

Personalizing Software and Web
Services by Leveraging

Unstructured Application Usage Traces



Longqi Yang*, Chen Fang⁺, Hailin Jin⁺, Matt Hoffman⁺, Deborah Estrin*

*Computer Science, Cornell Tech
Cornell University

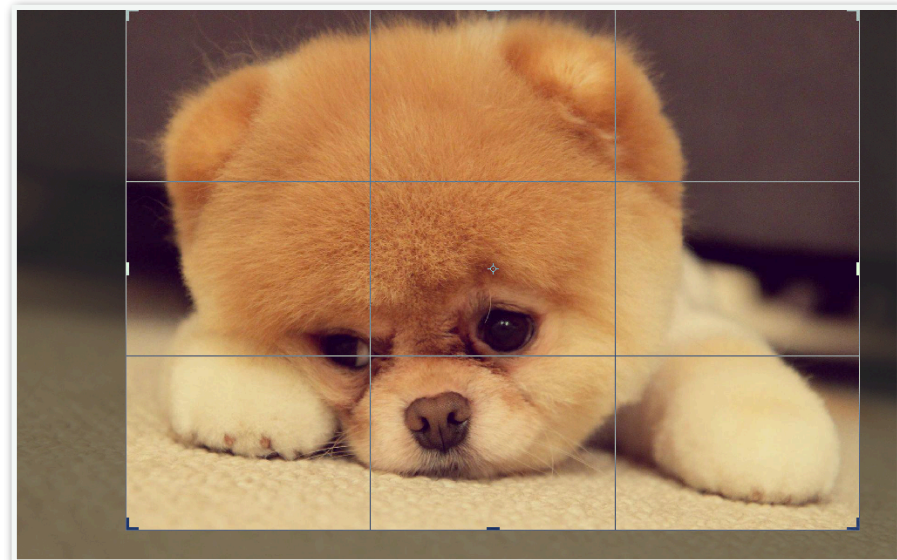
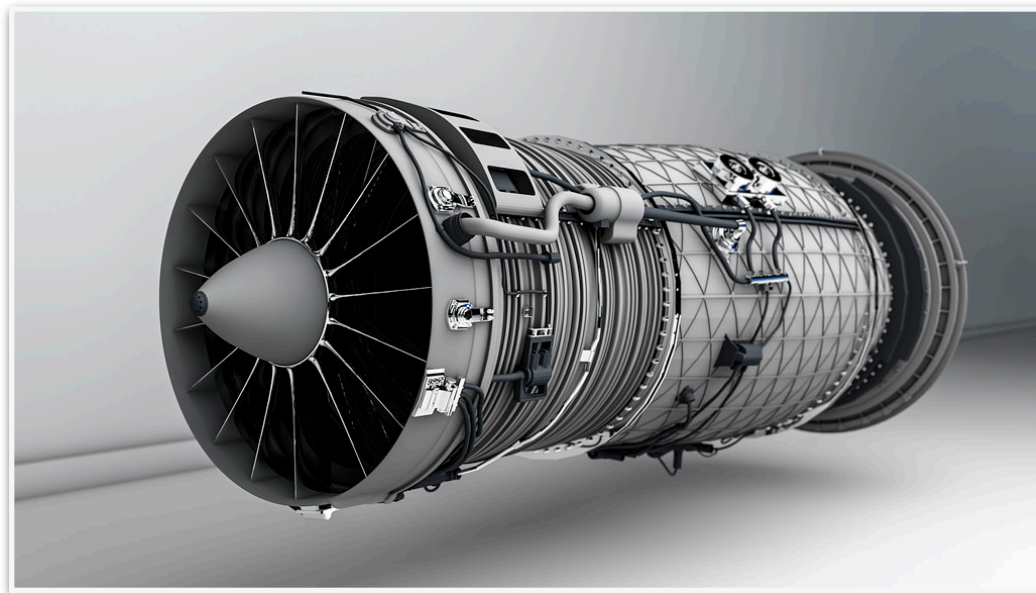
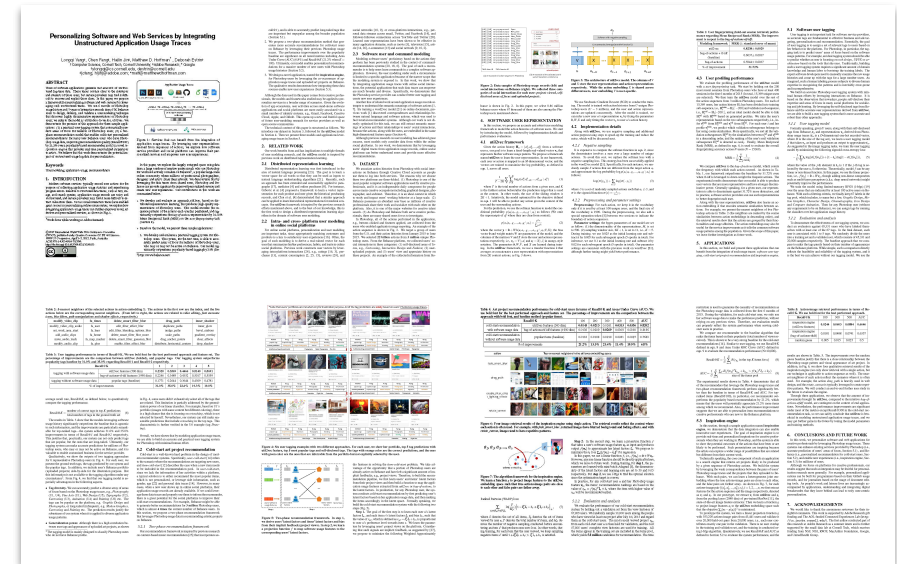
⁺Adobe Research

