OpenRec: A Modular Framework for Extensible and Adaptable Recommendation Algorithms


Longqi Yang


Eugene Bagdasaryan


Joshua Gruenstein


Cheng-Kang(Andy) Hsieh


Deborah Estrin

CORNELL TECH

Cornell CIS Computer Science

Funders: NSF Oath: A Verizon company

Promising future of personalization and recommender systems

Education

Healthcare

Social network

Media

Food and Diet

e-Commerce

Recommendation algorithms are increasingly complex

Recommendation algorithms are increasingly complex

Diverse user feedback signals

rating like skip follow

click save watch listen ...

Recommendation algorithms are increasingly complex

Diverse user feedback signals

rating like skip follow

click save watch listen …

Heterogeneous data streams and context

user demographics user social media posts

Item descriptions videos images

activities location mood …

Recommendation algorithms are increasingly complex

Diverse user feedback signals

rating · like · skip · follow · click · save · watch · listen · …

Heterogeneous data streams and context

user demographics · user social media posts · Item descriptions · videos · images · activities · location · mood · …

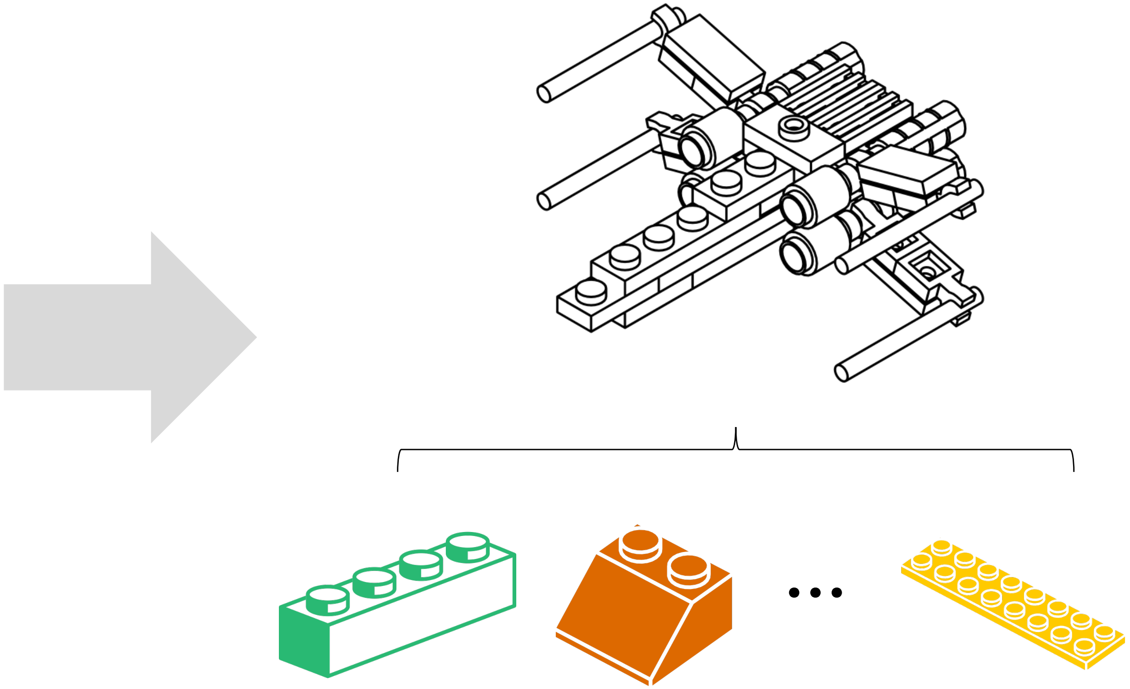
Complex goals

accuracy · diversity · novelty · fairness · quality · interpretability · …

Bag of algorithms

However, current recommendation algorithms lack <u>simplicity</u> and <u>modularity</u>.

Bag of algorithms



OpenRec

Apache License 2.0

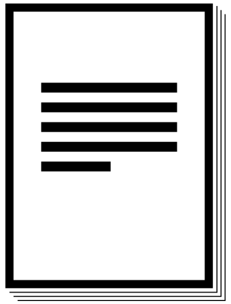
Modularity

- Easy to **extend** and **adapt** to various scenarios.
- Quick experimentation (e.g., model selection) and idea exploration.
- Comparable (sometimes even better) performance.

Current practice vs. OpenRec

Prior research | **Your research/application**

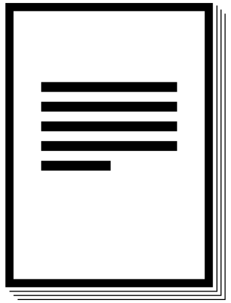
current practice



News recommender → Music recommender

Current practice vs. OpenRec

current practice

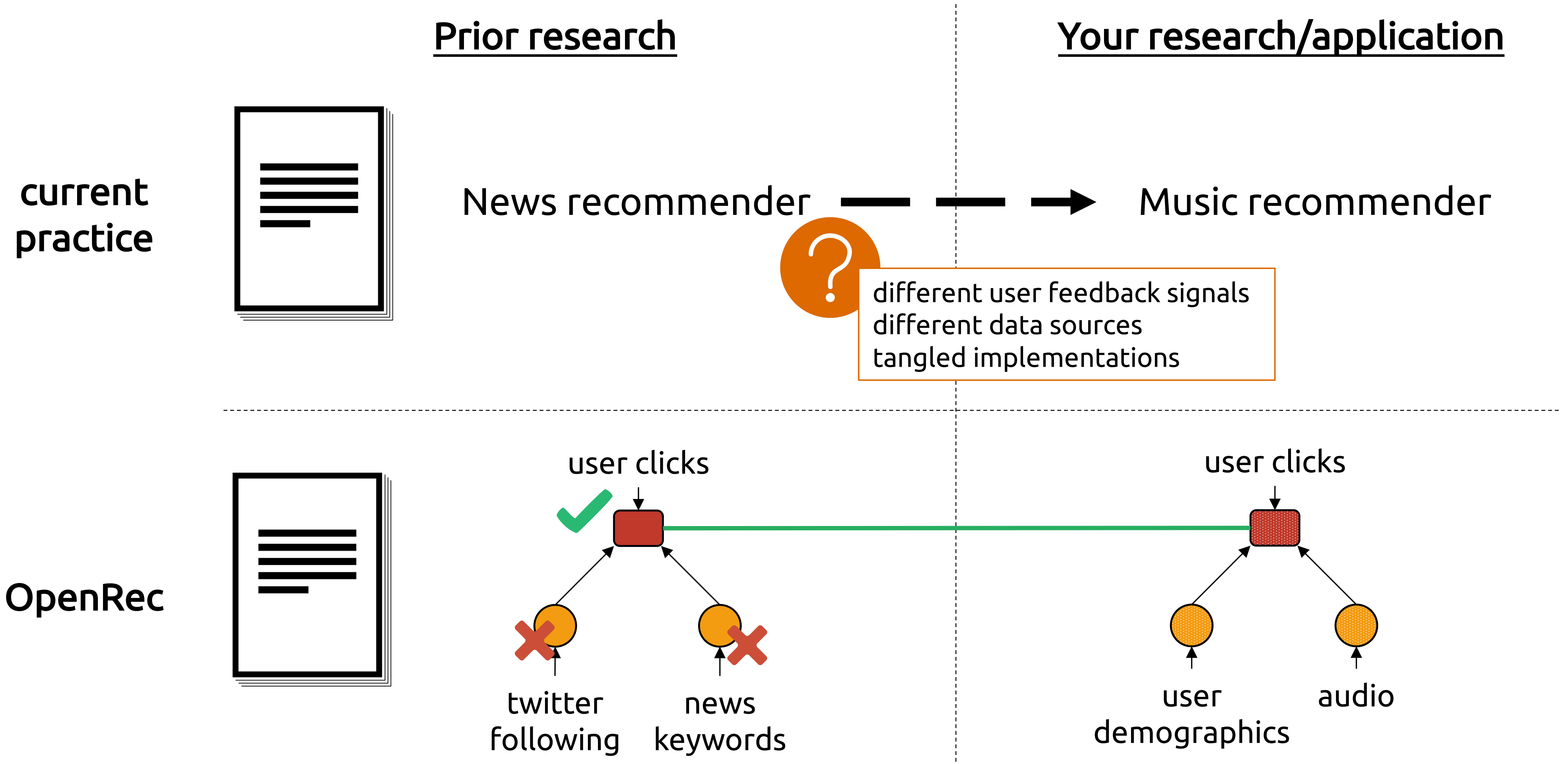


Prior research		Your research/application
News recommender	→	Music recommender

?

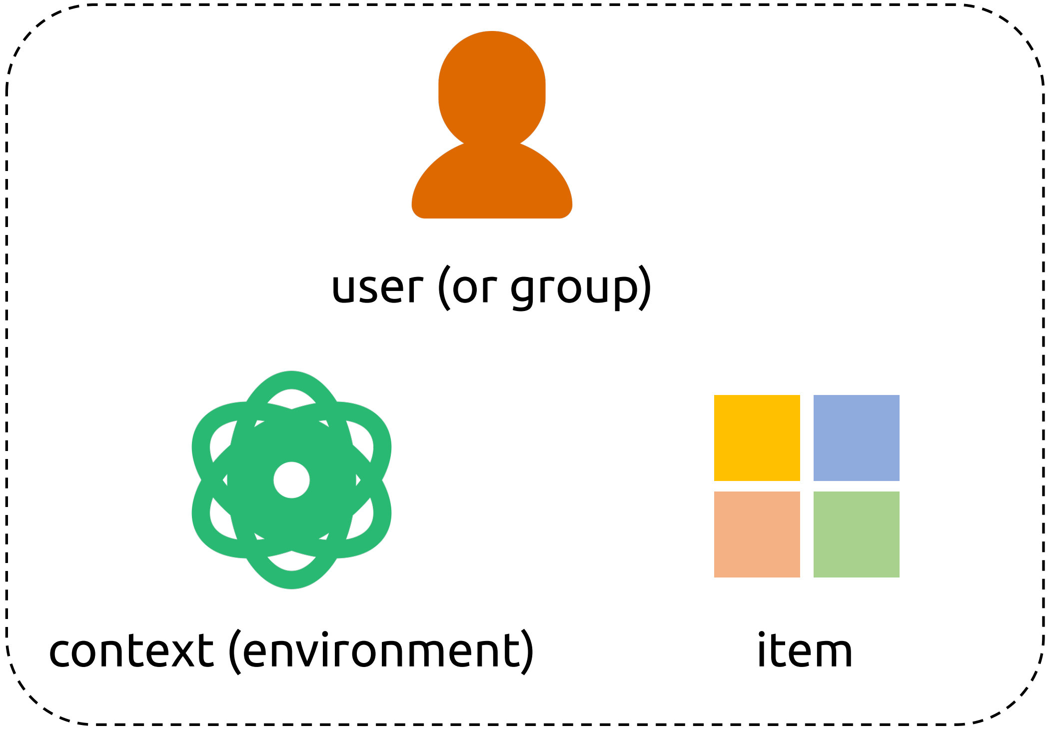
different user feedback signals
different data sources
tangled implementations

