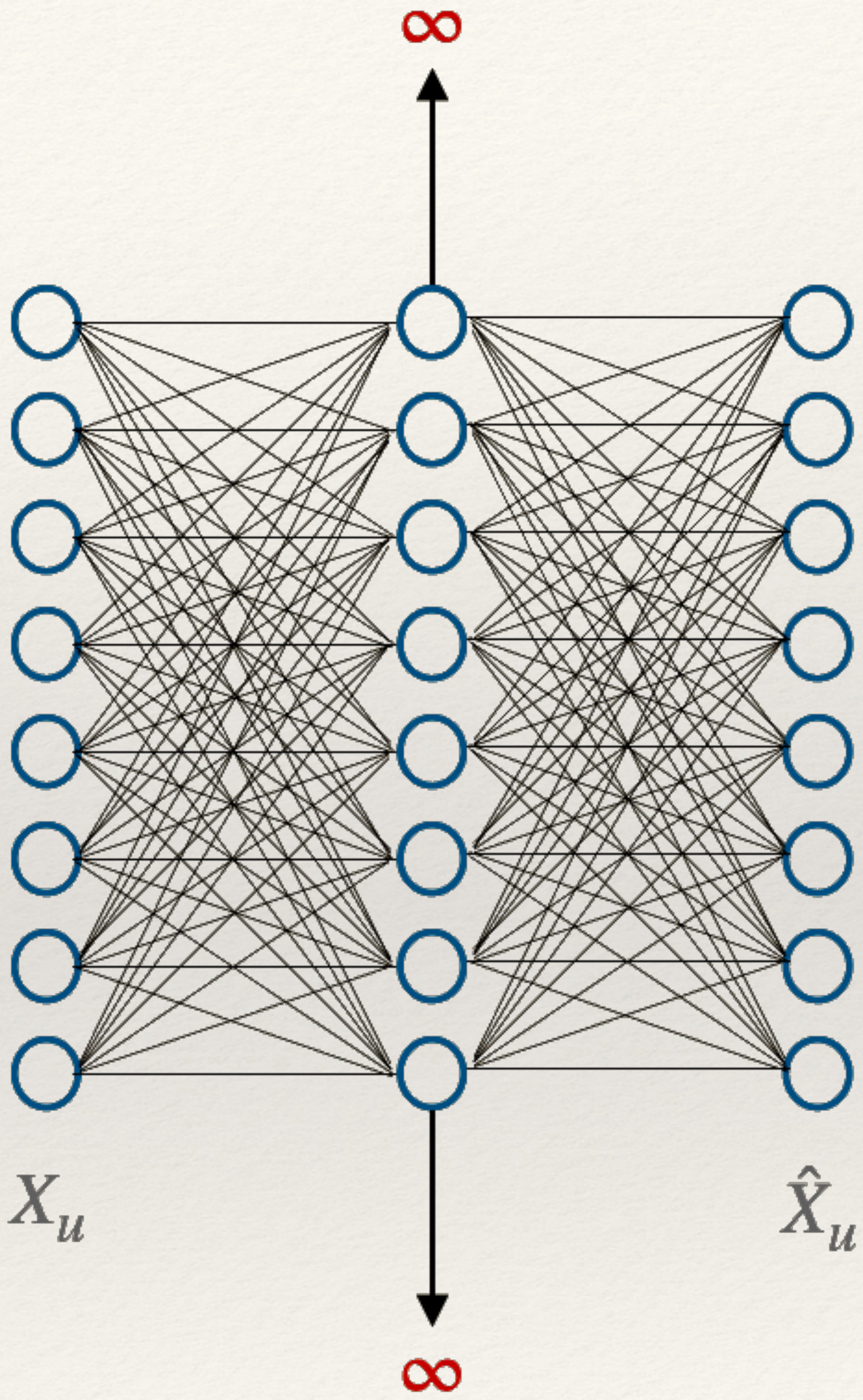
- Due to the infinite-width correspondence, ∞-AE optimizes in closed-form:

$$\hat{X} = K \cdot (K + \lambda I)^{-1} \cdot X \quad \text{s.t.} \quad K_{u,v} := \mathbb{K}(X_u, X_v) \quad \forall u, v$$

- The optimization has only a single hyper-parameter λ

- **Time complexity** Training: $\mathcal{O}(U^2 \cdot I + U^{2.376})$ Inference: $\mathcal{O}(U \cdot I)$

- **Memory complexity** Training: $\mathcal{O}(U \cdot I + U^2)$ Inference: $\mathcal{O}(U \cdot I)$



∞-AE

Experiments

Dataset	NeuMF	GCN	MVAE	EASE	∞-AE
Magazine	13.6	22.5	12.1	22.8	**23.0**
ML-1M	25.6	28.8	22.1	29.8	**32.8**
Douban	13.3	16.6	16.1	19.4	**24.9**
Netflix	12.0	—	20.8	26.8	**30.5***

Table 5: nDCG@10 performance (higher is better) of various recommendation algorithms.
* represents training on 5% random users.