Email: ylongqi@cs.cornell.edu

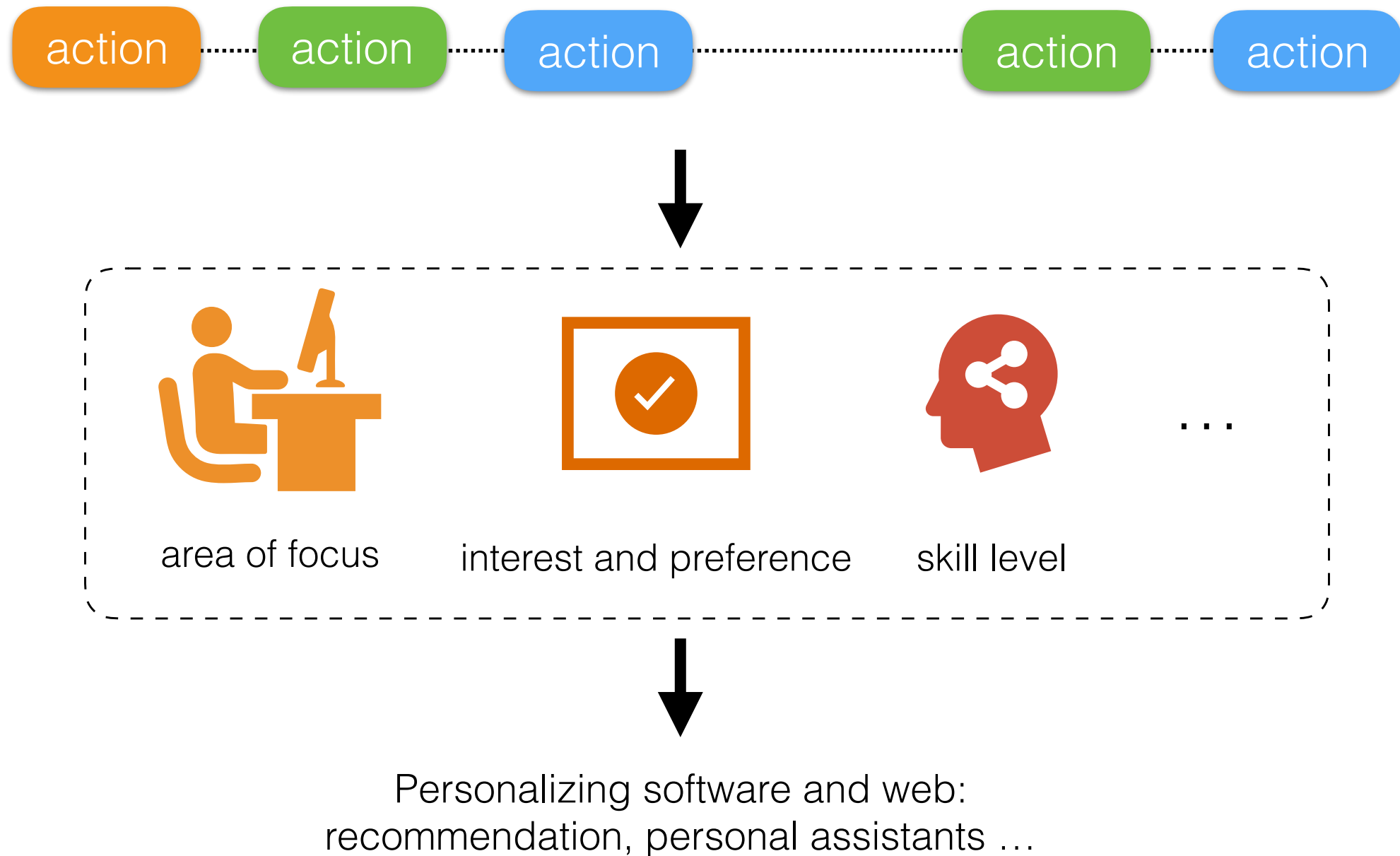
Day to day work activities are increasingly dependent on digital applications and services

```
/*
 * You may think you know what the following code does.
 * But you dont. Trust me.
 */

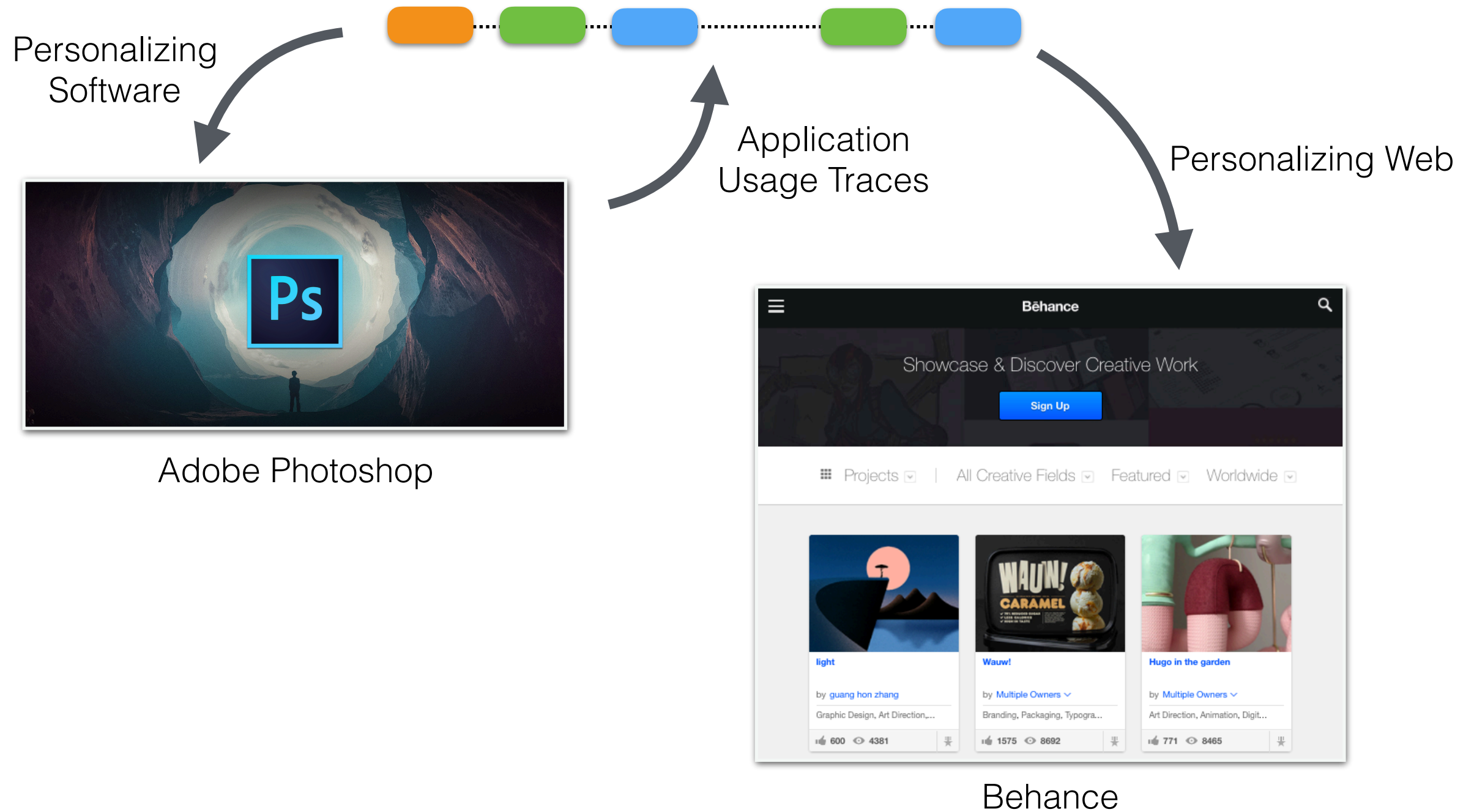
unsigned int reverse(register unsigned int x)
{
    x = (((x & 0xaaaaaaaa) >> 1) | ((x & 0x55555555) << 1));
    x = (((x & 0xcccccccc) >> 2) | ((x & 0x33333333) << 2));
    x = (((x & 0xf0f0f0f0) >> 4) | ((x & 0x0f0f0f0f) << 4));
    x = (((x & 0xff00ff00) >> 8) | ((x & 0x00ff00ff) << 8));
    return((x >> 16) | (x << 16));
}
```



Potentials of application usage traces

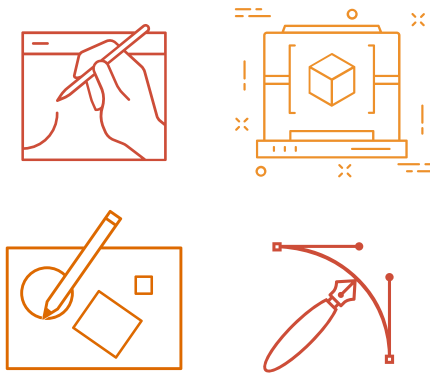


Use cases studied in this work



Outline

Part II



Personalizing software

Part III

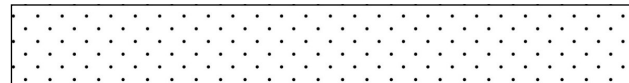


Personalizing web



inspiration engine

Part I



utilization-to-vector (**util2vec**)

action

action

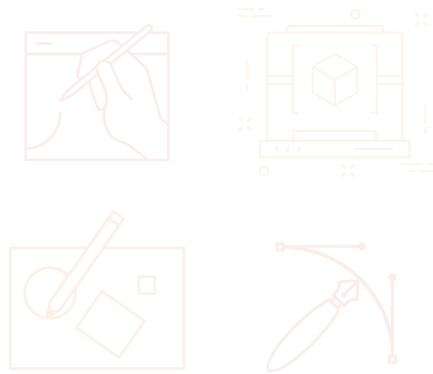
action

action

action

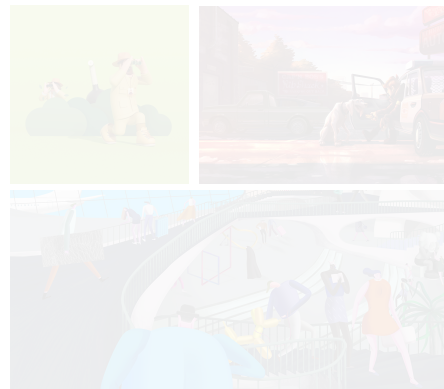
Part I. utilization-to-vector (util2vec)