Current practice vs. OpenRec



1. Abstraction and interface
2. Implementations
3. Simple use cases
4. Takeaways and Future work

Abstract entities in recommendation algorithms



1 Abstraction and interface

Building a recommendation algorithm

Entity	User	Context	Item

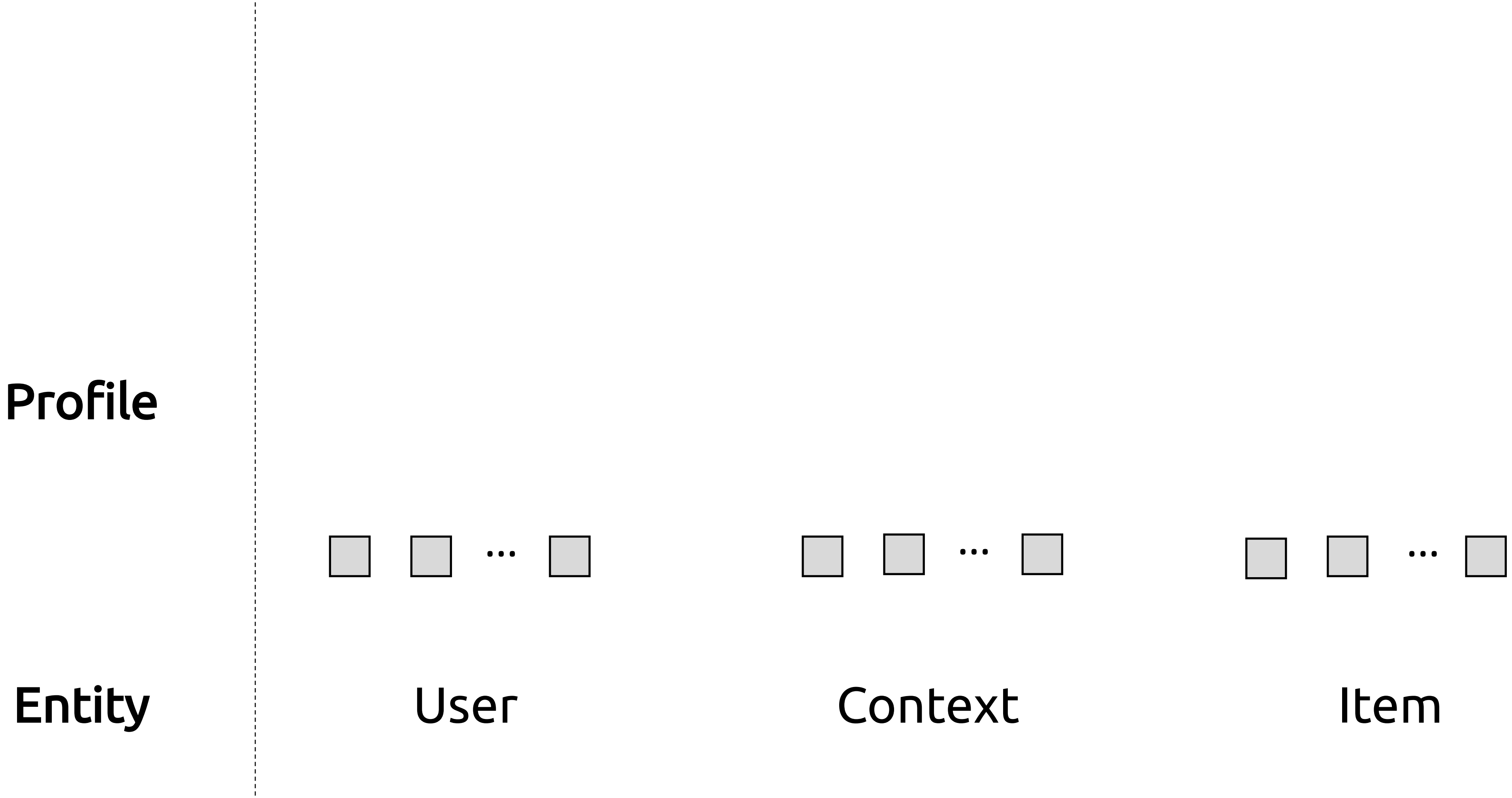
1 **Abstraction and interface**

Building a recommendation algorithm

Profile			
Entity	User	Context	Item

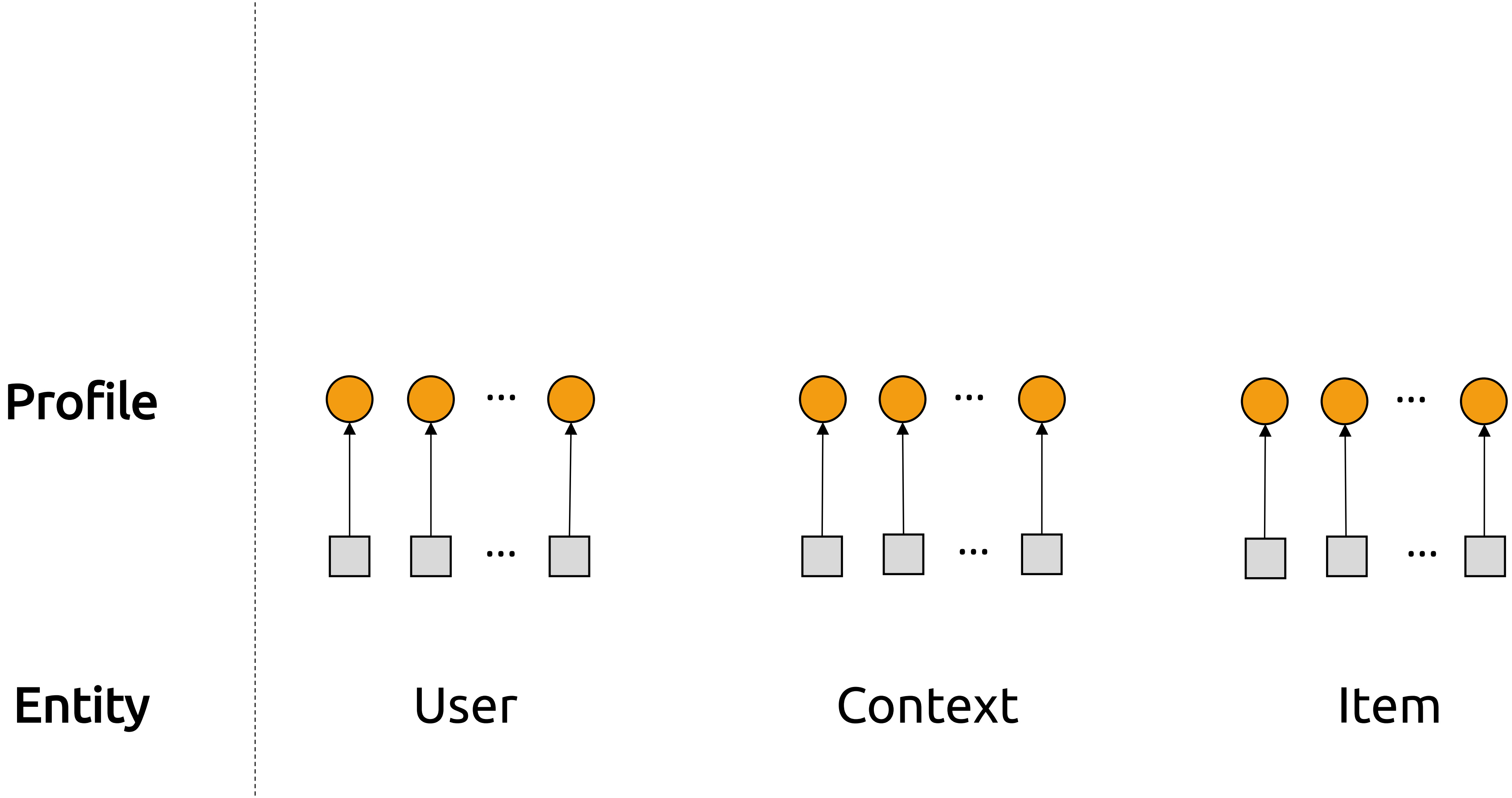
1 **Abstraction and interface**

Building a recommendation algorithm



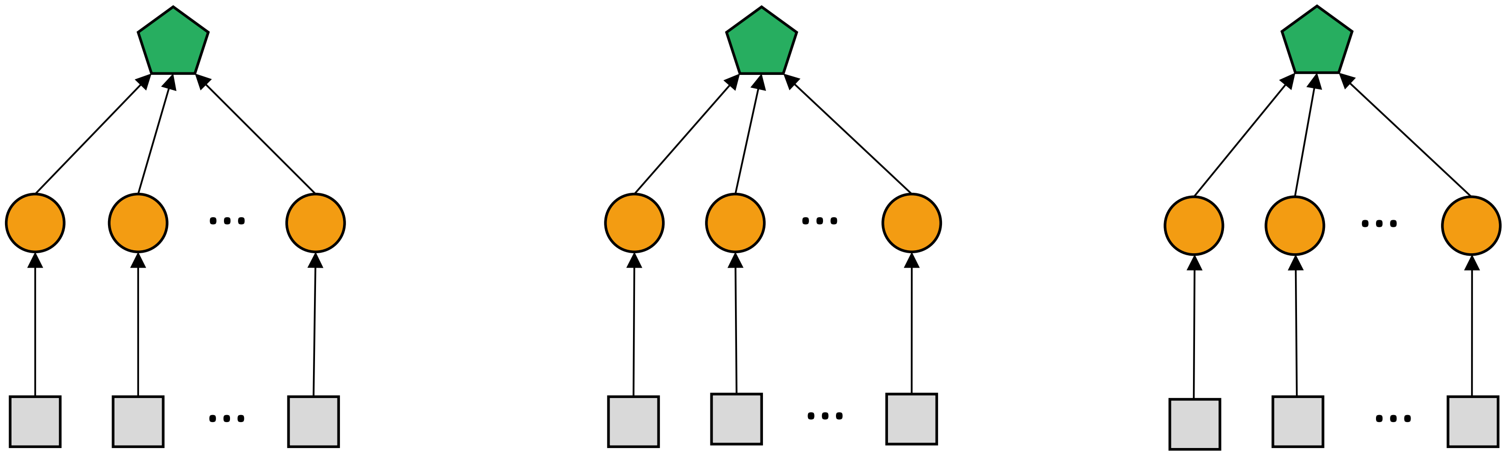