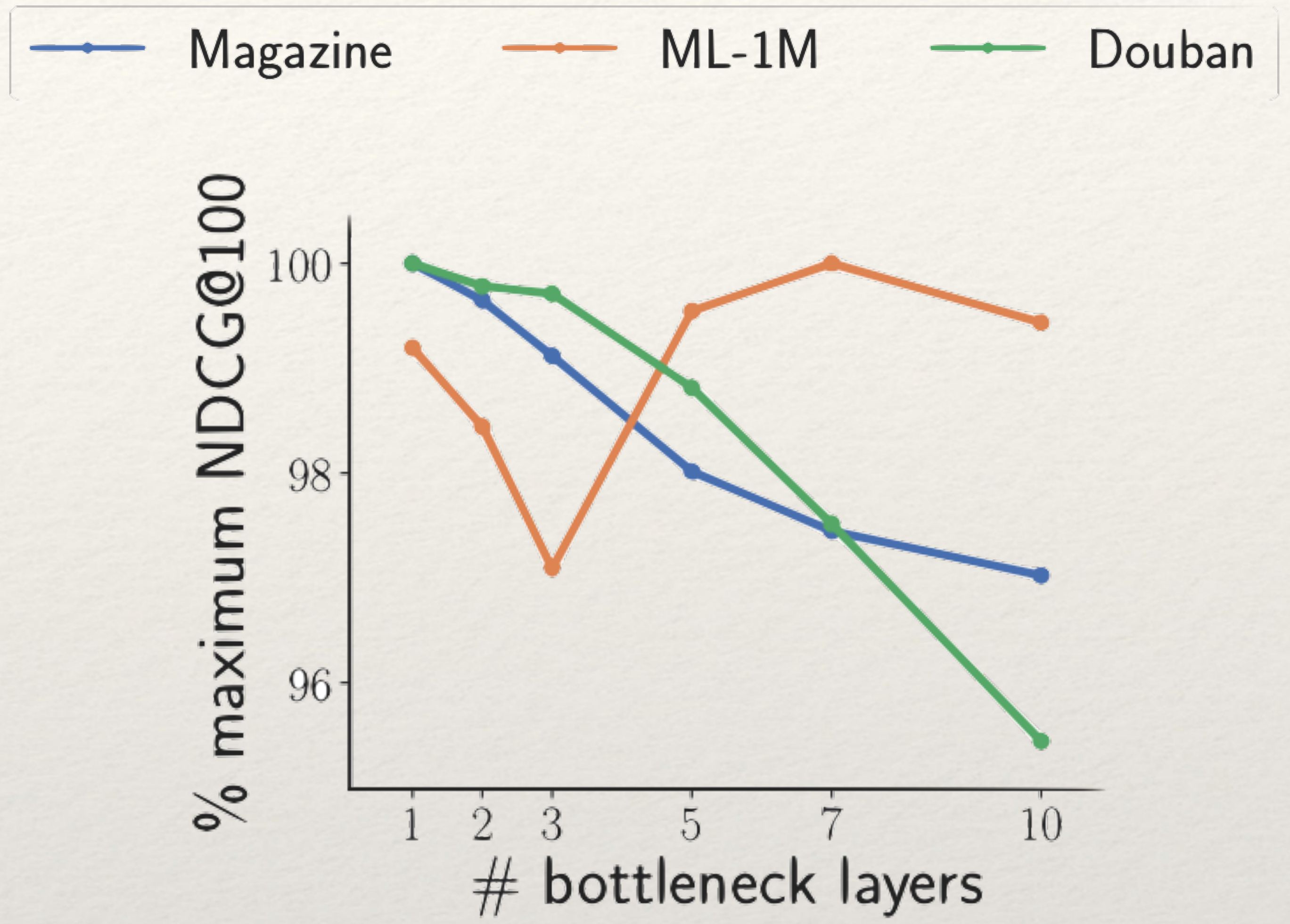


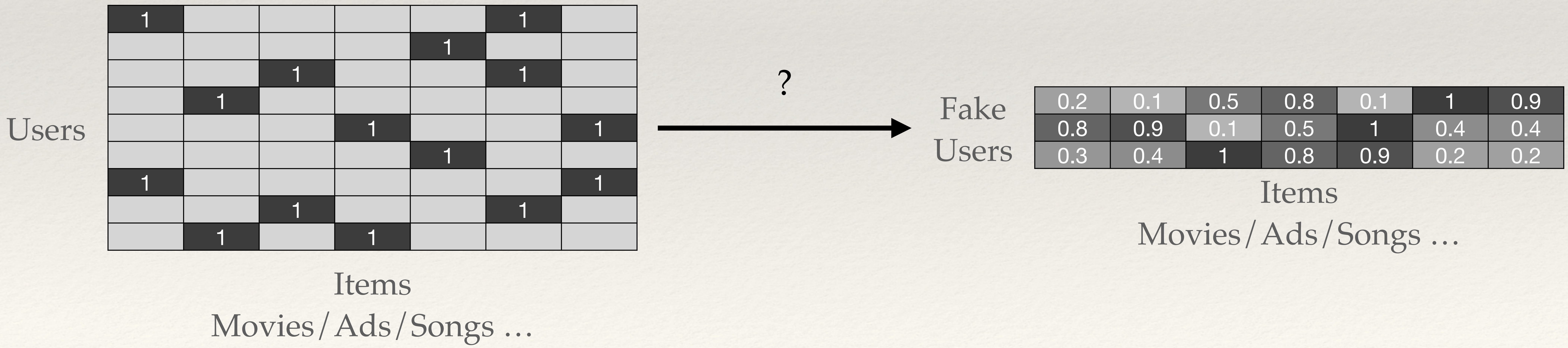
Figure 6: Performance of ∞-AE with varying depth.

- ∞-AE outperforms various state-of-the-art methods, even when trained on just 5% random users
- 1 layer seems to be enough for optimal recommendation performance: common folk-knowledge
- Even though the model is expensive; it is simplistic, easy to implement (thanks, `JAX`), and the performance is great! But how to scale it up? 🤔

Distill-CF

Data Distillation for Collaborative Filtering Data

Premise: Treat the to-be-synthesized data as **parameters**, and **learn** them through a bilevel optimization.



Distill-CF

Overview & Challenges

Challenges:

- D^s consists of **discrete** (u, i, r) tuples
- **Semi-structuredness**: some users/items are more popular than others
- D^s is typically extremely **sparse**

Outer loop — optimize the support set for a fixed learning algorithm

Support dataset

Differentiable cost-function

$$\underset{\boxed{\mathcal{D}^s}}{\arg\min} \; \mathbb{E}_{(u,i,r)\sim\mathcal{D}^{\text{val}}} \left[\boxed{l}\left(\phi^*_{\mathcal{D}^s}(u,i),r\right)\right]$$

$$\text{s.t.} \quad \boxed{\phi^*_{\mathcal{D}^s}} := \underset{\phi}{\arg\min} \; \mathbb{E}_{(u,i,r)\sim\mathcal{D}^s} \left[l\left(\phi(u,i),r\right)\right]$$