Part II



Personalizing software

Part III

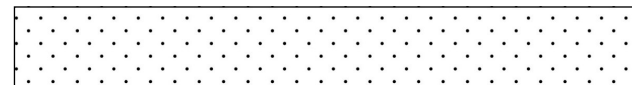


Personalizing web



inspiration engine

Part I



utilization-to-vector (**util2vec**)

action

action

action

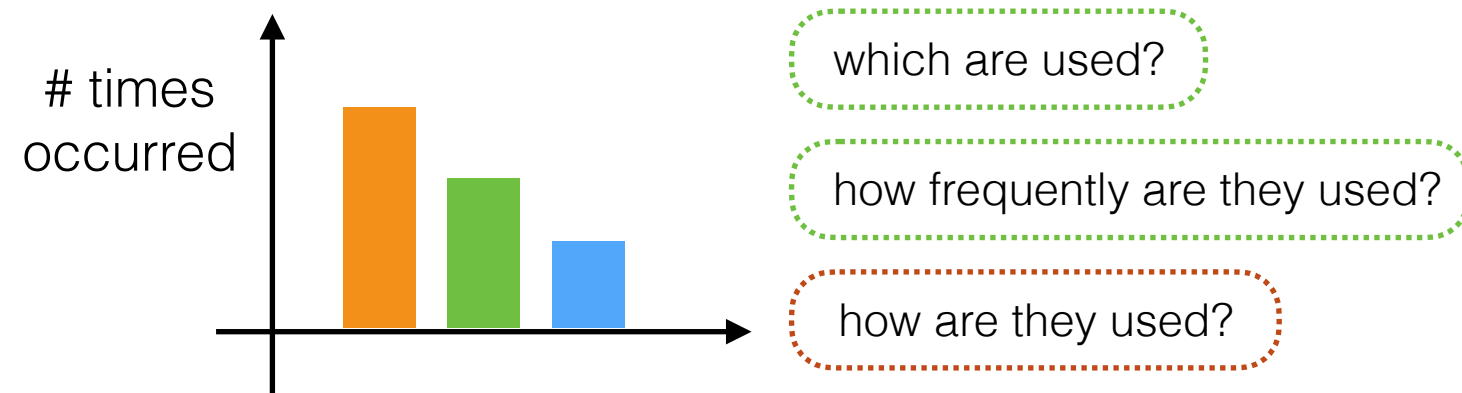
action

action

Software user representation learning

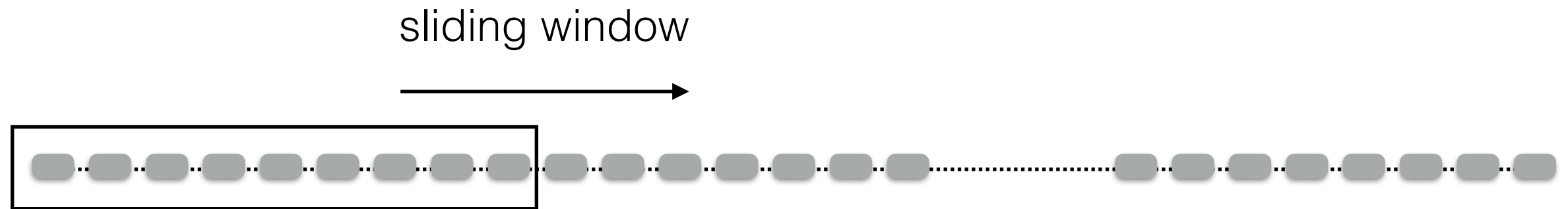


An intuitive approach: **Bag-of-Actions**



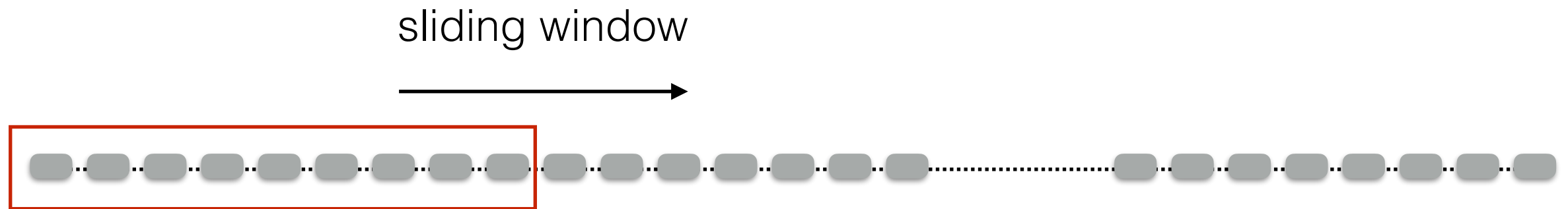
utilization-to-vector (**util2vec**)

sliding window



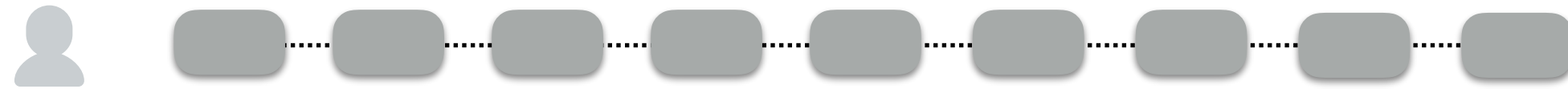
utilization-to-vector (**util2vec**)