1 Abstraction and interface

Building a recommendation algorithm



Building a recommendation algorithm

Profile



Entity

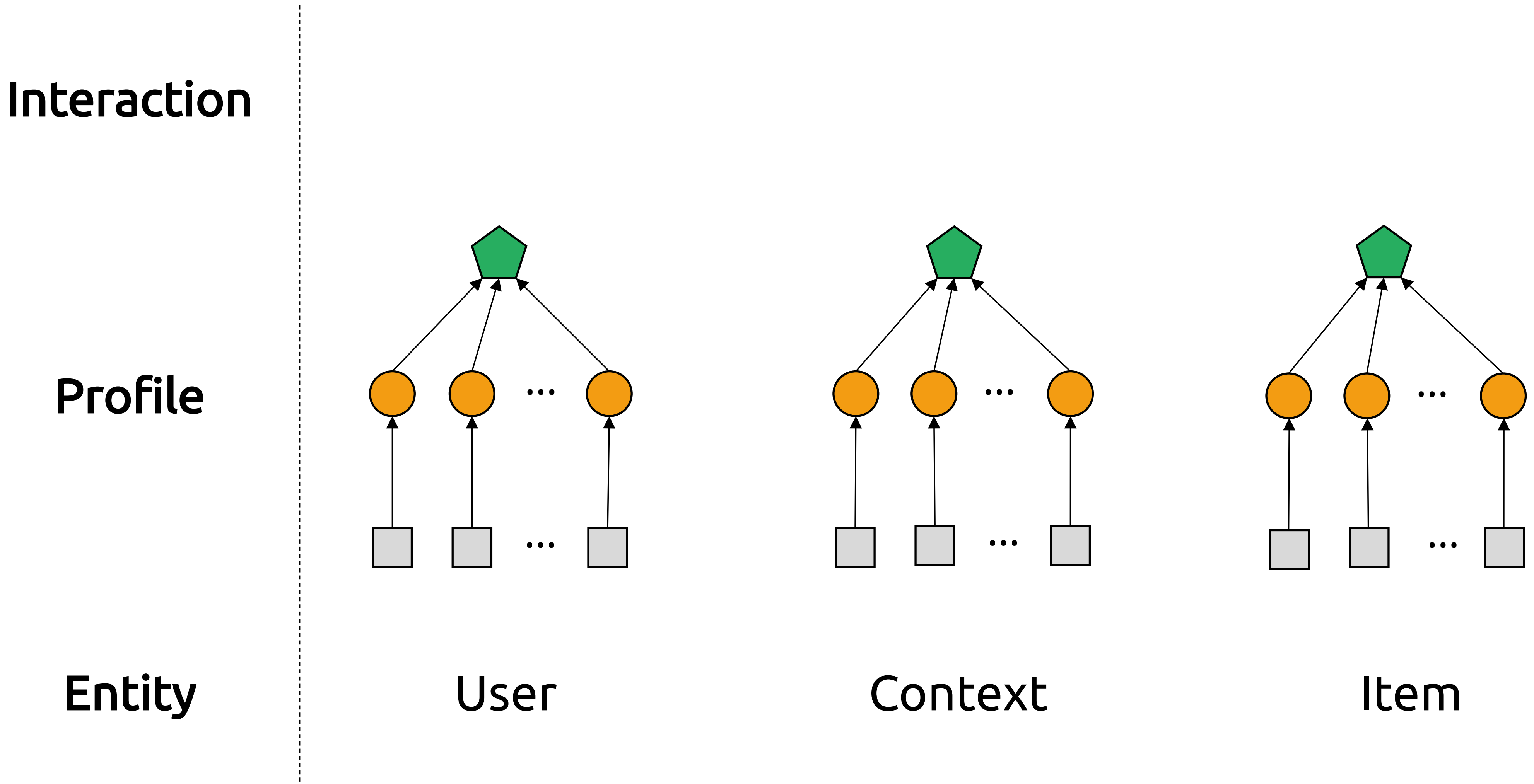
User

Context

Item

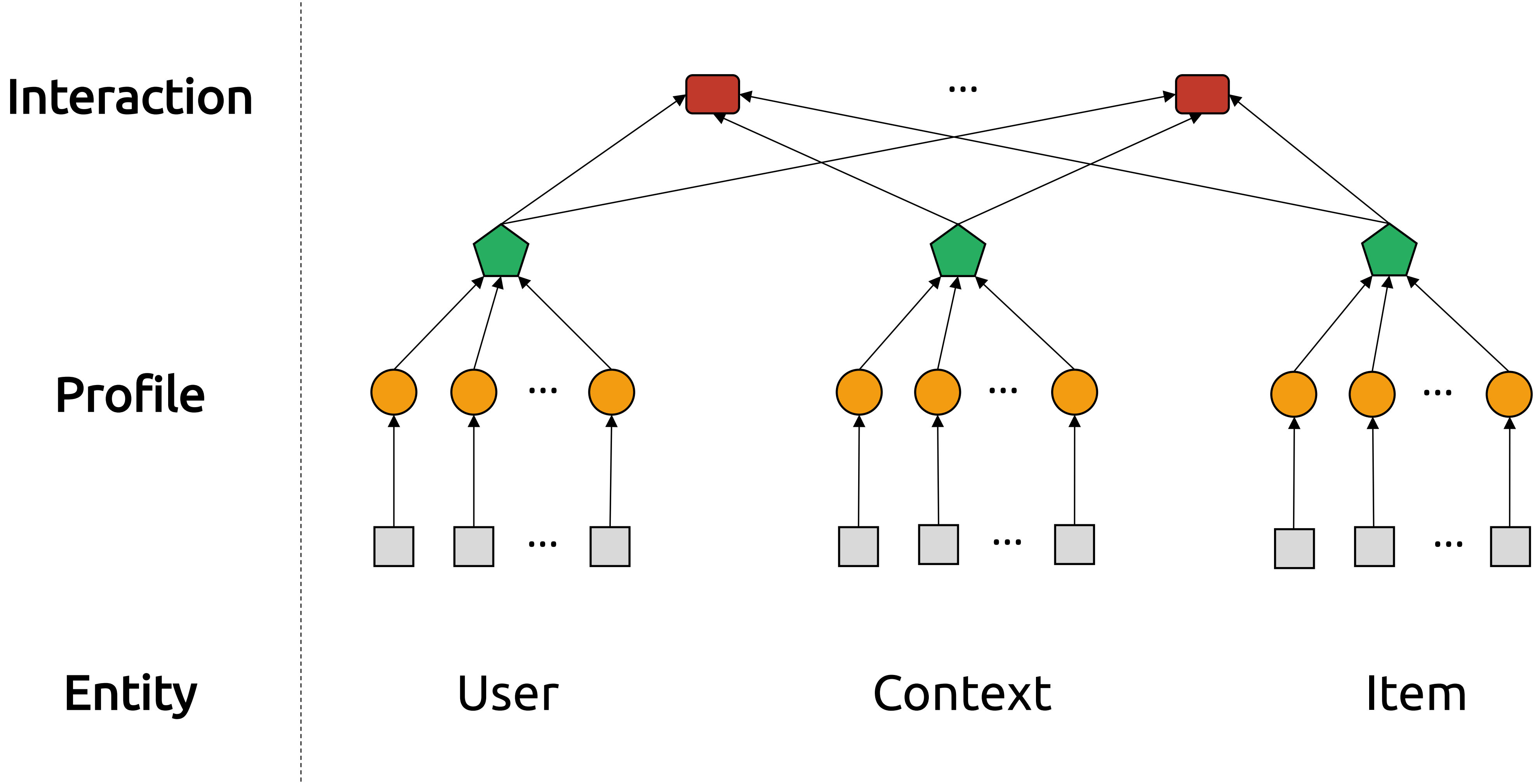
1 Abstraction and interface

Building a recommendation algorithm



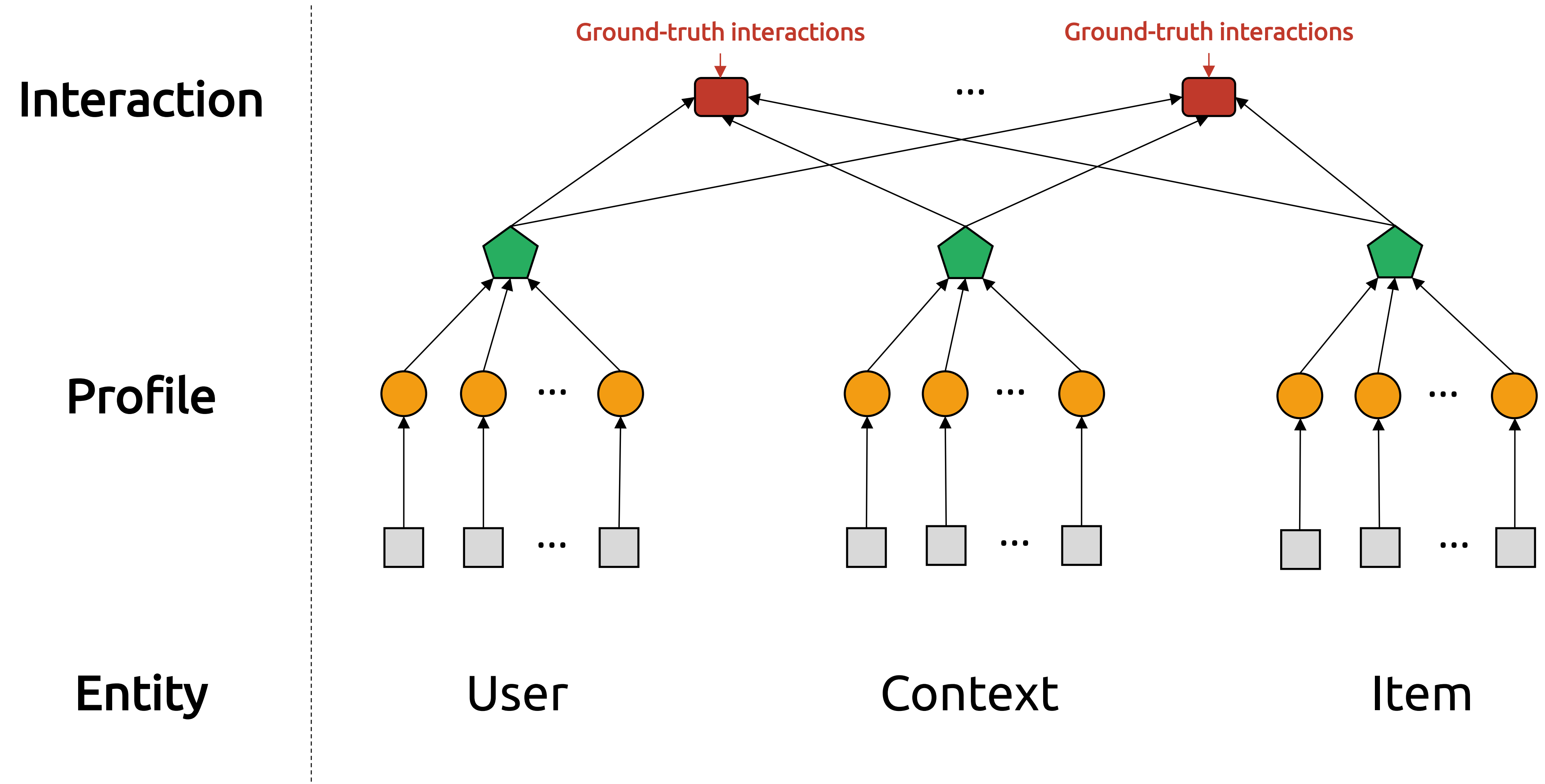
1 Abstraction and interface

Building a recommendation algorithm



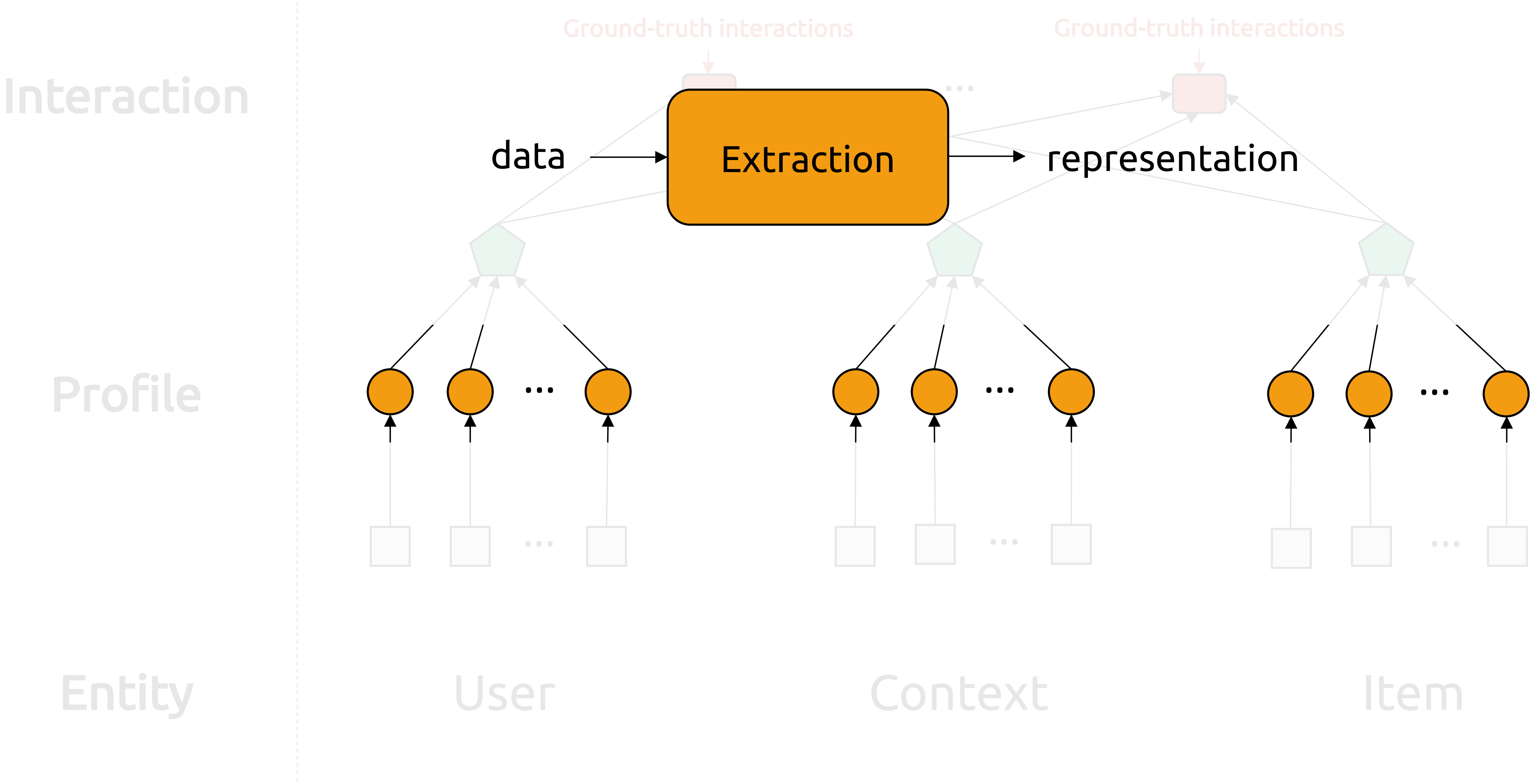
1 Abstraction and interface

Building a recommendation algorithm



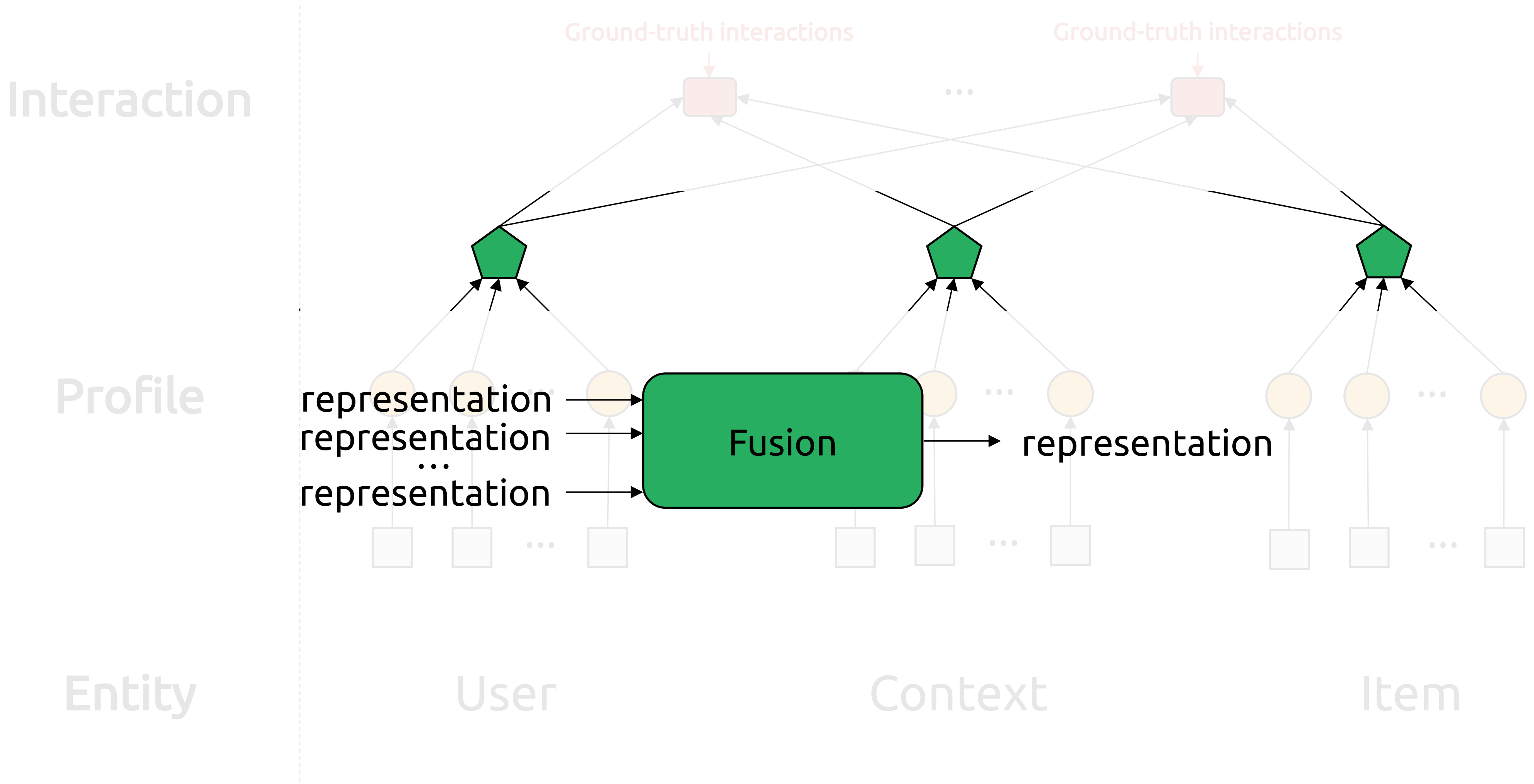
1 Abstraction and interface