Optimal recommendation algorithm trained on $\mathcal{D}^s$

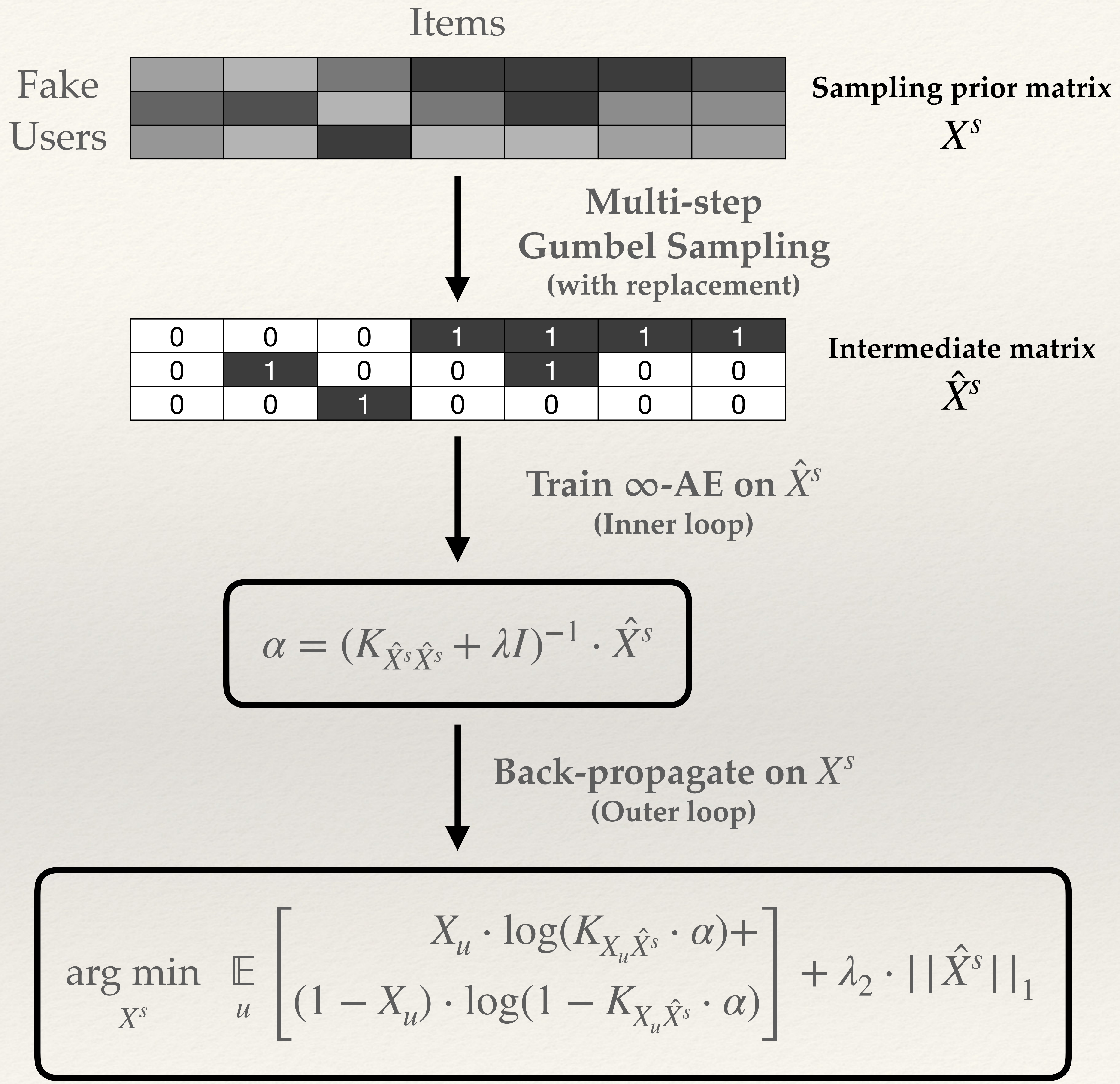
Inner loop — optimize the learning algorithm for a fixed support set

Distill-CF

Methodology

Robust framework:

- Uses Gumbel sampling on X^s to mitigate the heterogeneity of the problem
- Perform Gumbel sampling multiple times for each fake-user to handle dynamic user/item popularity
- Automatically control sparsity in $\hat{X}^s$ by controlling the entropy in X^s
- **Optimizes** for data-quality rather than quantity



Distill-CF

Experiments

- Using Distill-CF, we can get **96-105%** of full-data performance on as small as **0.1%** data sub-samples, leading to as much as **~1000x** time speedup!
- Distill-CF works well even for the second-best EASE model, even though data isn't optimized for it

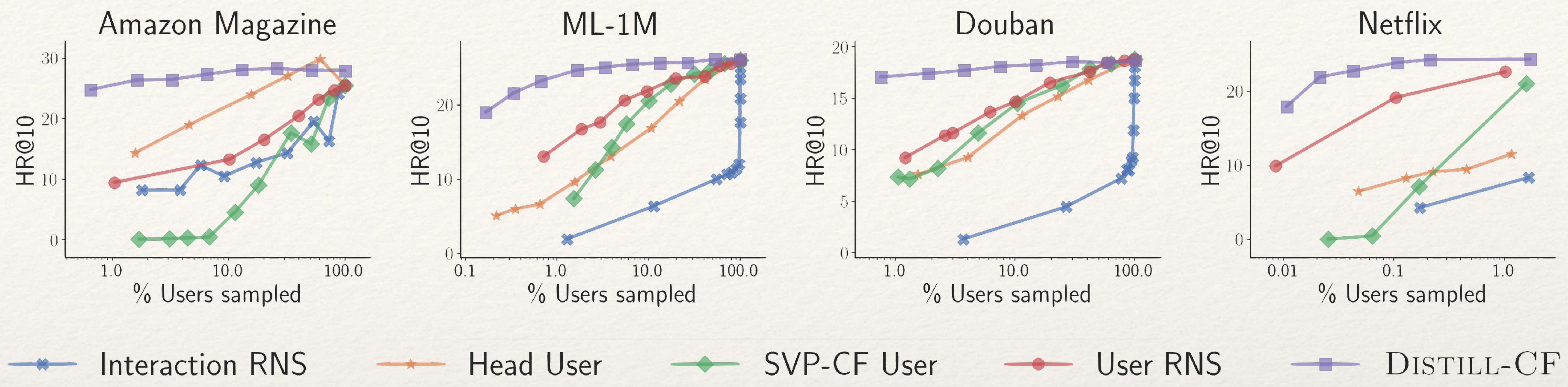


Figure 7: Does Distill-CF outperform other samplers? (Log-scale)

Dataset	NeuMF	GCN	MVAE	EASE	∞-AE	∞-AE (Distill-CF)
Magazine	13.6	22.5	12.1	22.8	23.0	**23.8**
ML-1M	25.6	28.8	22.1	29.8	**32.8**	32.5
Douban	13.3	16.6	16.1	19.4	**24.9**	24.2
Netflix	12.0	—	20.8	26.8	**30.5***	**30.5**

Table 8: nDCG@10 performance of various recommendation algorithms. * represents training on 5% random users. Distill-CF has a user budget of just 500 (0.1% for Netflix).

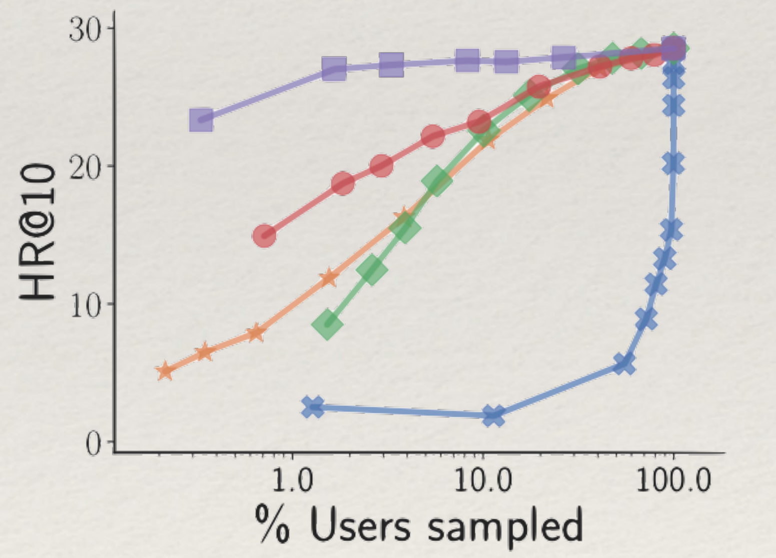


Figure 9: Distill-CF + EASE for the ML-1M dataset.

Distill-CF

Experiments (Contd.)

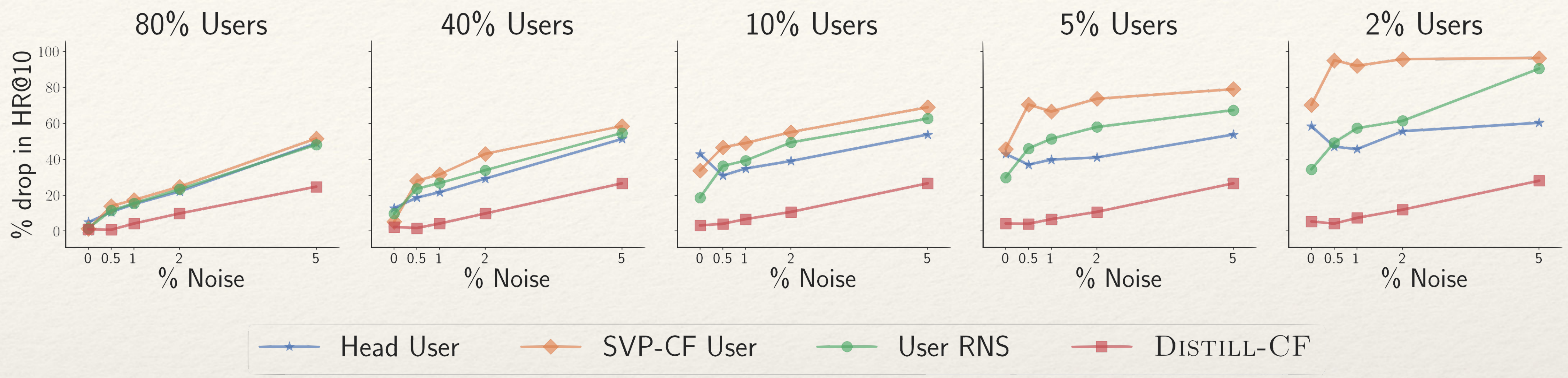


Figure 10: Performance of different samplers when there is noise in the original data.

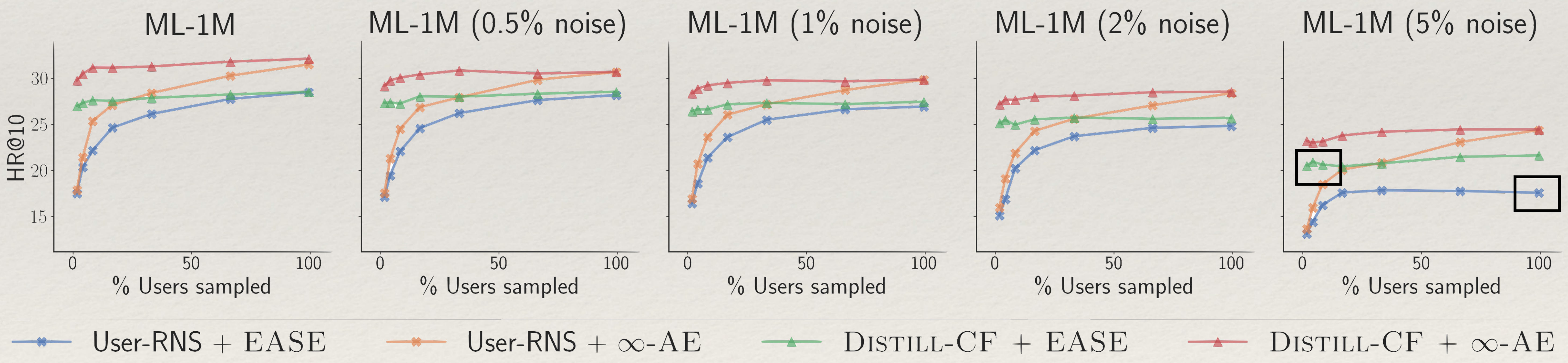


Figure 11: Performance comparison of ∞-AE *vs.* EASE when trained on down-sampled, noisy data.

- Distill-CF is **robust to noise** (even though not optimized for it), and is able to offer significant performance even at high noise ratios and very small support datasets!
- **Less is more**: EASE is more accurate when trained on lesser amounts of data generated by Distill-CF, compared to training on the full-data

Future Directions

Extensions

∞ Recommendation Networks

- Making it more scalable — sparse kernel computations
- More applications — search, XC, …
- Extending to sequential recommendation

Ranksets

- Formalize the notion of variance-sensitive sampling
- 🧞 DATA-GENIE is still a two step-process. How to optimize for a rankset?

Fairness & Privacy

- How to optimize for these while sampling/distilling
- Guaranteeing data privacy in distills, such that de-anonymization is impossible

Applications

- Continual Learning — catastrophic forgetting
- NAS, Hyper-parameter Optimization

Acknowledgement

Advisor

Julian McAuley
UC San Diego

Collaborators

Microsoft Research

Cornell University

International Institute of Information Technology Hyderabad

Netflix Research

Meta

References

[1] **On Sampling Collaborative Filtering Datasets**. *Sachdeva, Wu, McAuley*. In WSDM '22.

[2] **Infinite Recommendation Networks: A Data-Centric Approach**. *Sachdeva, Dhaliwal, Wu, McAuley*. arXiv '22.

Thank you!
Questions?

@noveens97

For papers, code, and these slides:

`noveens.com`