sliding window

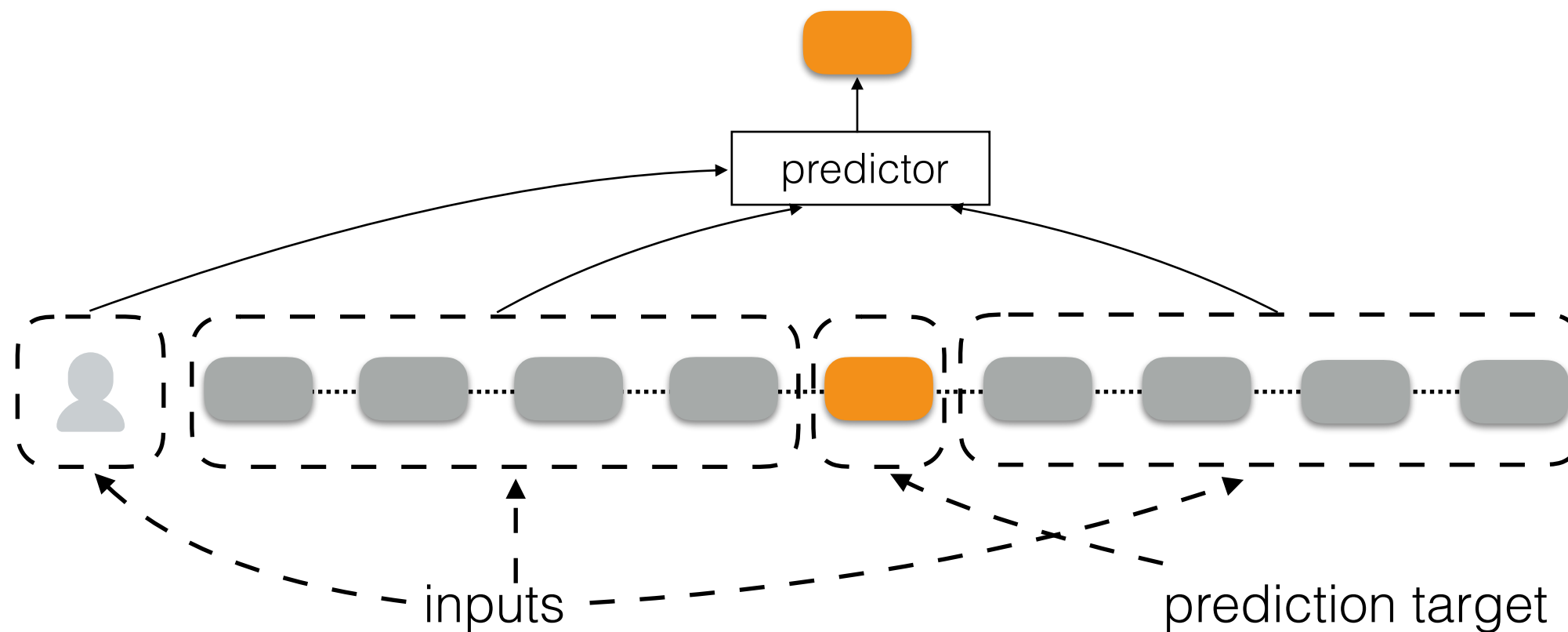


utilization-to-vector (**util2vec**)

inside each window

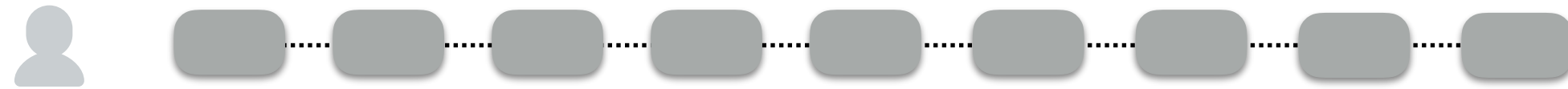


$2n+1$ actions ($n=4$)

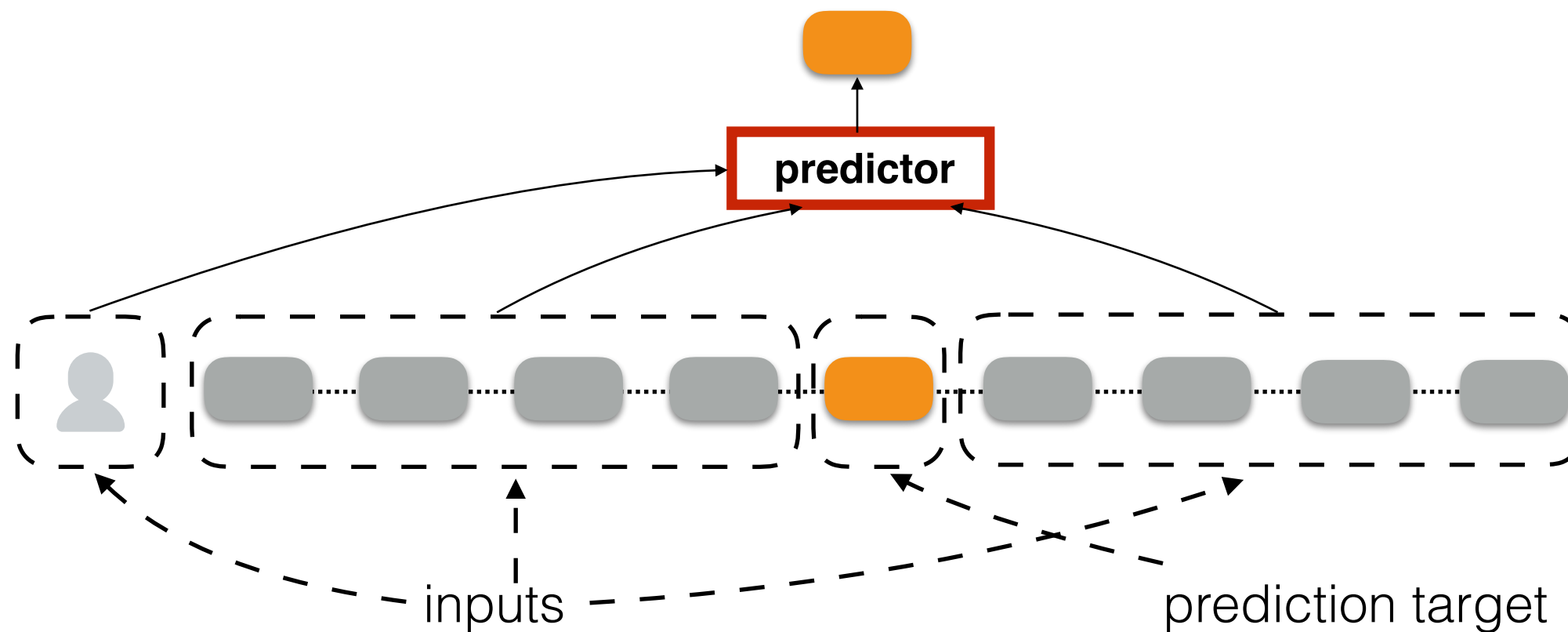


utilization-to-vector (**util2vec**)