Extraction: extract representations



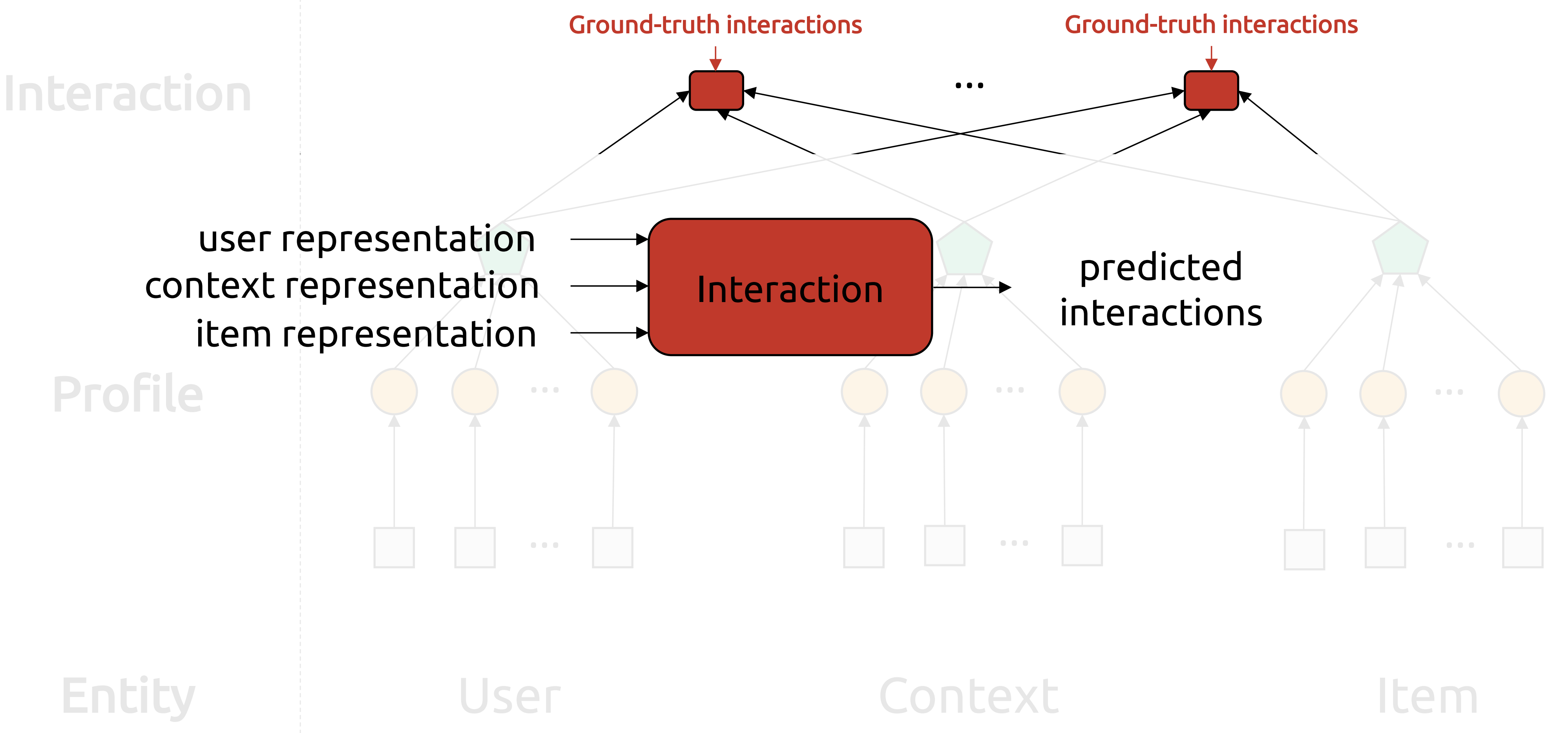
1 Abstraction and interface

Fusion: fuse representations



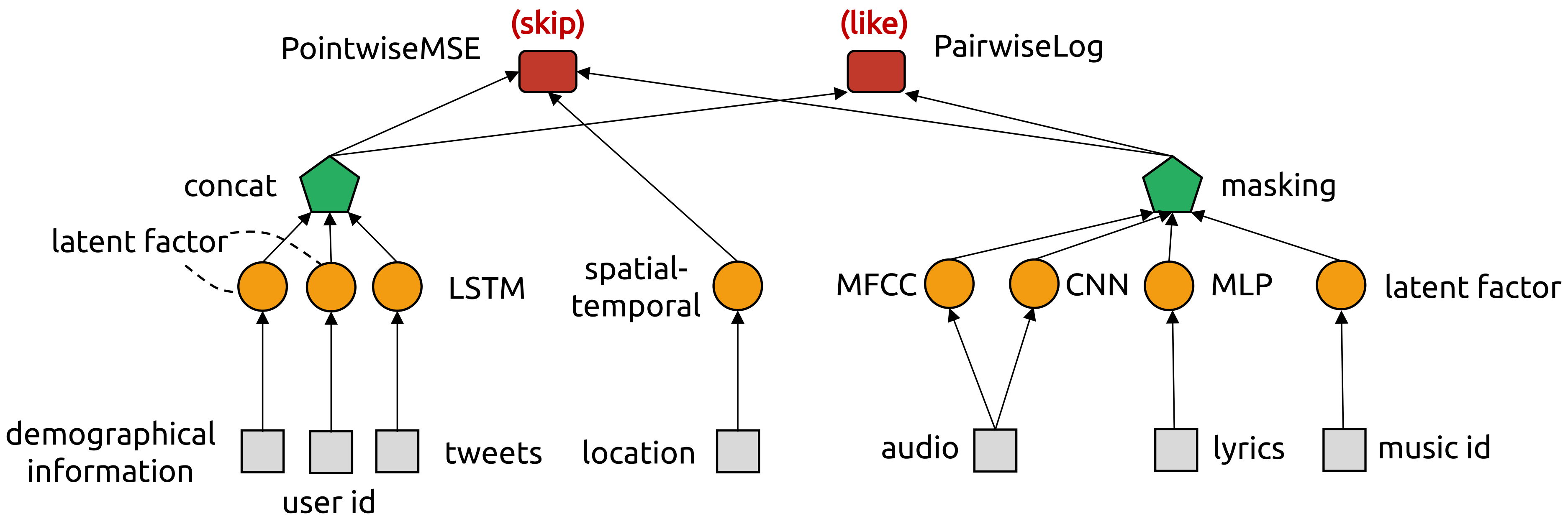
1 **Abstraction and interface**

Interaction: predict clicks/likes/ratings…



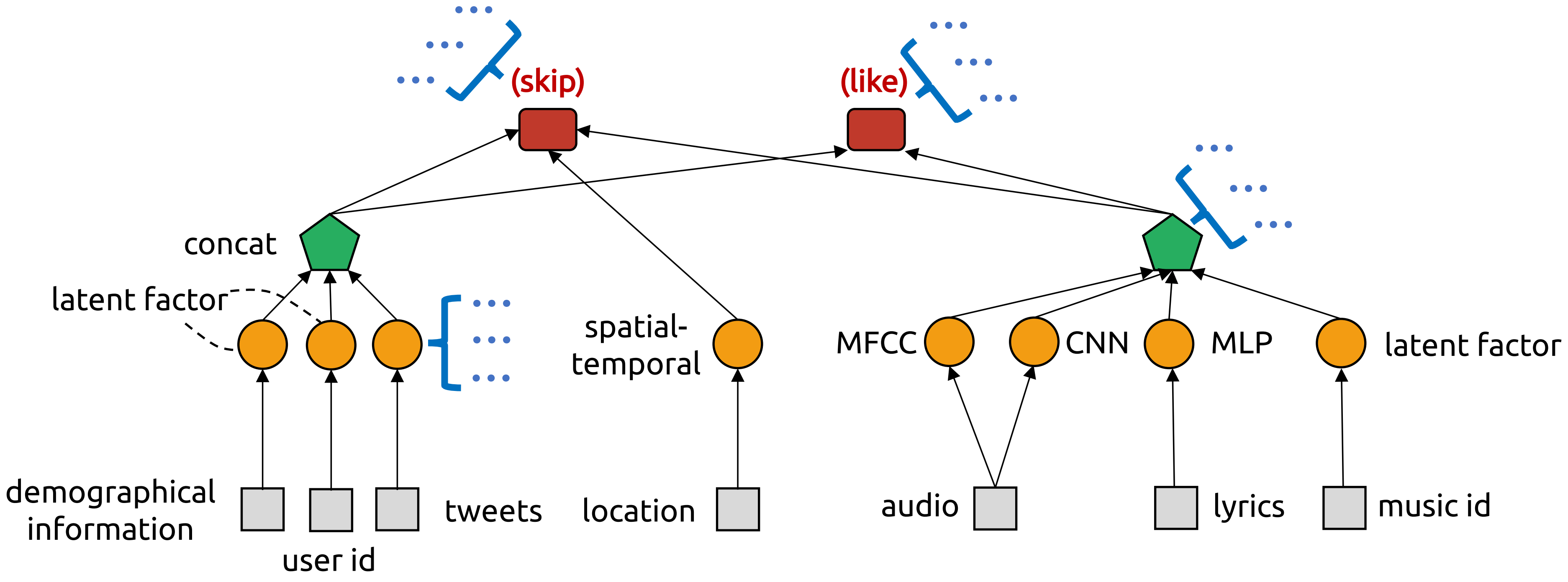
1 Abstraction and interface

A hypothetical music recommendation algorithm



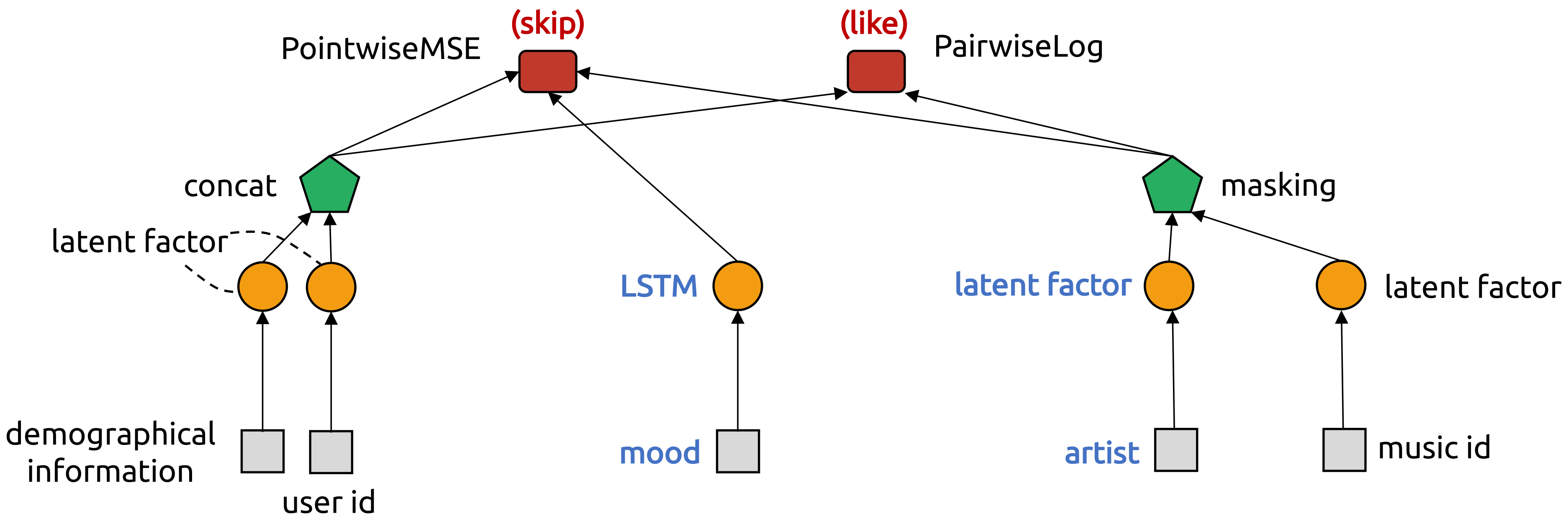
1 **Abstraction and interface**

A hypothetical music recommendation algorithm



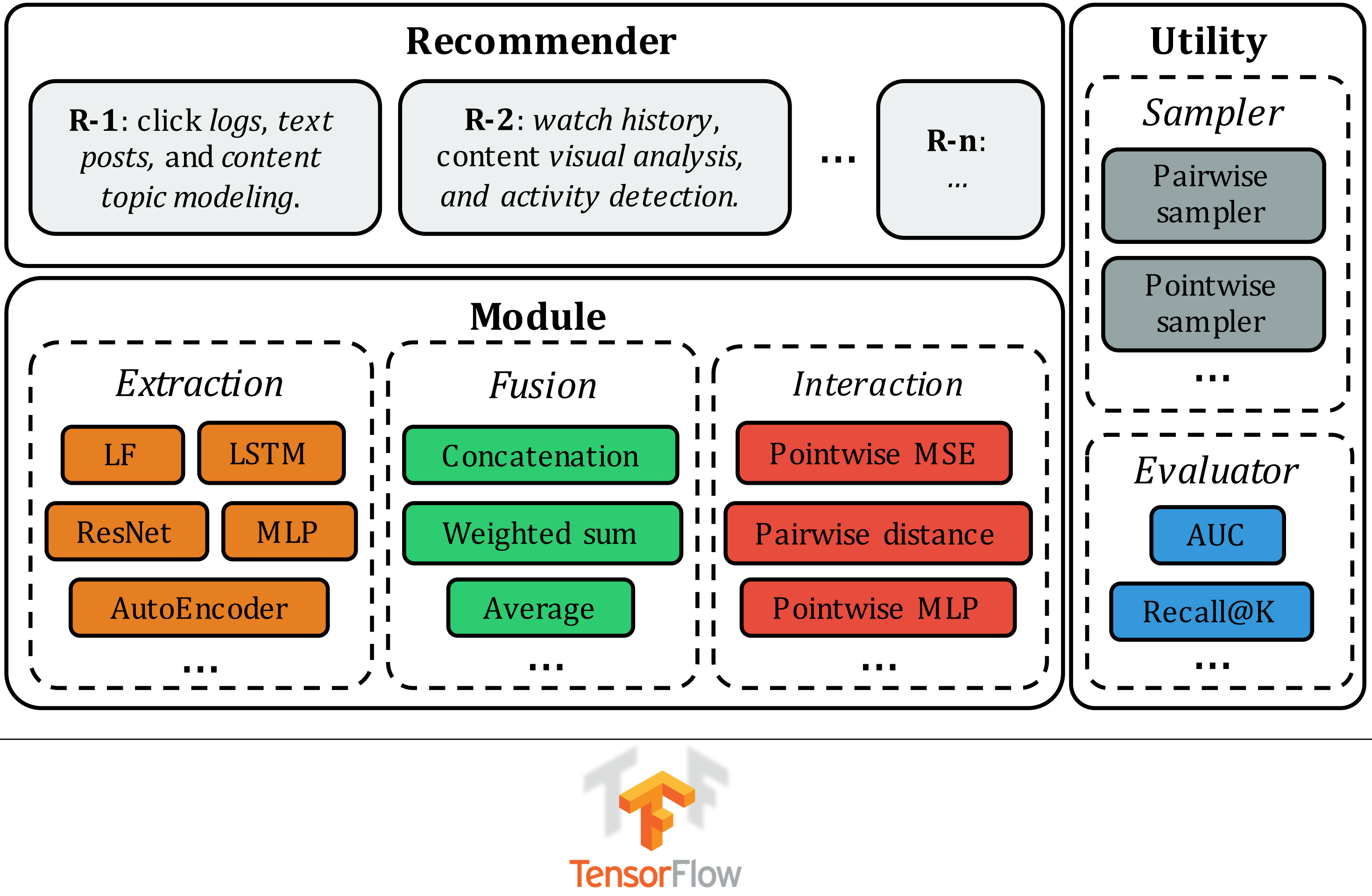
1 **Abstraction and interface**

A hypothetical music recommendation algorithm

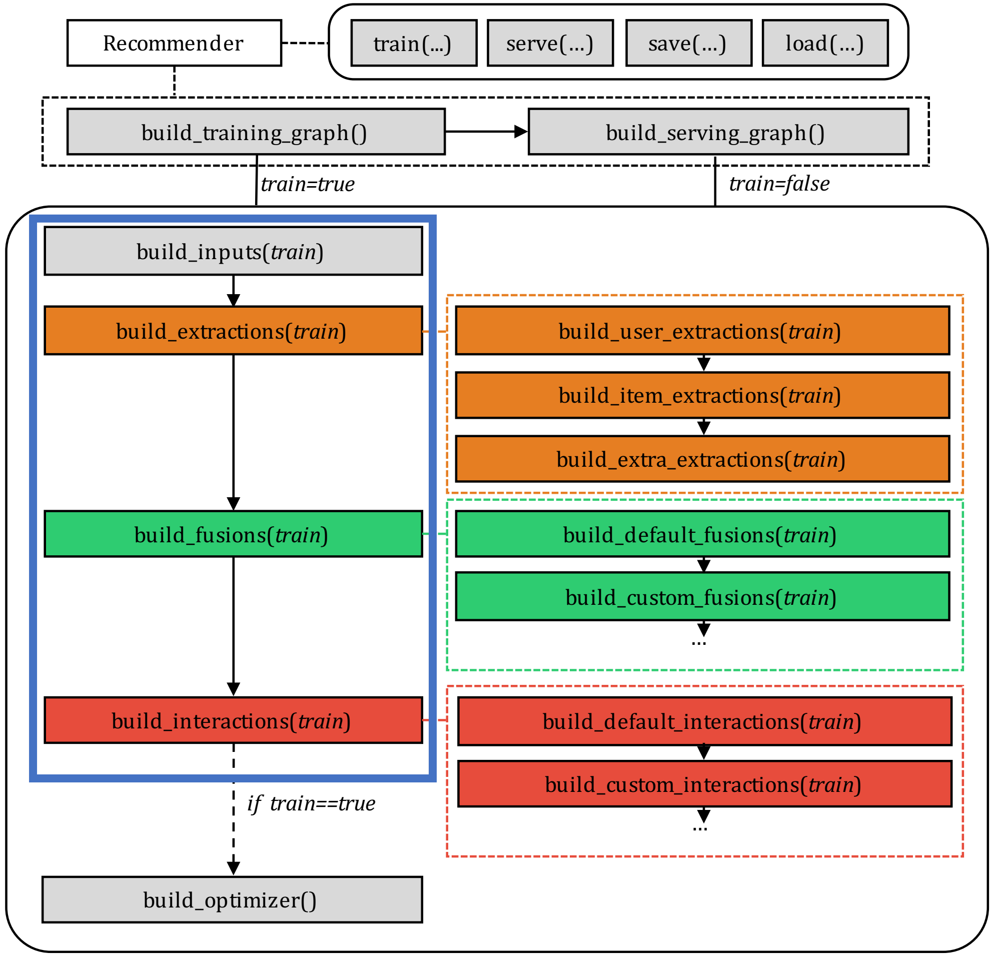


1 Abstraction and interface

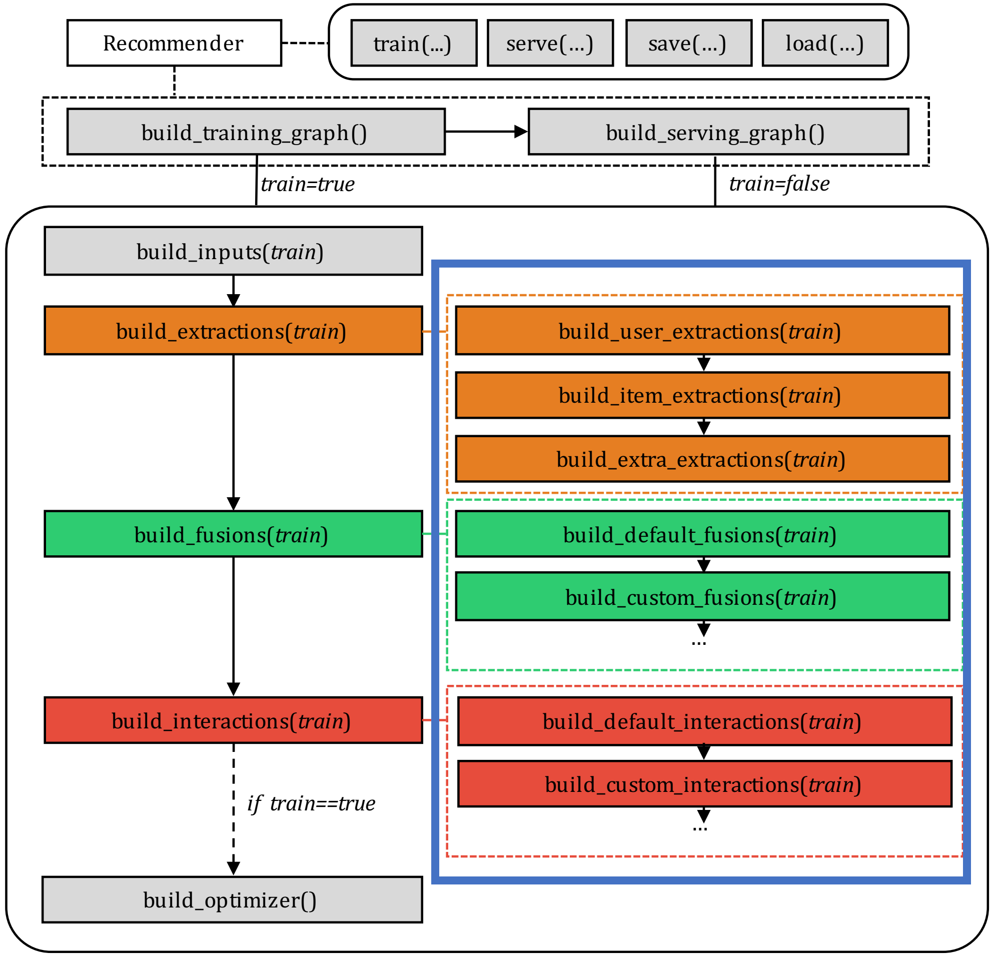
OpenRec framework structure



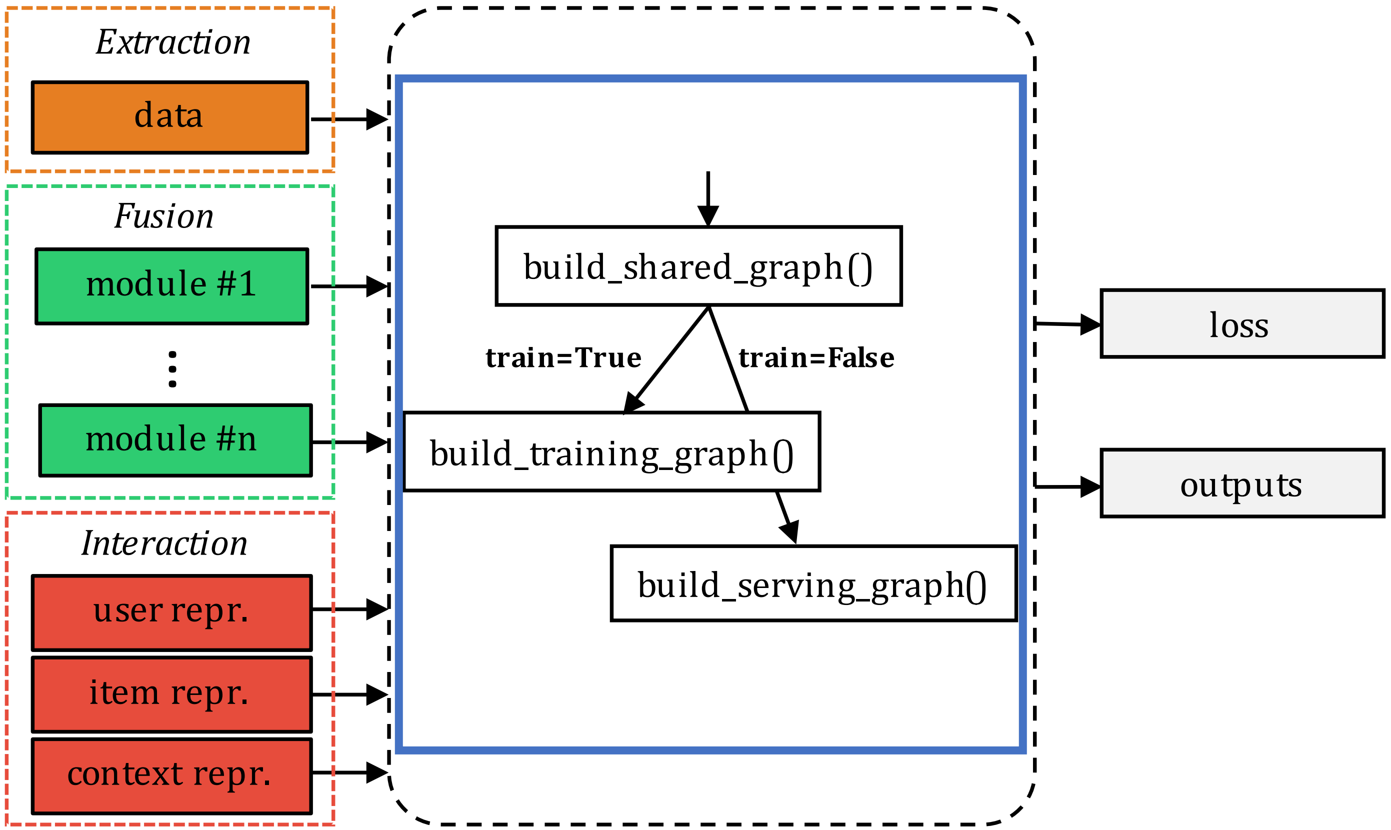
Inside a *Recommender*



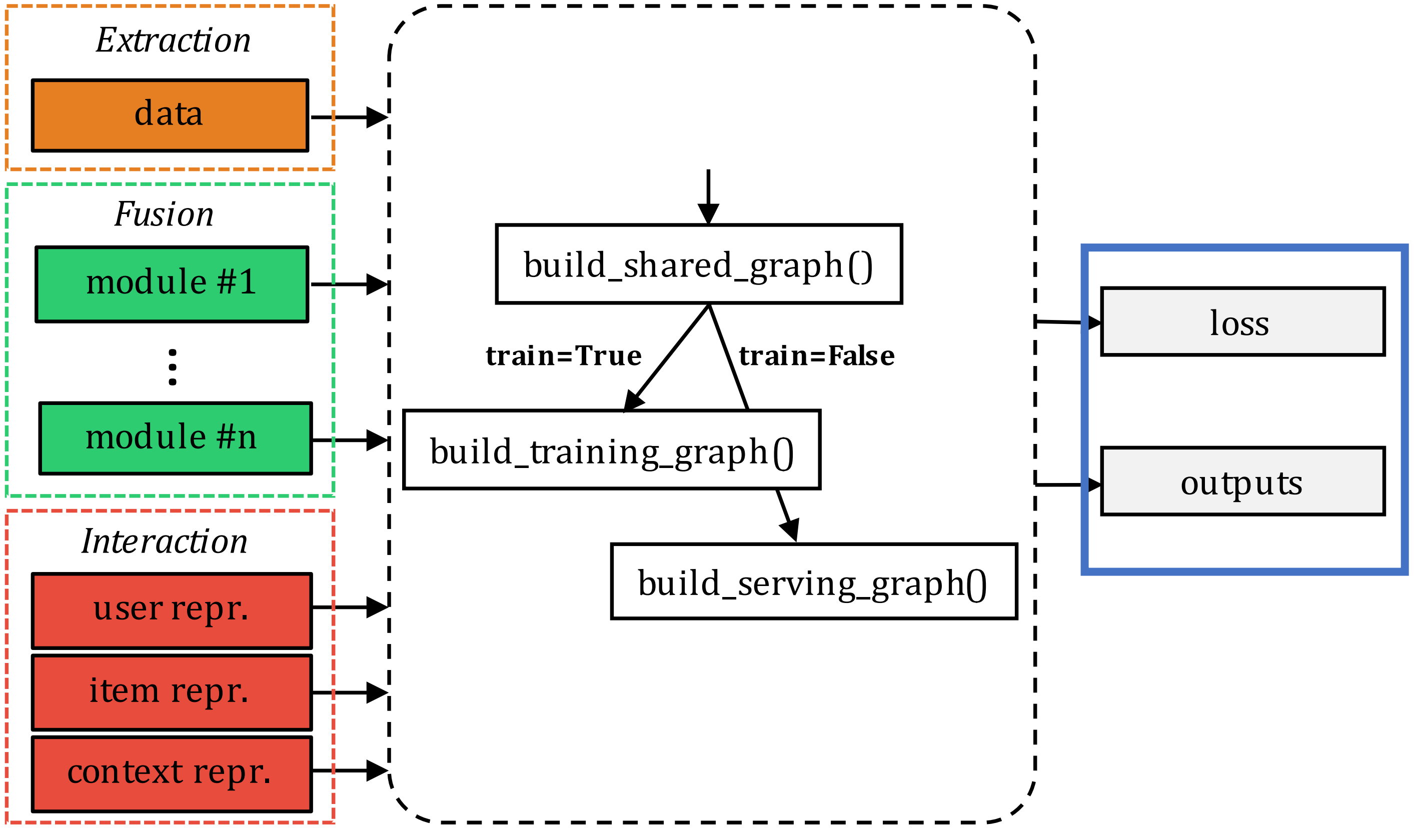
Inside a *Recommender*



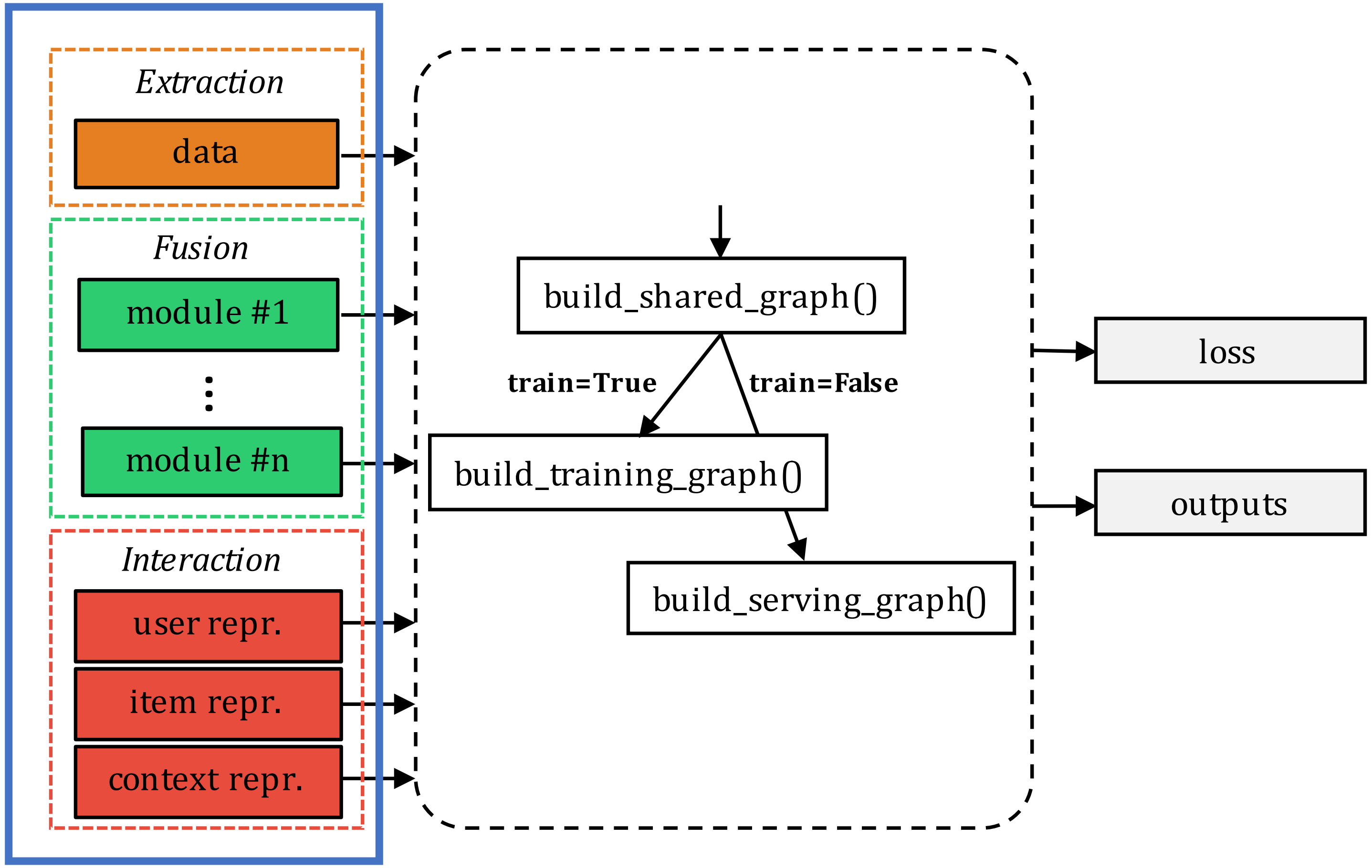
Inside a *Module*



Inside a *Module*



Inside a *Module*



3 Simple use cases

- Conduct model selection (E-commerce book recommendation).
- Develop new algorithms -- *brief*
- Compare modular and monolithic implementations.