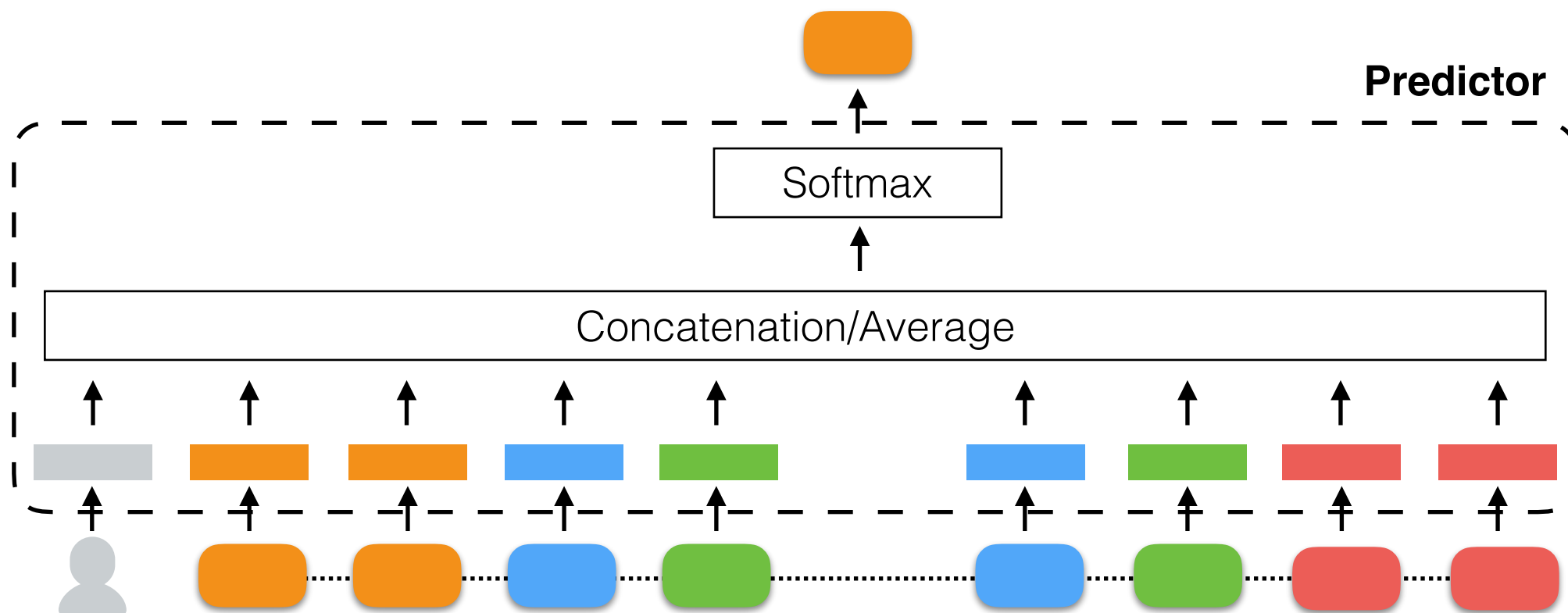
inside each window



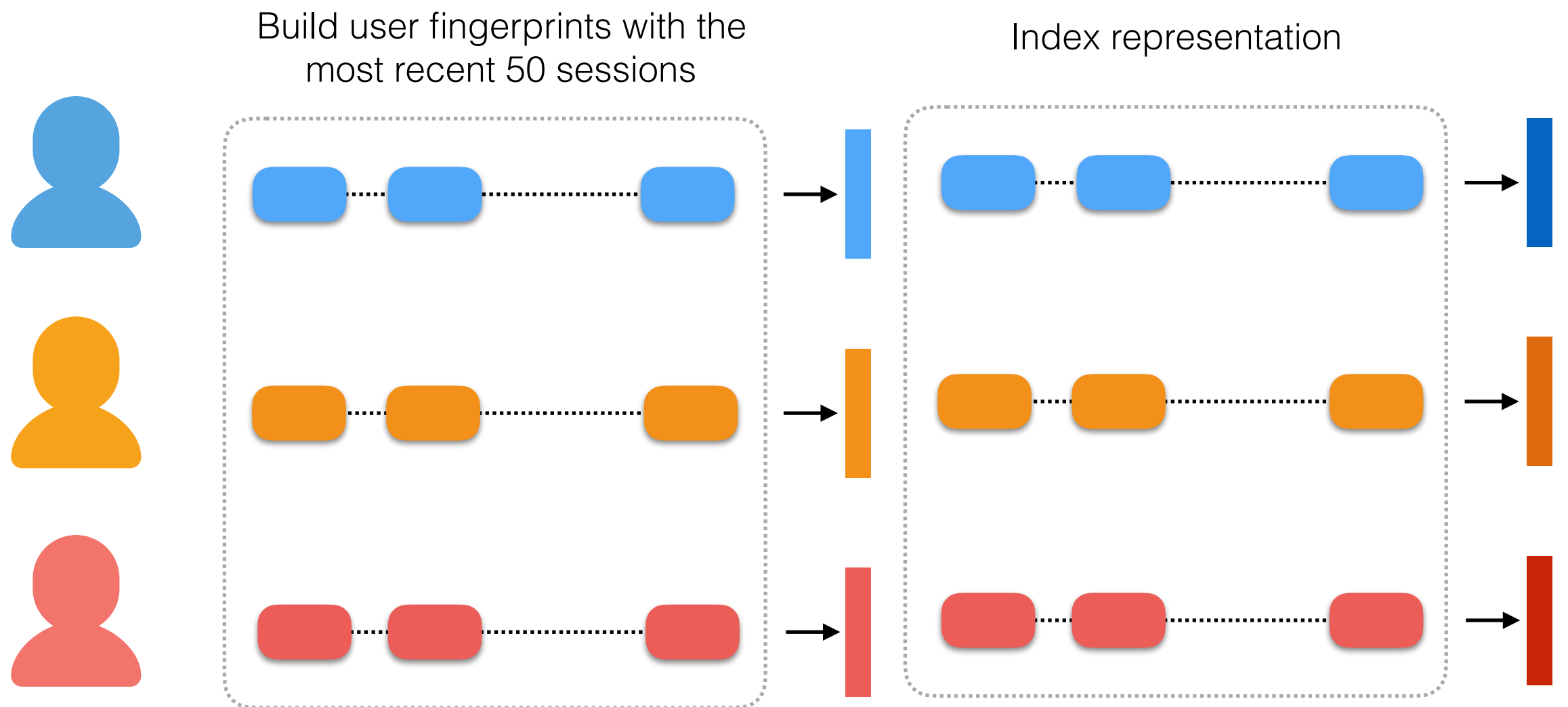
$2n+1$ actions ($n=4$)



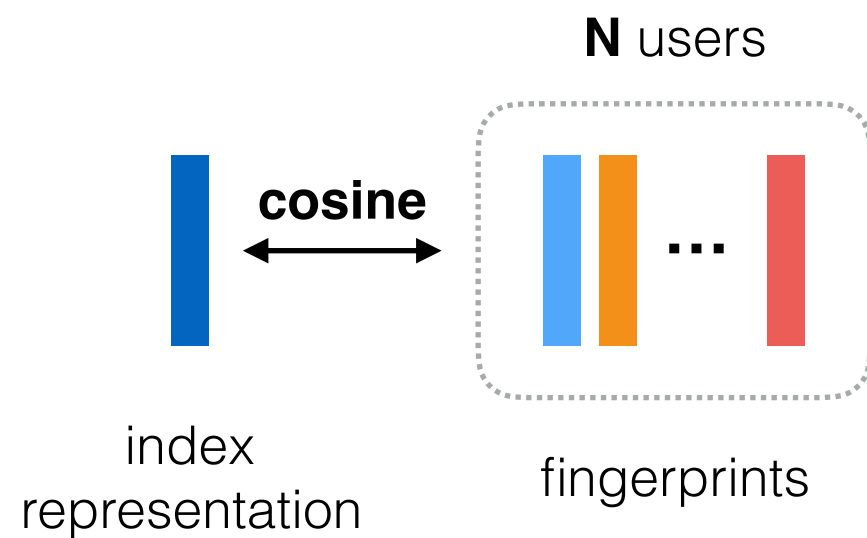
utilization-to-vector (**util2vec**) *predictor*



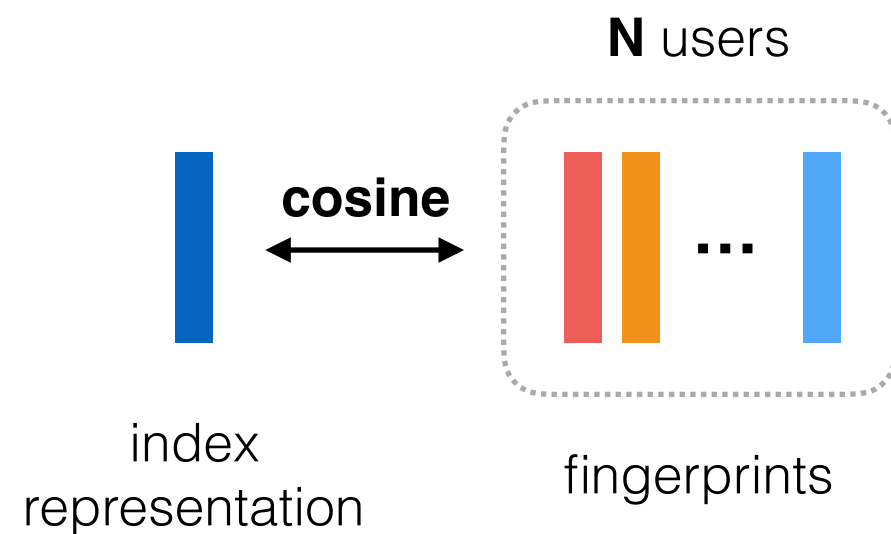
Evaluation of **util2vec**



Evaluation of **util2vec**



Reciprocal Rank (RR) = 1



Reciprocal Rank (RR) = $1/N$



Evaluation of **util2vec**

Model Training: **22 billion** actions from **3 million** users

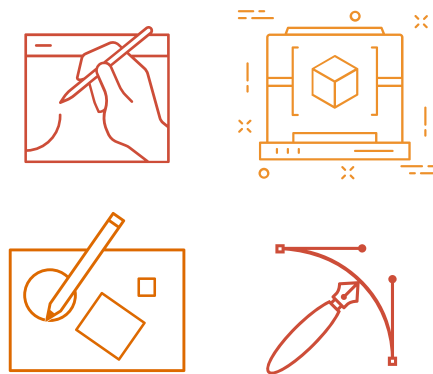
Model Testing: randomly selected **15K** users

each had more than 100 sessions

Model	Mean Reciprocal Rank
util2vec	0.824
bag-of-actions+tf-idf	0.604
bag-of-actions	0.594
% of improvement	31.72%