Two kinds of model selection

structure selection: what data traces to incorporate and how

module selection: select best modules given a structure

Amazon dataset [McAuley et. al. 15]

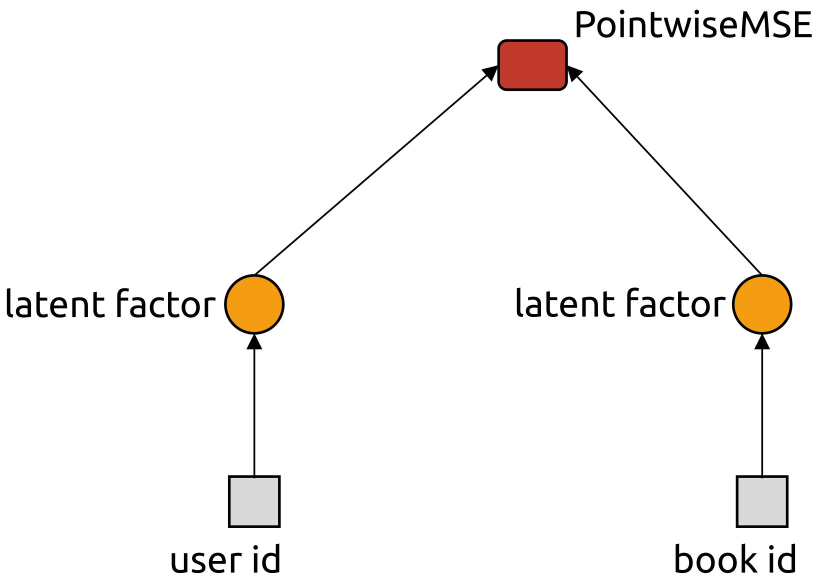
User data: user id & purchases in other categories