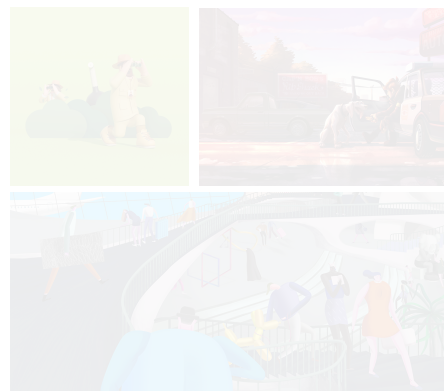
Part II. Personalizing Software

Part II



Personalizing software

Part III

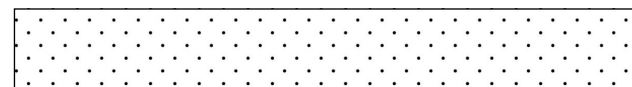


Personalizing web



inspiration engine

Part I



utilization-to-vector (**util2vec**)

action

action

action

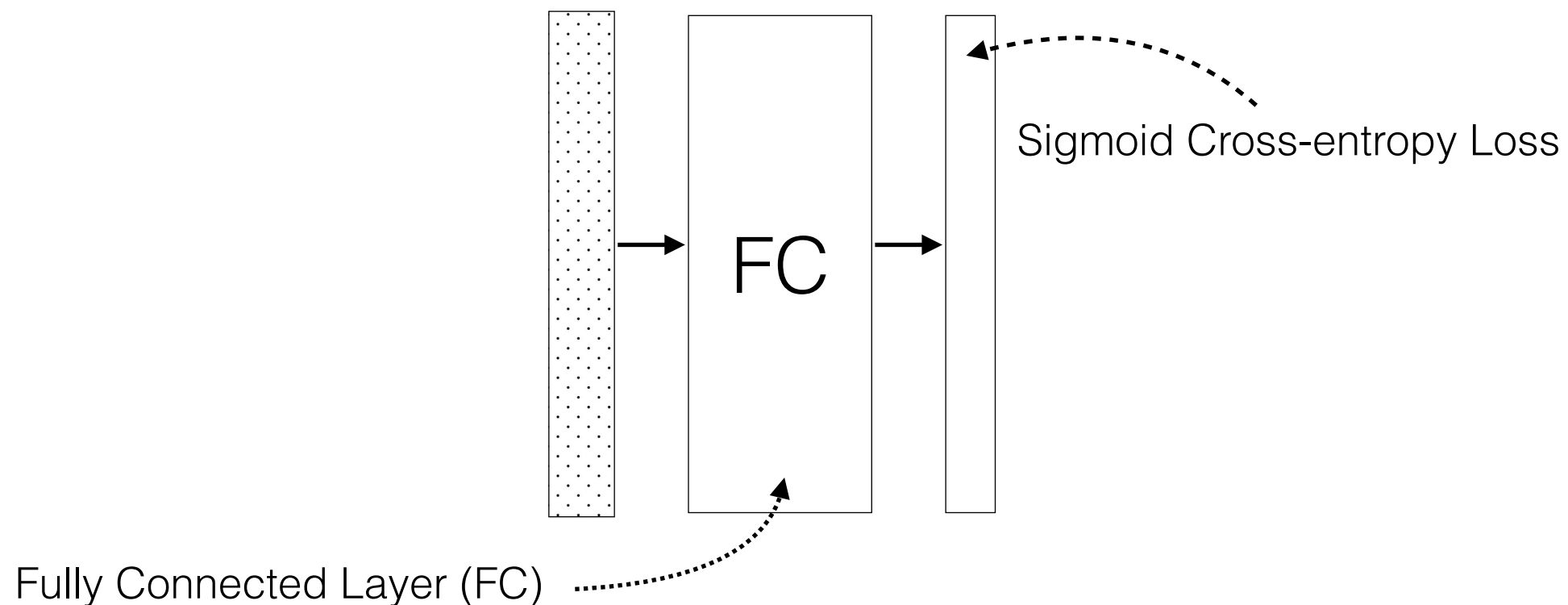
action

action

Personalizing software: software user tagging



 Web design, Photography, Graphic Design...



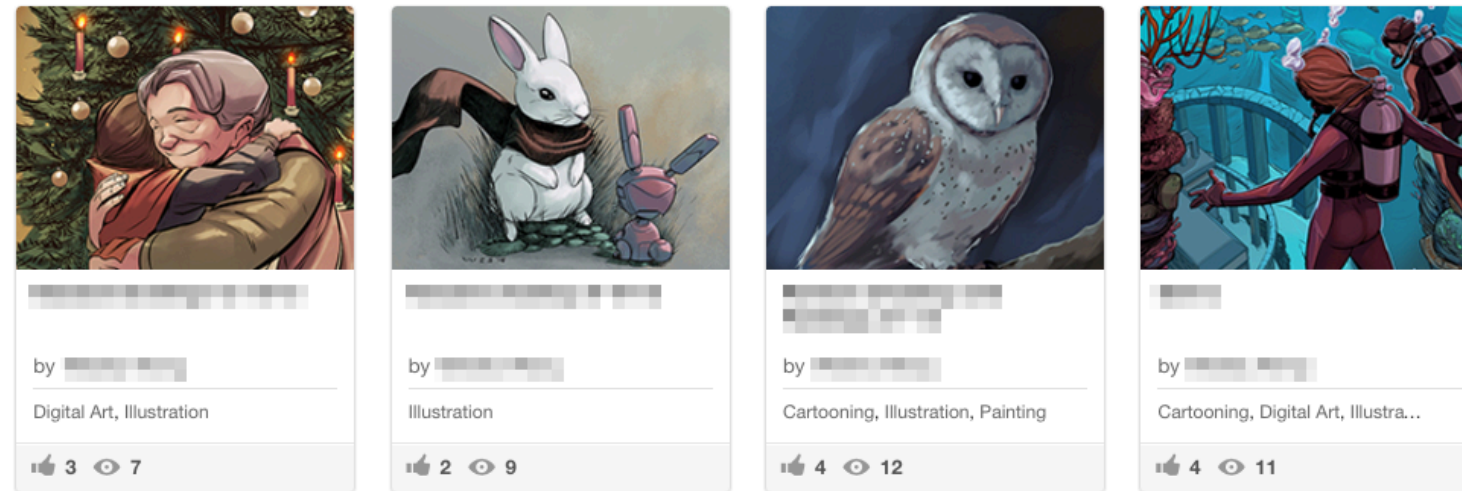
Quantitative evaluation

67 tags self-disclosed by **65,331** users (on Behance).

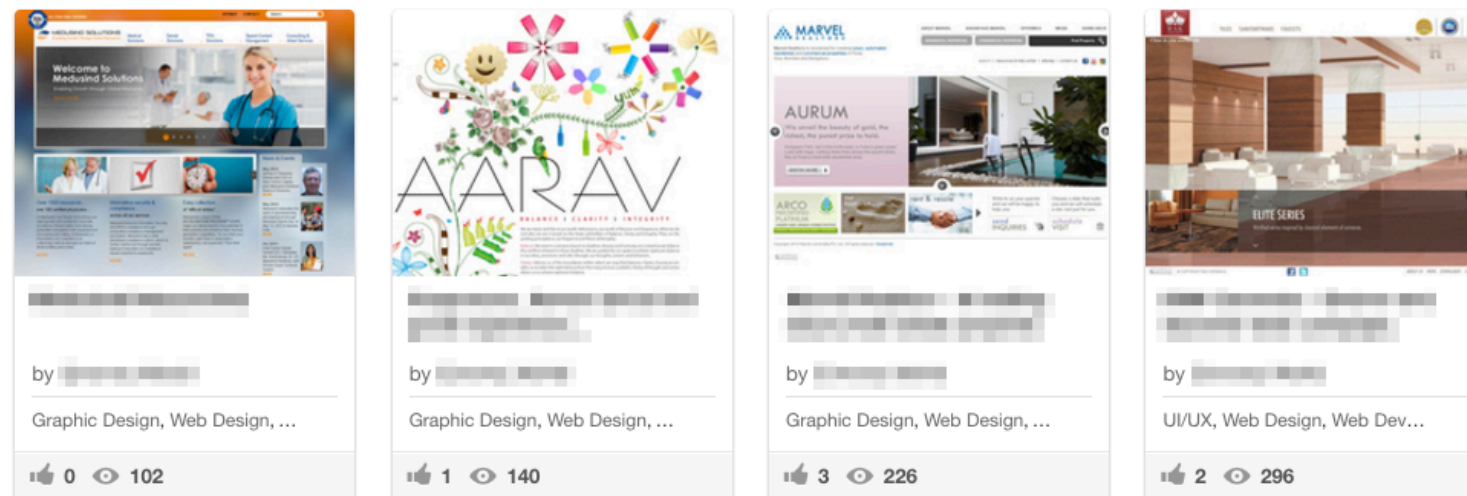
45,331 users for training, **20,000** users for testing

Recall@K	1	2
util2vec	0.232	0.357
popular tags	0.177	0.264
% improvement	31%	35%

Qualitative evaluation



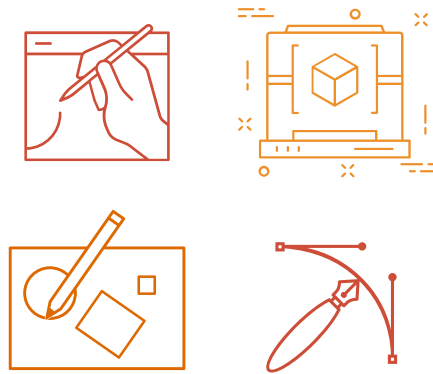
Our predictions: Illustration, Digital Art, Character Design, Cartooning, Graphic Design



Our predictions: Web Design, Web Development, UI/UX, Graphic Design, Branching

Part III. Personalizing web

Part II



Personalizing software

Part III

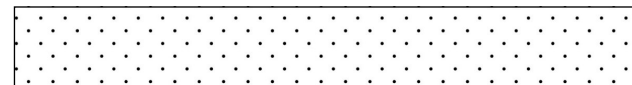


Personalizing web



inspiration engine

Part I



utilization-to-vector (**util2vec**)