Book data: book id & book cover image

Interaction data: user reviews

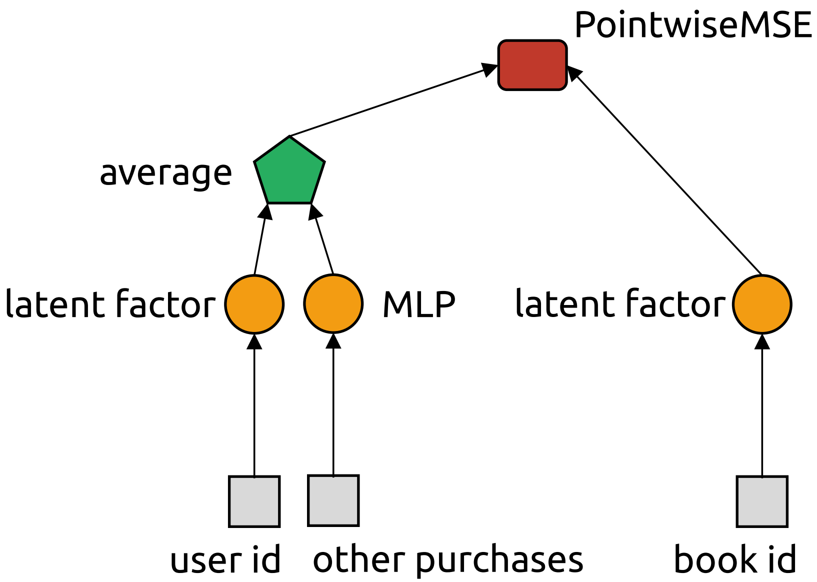
Exp 1. structure selection

PMF



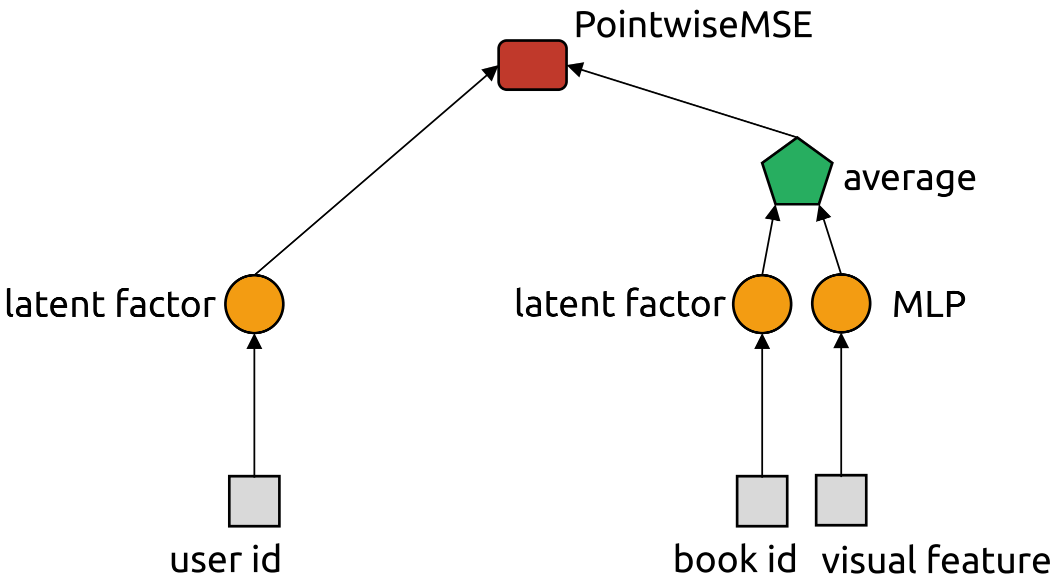
Exp 1. structure selection

UserPMF



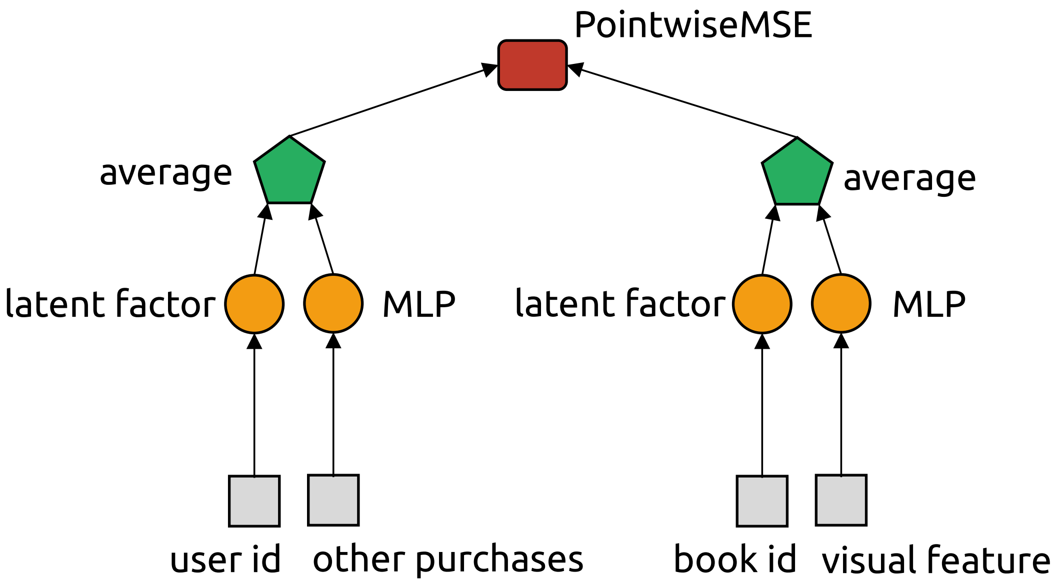
Exp 1. structure selection

VisualPMF



Exp 1. structure selection

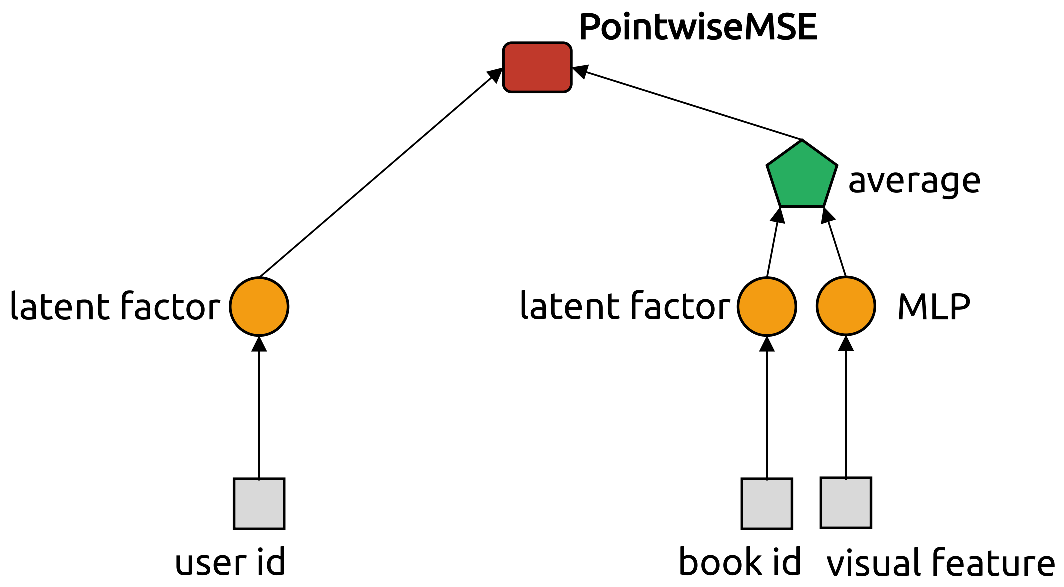
UserVisualPMF



3 **Conduct model selection - E-commerce book recommendation**

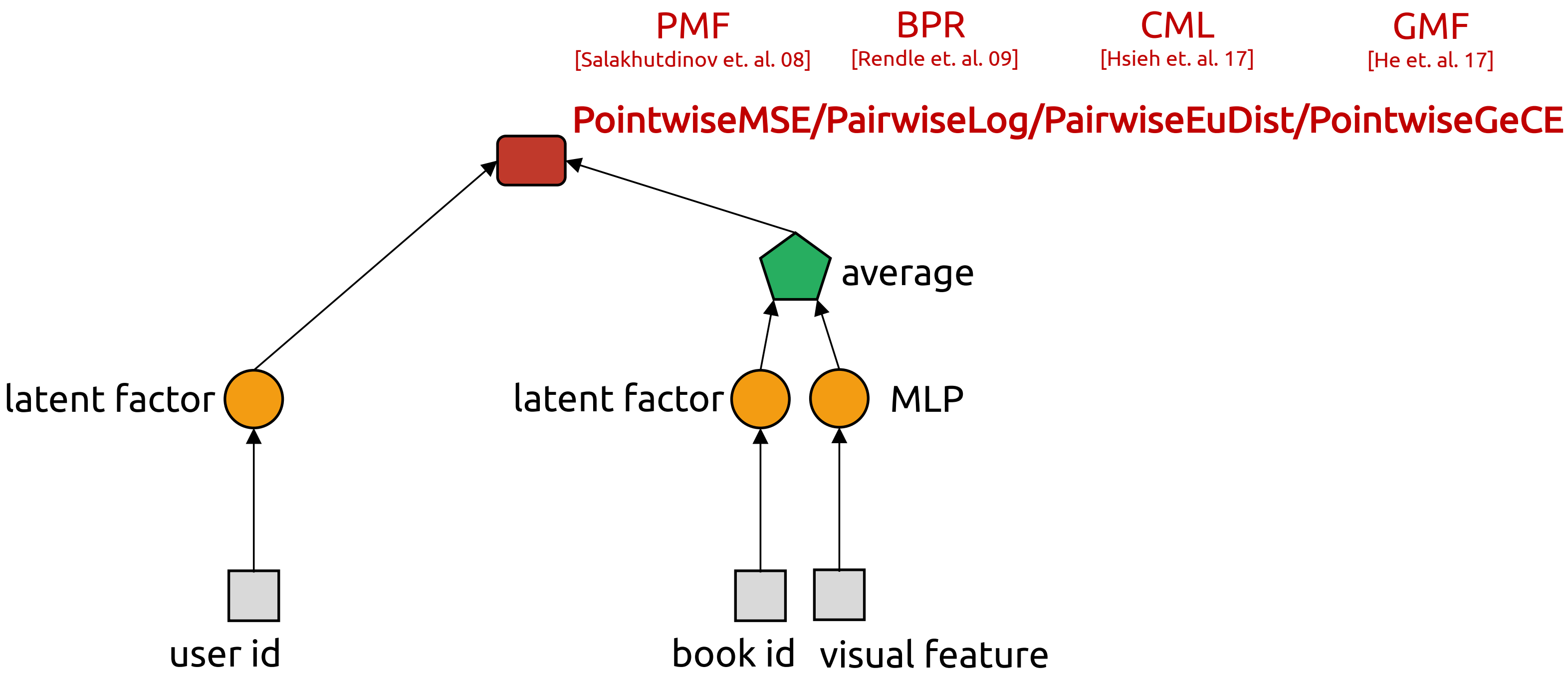
Exp 2. module selection

VisualPMF



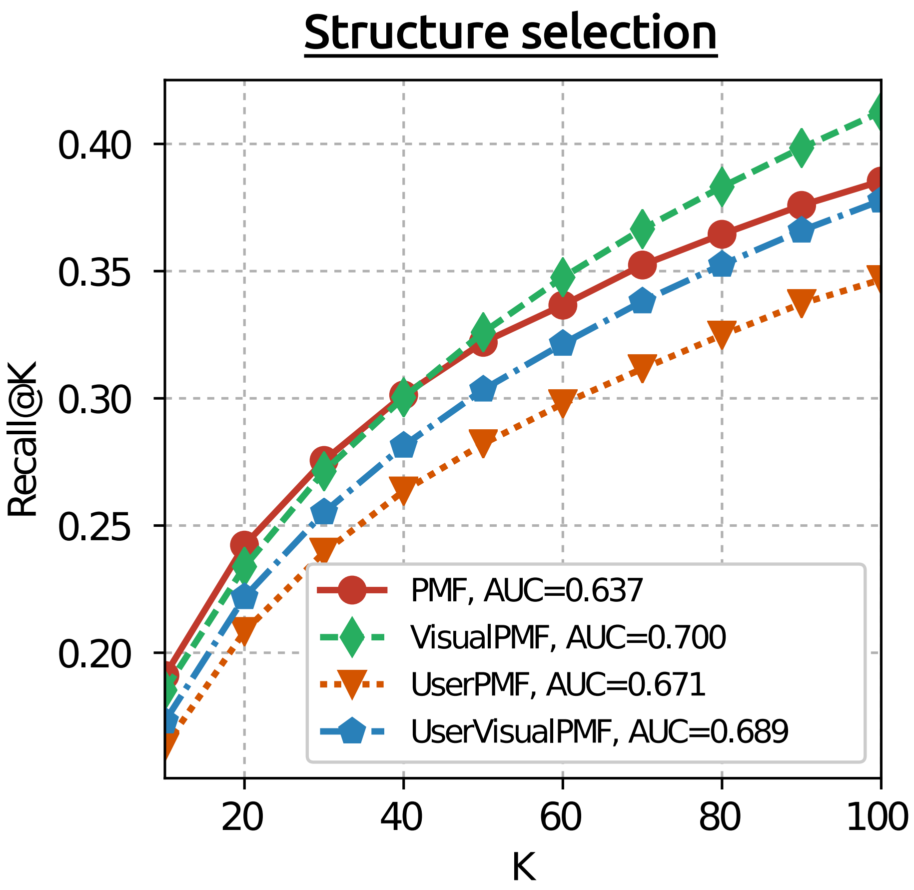
Exp 2. module selection

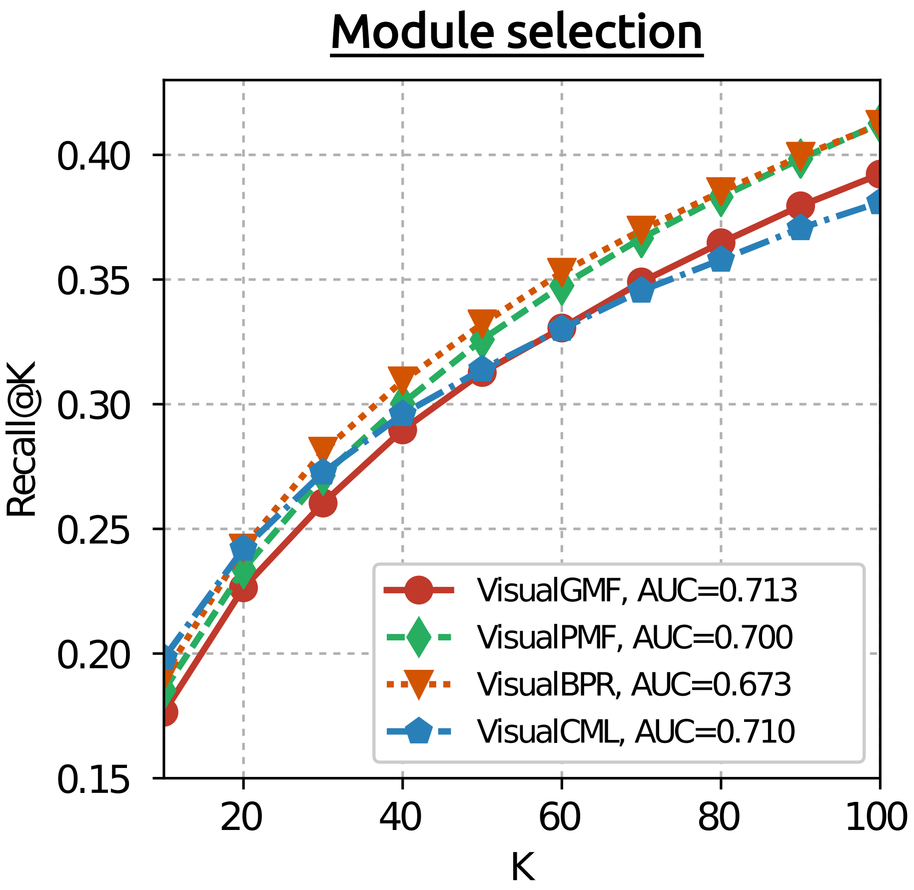
VisualPMF/VisualBPR/VisualCML/VisualGMF



3 Conduct model selection - E-commerce book recommendation

Experimental Results





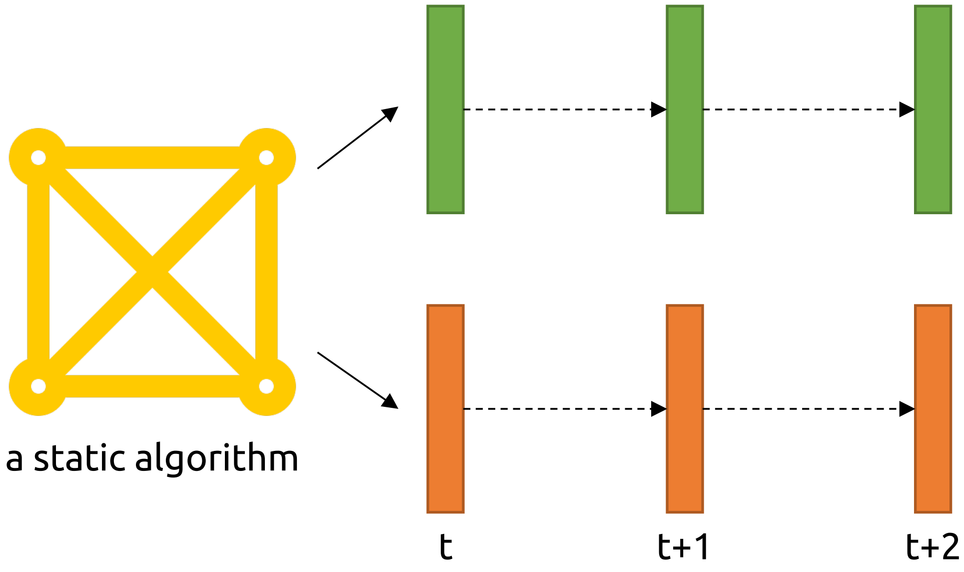
3 Conduct model selection - E-commerce book recommendation

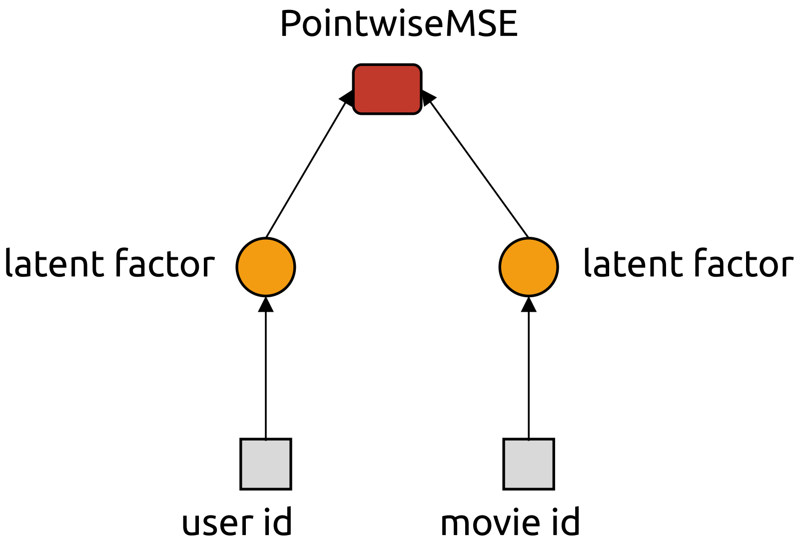
3 Simple use cases

- **Conduct model selection** (E-commerce book recommendation).
- **Develop new algorithms** -- *brief*
- Compare modular and monolithic implementations.

Iterative recommendation

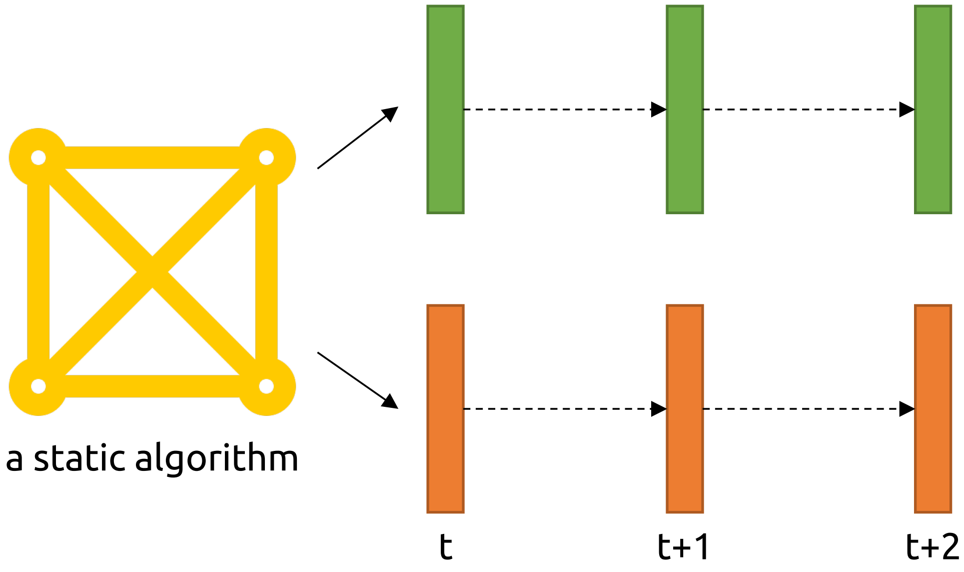
Netflix dataset

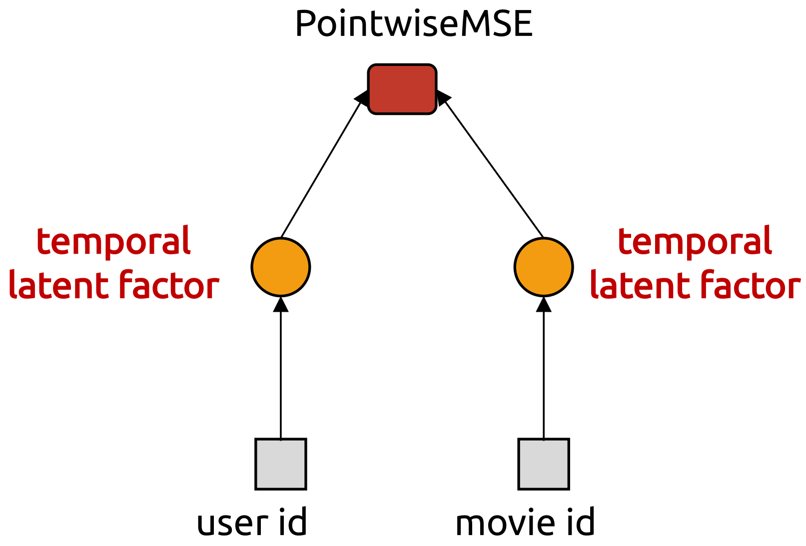




Iterative recommendation

Netflix dataset





6% MSE improvements compared to static model

Takeaways

OpenRec for researchers:

- Demonstrate model generalizability.
- Facilitate comparisons.
- Encourage usage.