action

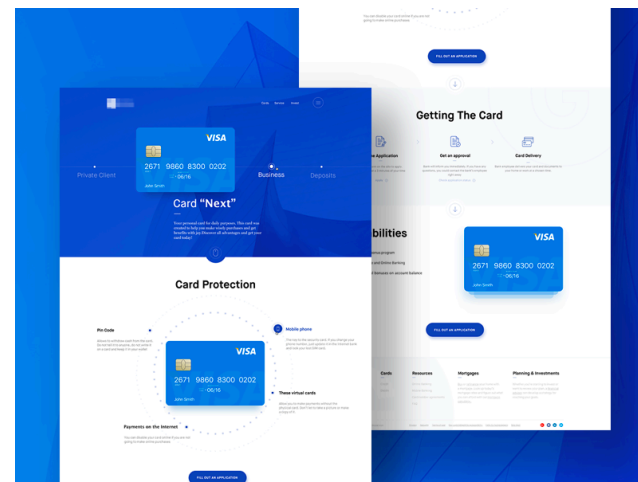
action

action

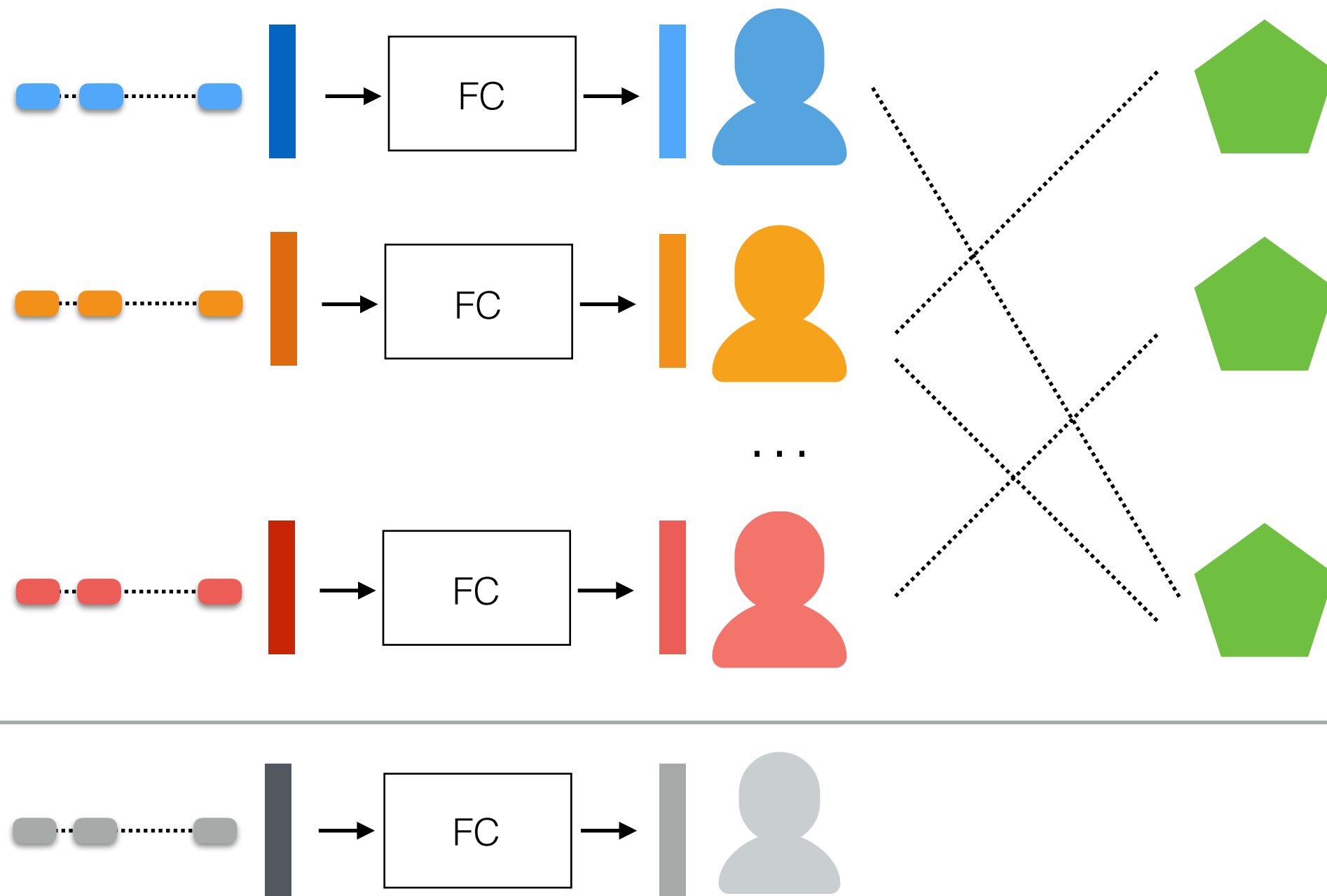
action

action

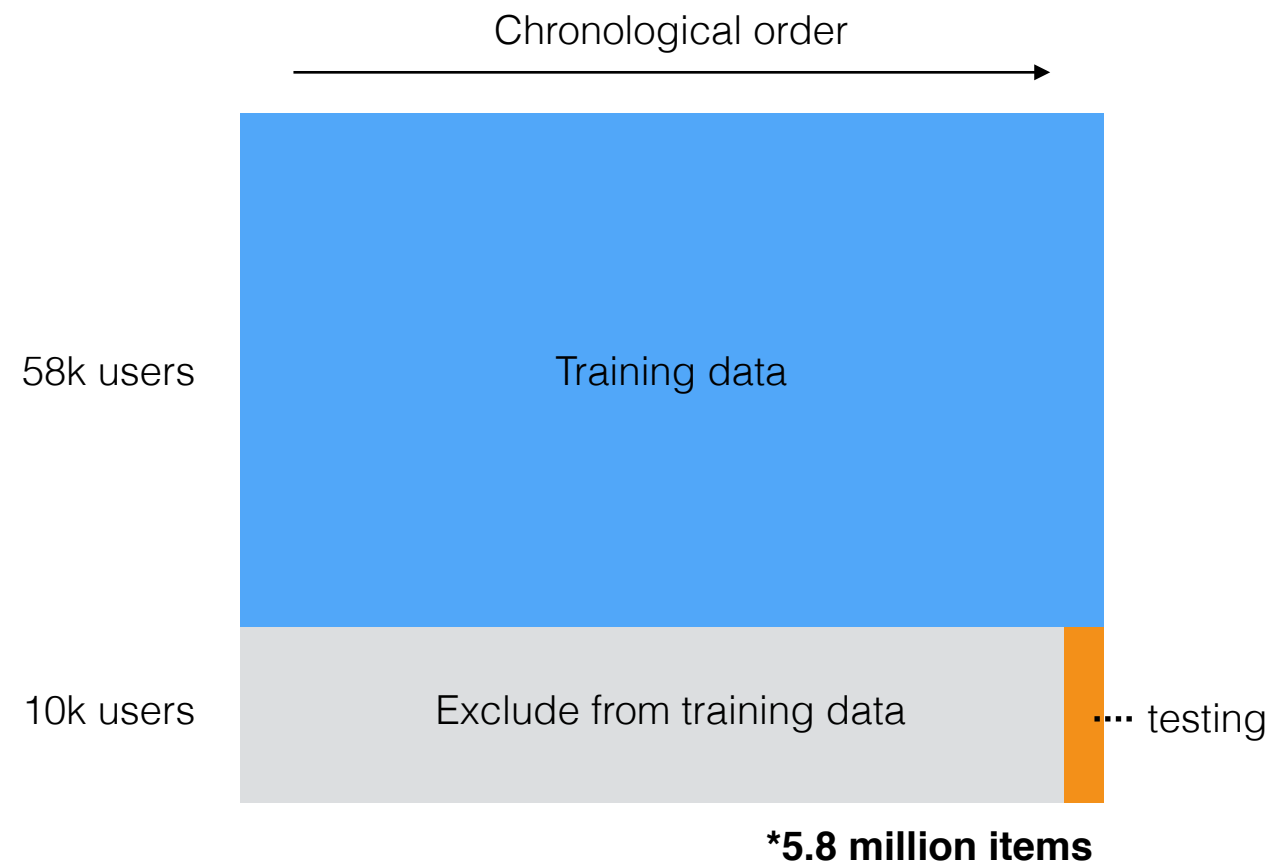
Personalizing web: cold-start creative content recommendation



Two-step recommendation algorithm

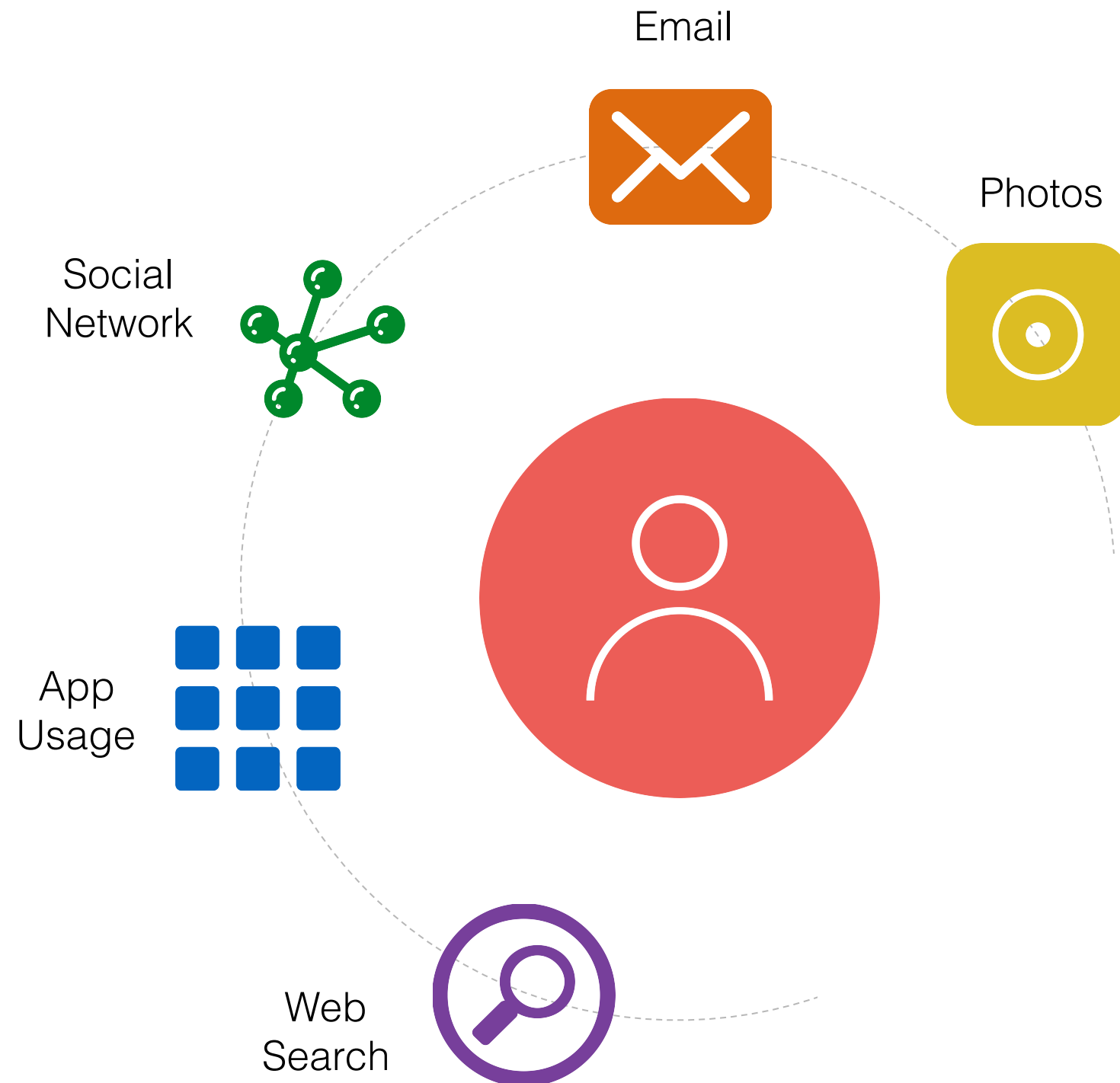


Evaluation



Recall@K	100	200
util2vec	0.0143	0.0213
popularity	0.0118	0.0188
% improvement	21.2%	13.3%

Conclusion: User-centric Personalization



Thank you!

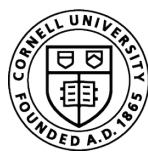
Longqi Yang

Ph.D. Student, Computer Science, Cornell Tech, Cornell University

Email: ylongqi@cs.cornell.edu

Web: bit.ly/longqi

Twitter: [@ylongqi](https://twitter.com/ylongqi)



**CORNELL
TECH**



Cornell University
Department of Computer Science



the small data lab