OpenRec for practitioners:

- Select models/parameters.
- Adapt state-of-the-art solutions.

Share the same programming model and low-level APIs with Tensorflow/Keras.

Future work

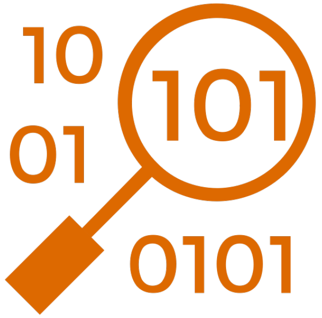
Enriching modules, recommenders and utility functions.

- Your recommendation paper/code.
- Your favorite recommendation algorithms.
- Become a contributor.

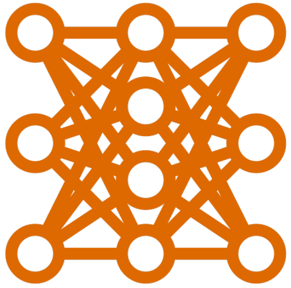
Non-neural network models.

- Tree and graph based models.

Modularity in other domains



Programming language	**Machine language** Specify where to store each bit	**High-level languages** OS, file system, virtual memory	**Modern languages** More abstractions, e.g., save, load.



DNN	**Pre-caffe era** Write CUDA code for any matrix operation	**caffe era** Some layer implementations in C++	**Post-caffe era (Tensorflow, Pytorch, mxnet, etc.)** High-level python API

"You will never succeed in extracting simplicity If don't recognize it is different from mastering complexity."

- *Scott Shenker*

"Modularity based on abstractions is the way things get done"

- *Barbara Liskov*

OpenRec

http://www.openrec.ai

Github link, documents, and tutorials

Longqi Yang

Ph.D. candidate

Computer Science, Cornell Tech, Cornell University

Email: ylongqi@cs.cornell.edu

Web: bit.ly/longqi

Twitter: @ylongqi

Connected Experiences Lab
http://cx.jacobs.cornell.edu/

Small Data Lab
http://smalldata.io/

Funders: