RTB HOUSE

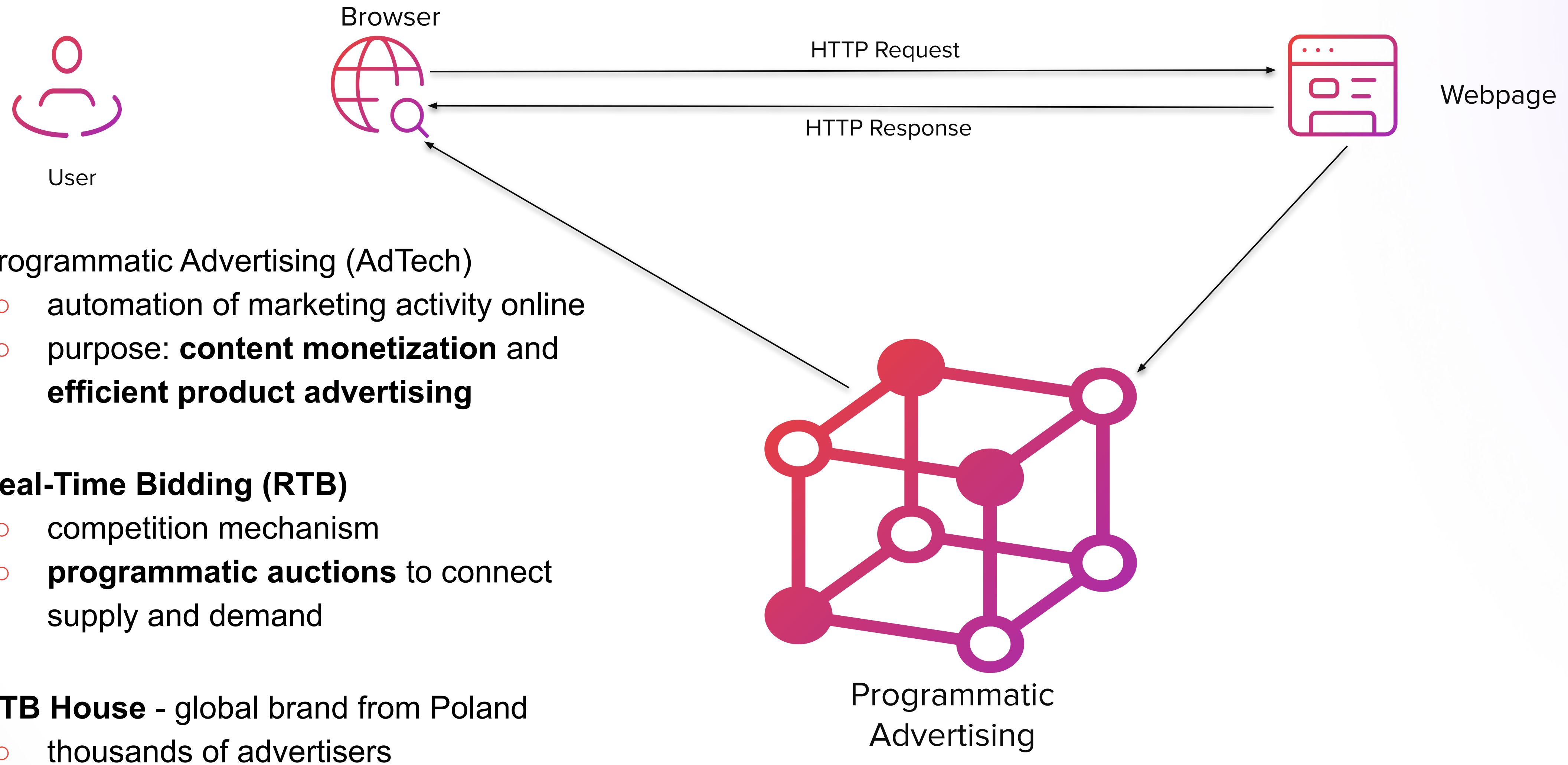
2025-10-15

Sequential Representation Learning for Real-Time Bidding

Mateusz Błajda | Maciej Zdanowicz
ML Researcher | ML Research Team Lead

RTB HOUSE

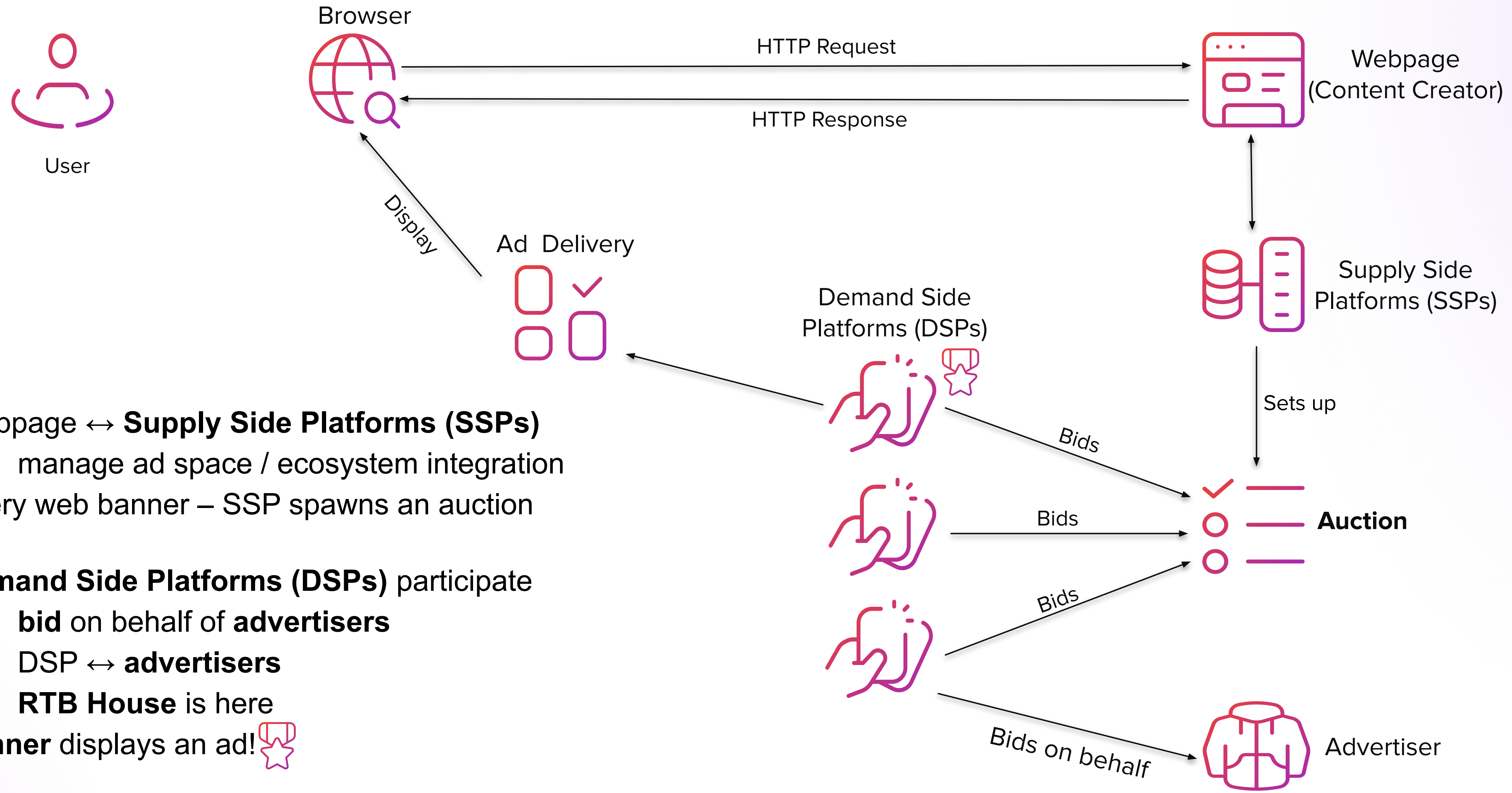
What is Real Time Bidding?



- Programmatic Advertising (AdTech)
    - automation of marketing activity online
    - purpose: **content monetization** and **efficient product advertising**
- **Real-Time Bidding (RTB)**
    - competition mechanism
    - **programmatic auctions** to connect supply and demand
- **RTB House** - global brand from Poland
    - thousands of advertisers
    - **ML-based** techniques

RTB HOUSE =

What is Real Time Bidding?

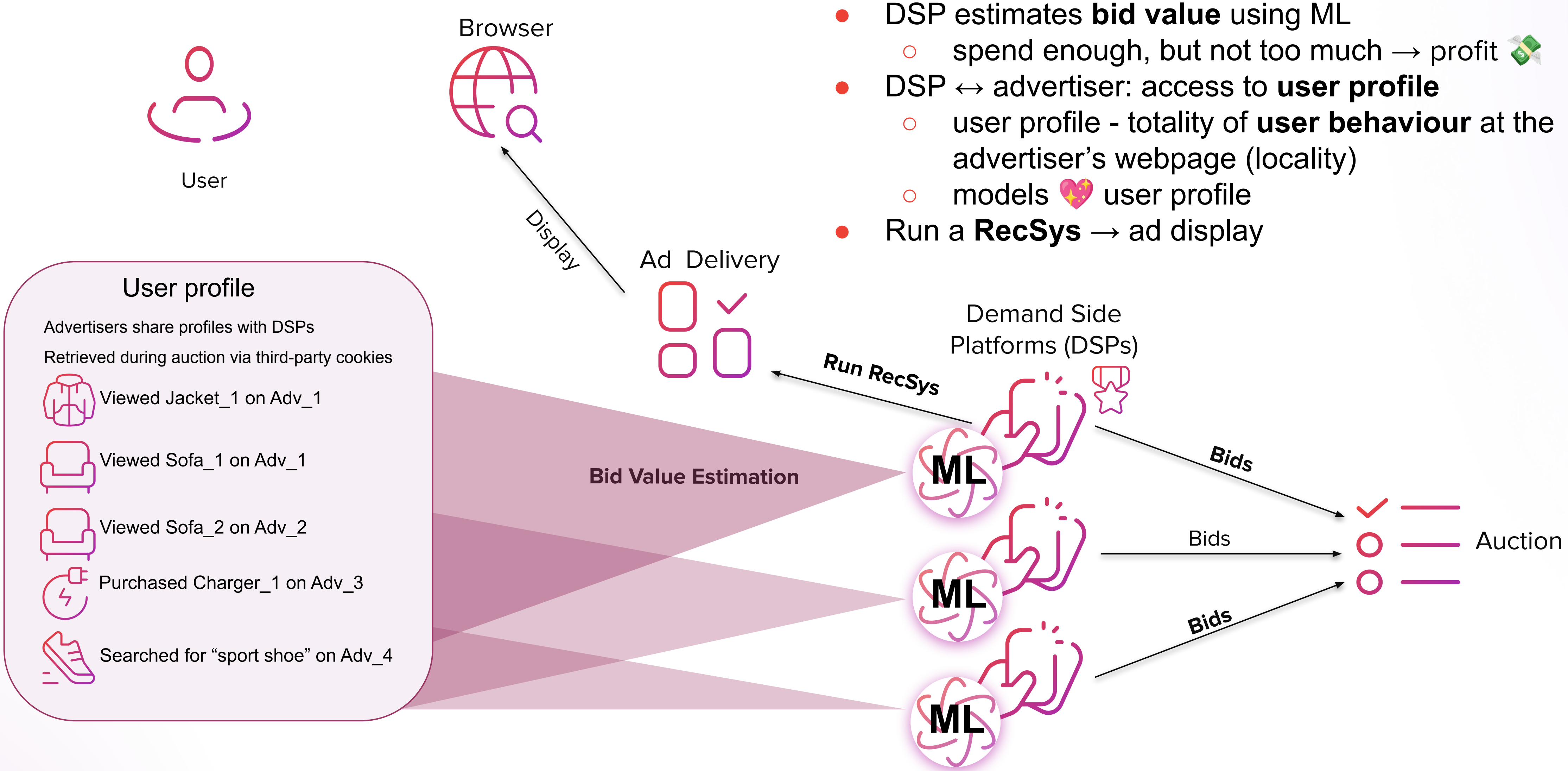


- Webpage ↔ **Supply Side Platforms (SSPs)**
  - manage ad space / ecosystem integration
- Every web banner – SSP spawns an auction
- **Demand Side Platforms (DSPs)** participate
  - **bid** on behalf of **advertisers**
  - DSP ↔ **advertisers**
  - **RTB House** is here
- **Winner** displays an ad!

RTB HOUSE

What is Real Time Bidding?

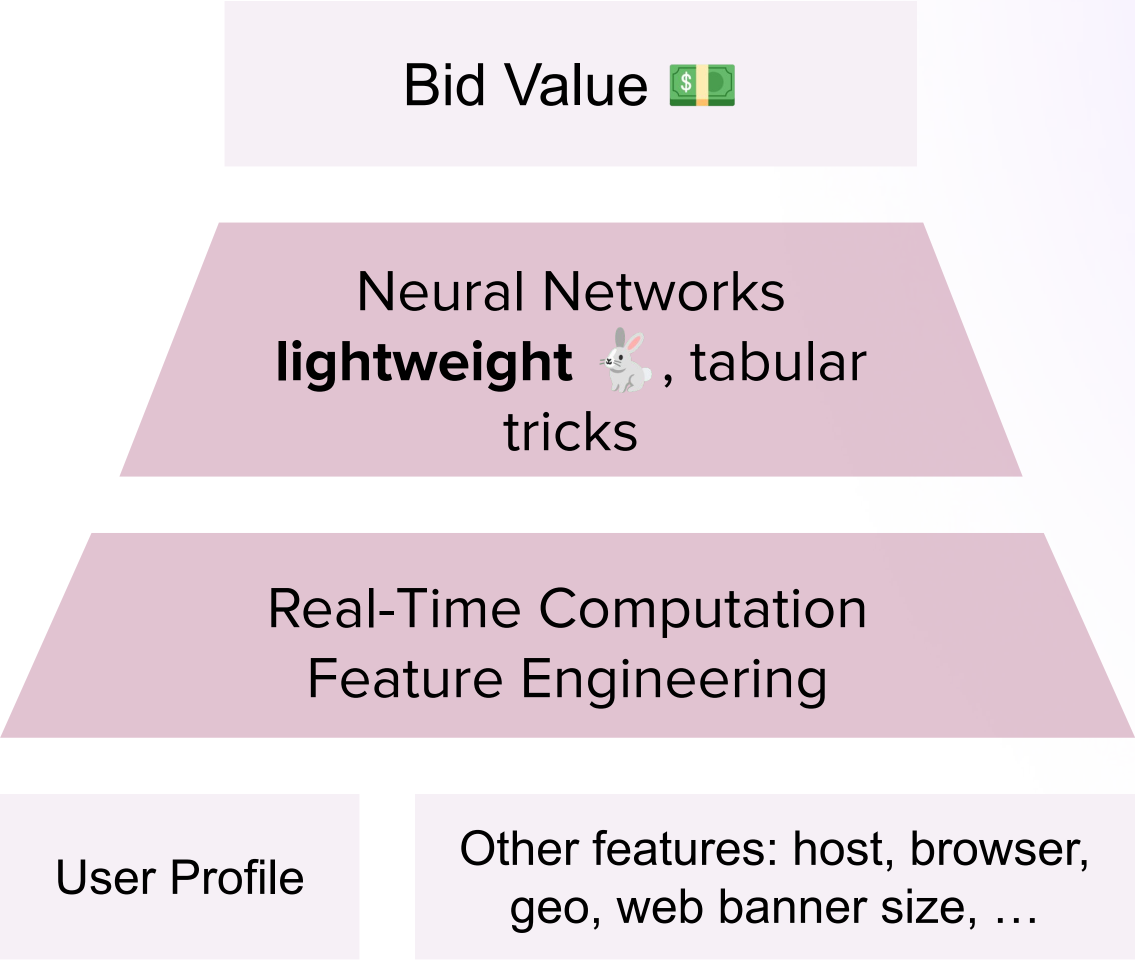
- DSP estimates **bid value** using ML
  - spend enough, but not too much → profit 💸
- DSP ↔ advertiser: access to **user profile**
  - user profile - totality of **user behaviour** at the advertiser's webpage (locality)
  - models 💖 user profile
- Run a **RecSys** → ad display



Flavours of Machine Learning in RTB

Legacy Bid Value Estimation

- Modeling components:
  - **click-through rate** (CTR)
    - user will click?
  - **conversion rate** (CVR)
    - user will buy?
- Guts of the models:
  - SOTA neural networks for tabular data
  - lightweight with some tricks
- **Peak traffic: Hundreds of mln** 🤯 ML eval/sec
  - prohibitive **latency** constraints
  - real-time, synchronous
  - caching (short TTL)



RTBHOUSE

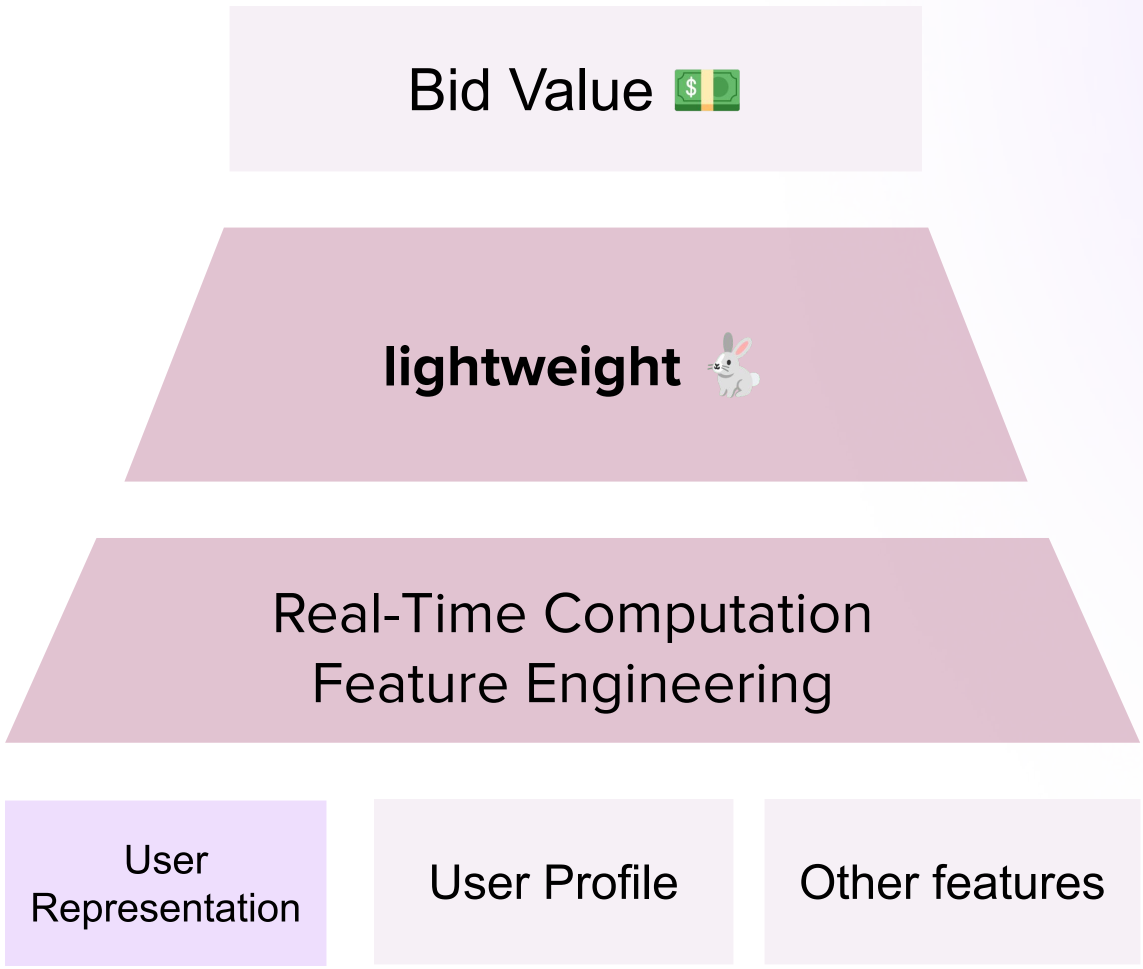
Flavours of Machine Learning in RTB

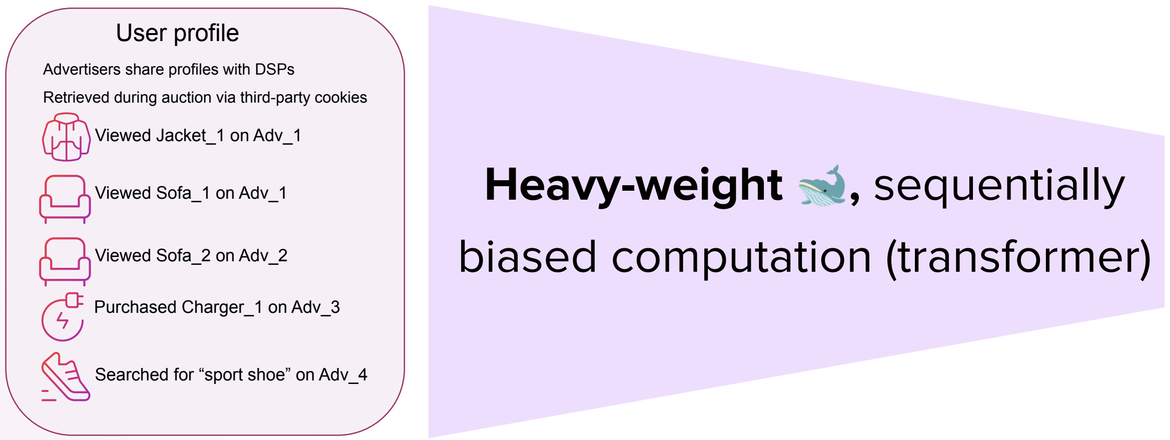
Modern Bid Value Estimation

Goal: Technological Shift!

- Add a **heavyweight, expressive** component
- Leverage sequential structure of UP
- Build **infrastructure** (inference/caching)

- Freshness / compute trade-off (costs)
- User profiles statistics:
    - **hundreds of thousands** changes/sec
    - three orders of magnitude difference





How do we create user representations?

Tags – a base of our Data

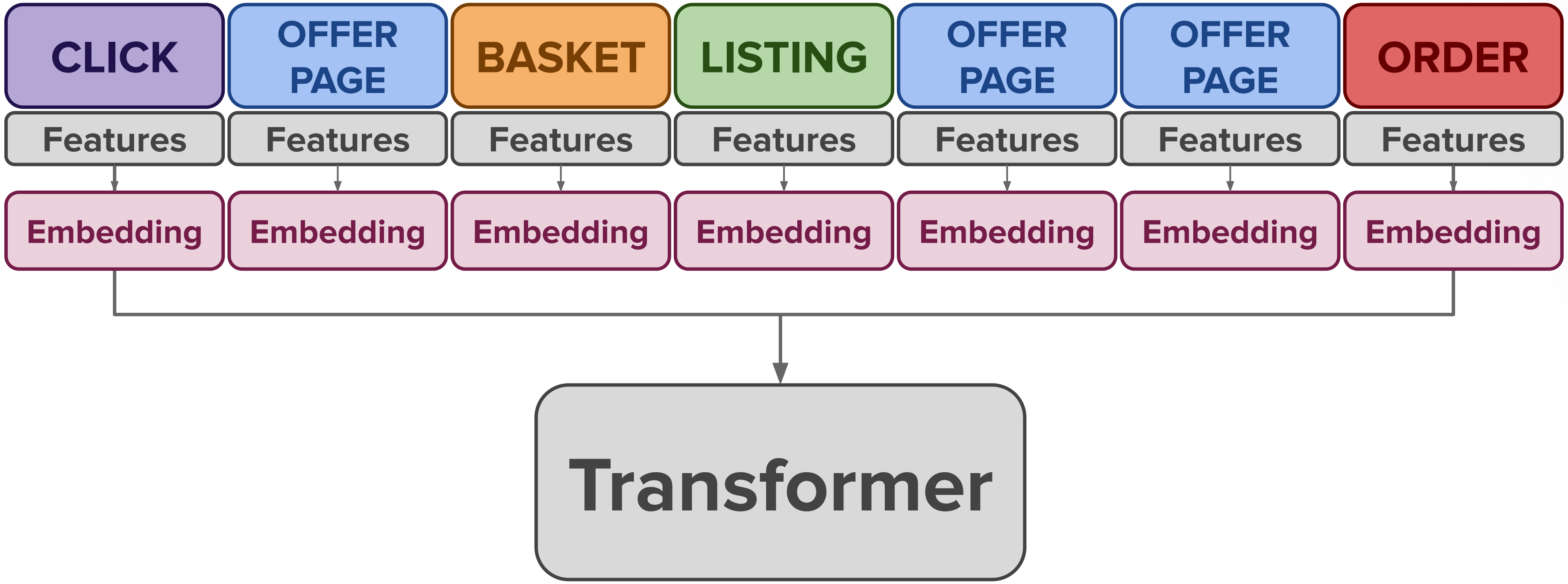
CLICK	User sees an ad and clicks it
OFFER PAGE	User browses a product page
BASKET	User adds a product to a basket
LISTING	User searches for some products
ORDER	User places an order

Additional Data in Tags

- Timestamps
- Offer related info
    - Offer IDs
    - Prices
    - Categories
- Conversion Value
- Etc...

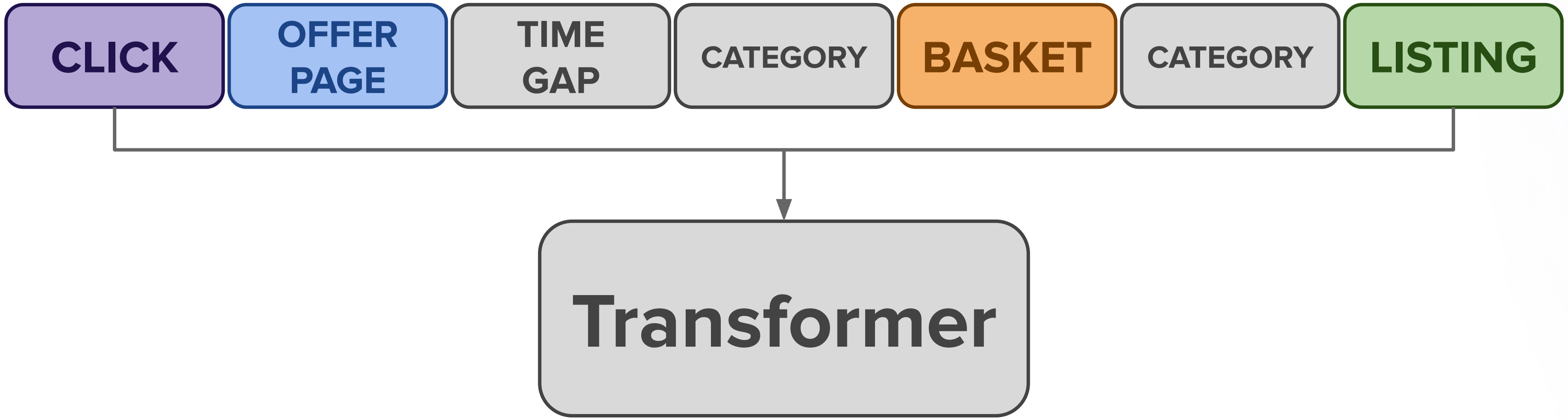
We are training Transformer models from scratch on the users activity data.

Training Transformers on the Tag Data



1 token = 1 tag

Alternative approach

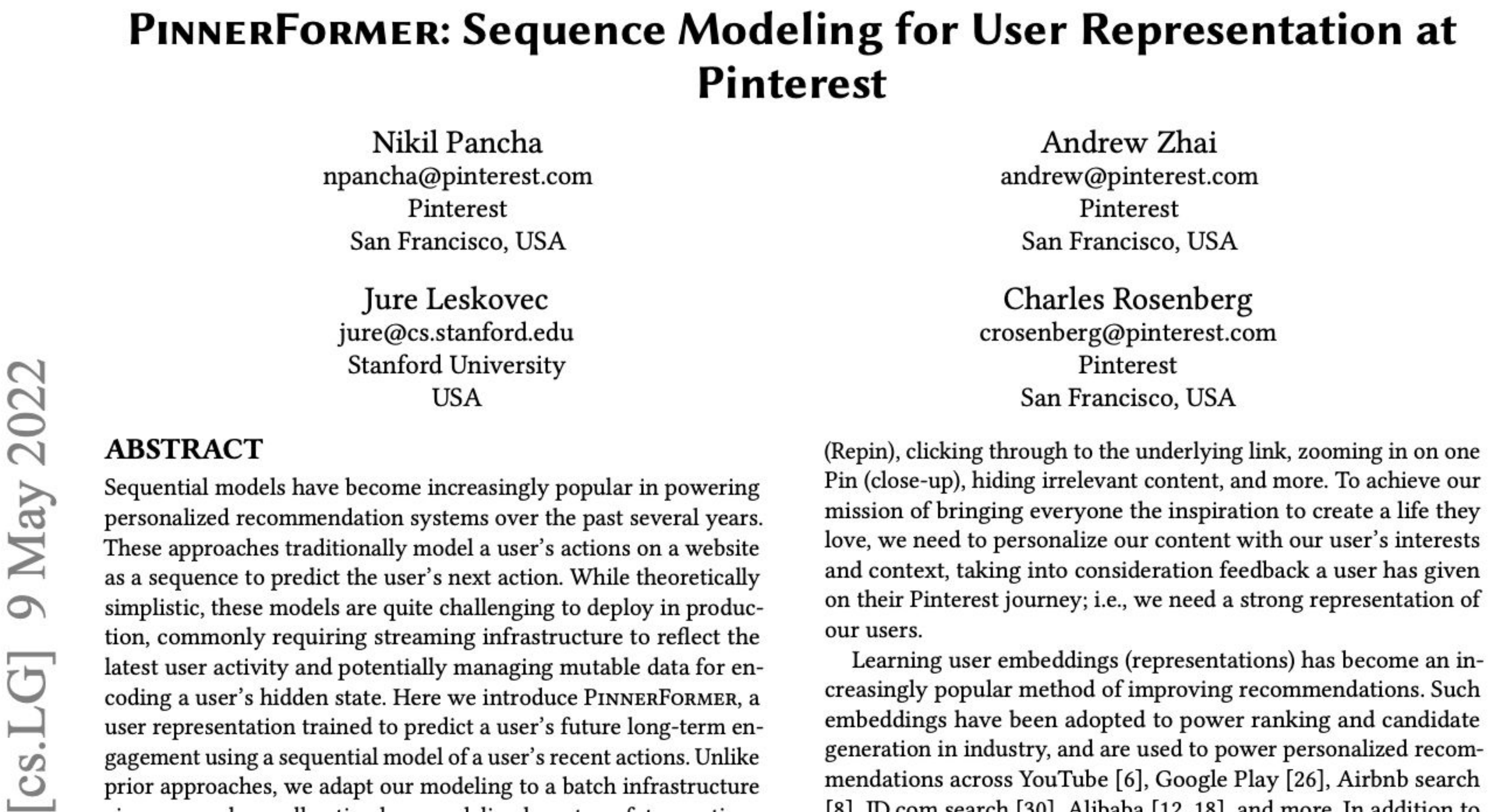


1 token = 1 tag or 1 discrete feature

We decided not to use it, because:

- Models are hard to compare (different dictionary)
- We fit less tags into context

Training: our Inspiration



PinnerFormer: Sequence Modeling for User Representation at Pinterest

Nikil Pancha
npancha@pinterest.com
Pinterest
San Francisco, USA

Andrew Zhai
andrew@pinterest.com
Pinterest
San Francisco, USA

Jure Leskovec
jure@cs.stanford.edu
Stanford University
USA

Charles Rosenberg
crosenberg@pinterest.com
Pinterest
San Francisco, USA

[cs.LG] 9 May 2022

ABSTRACT

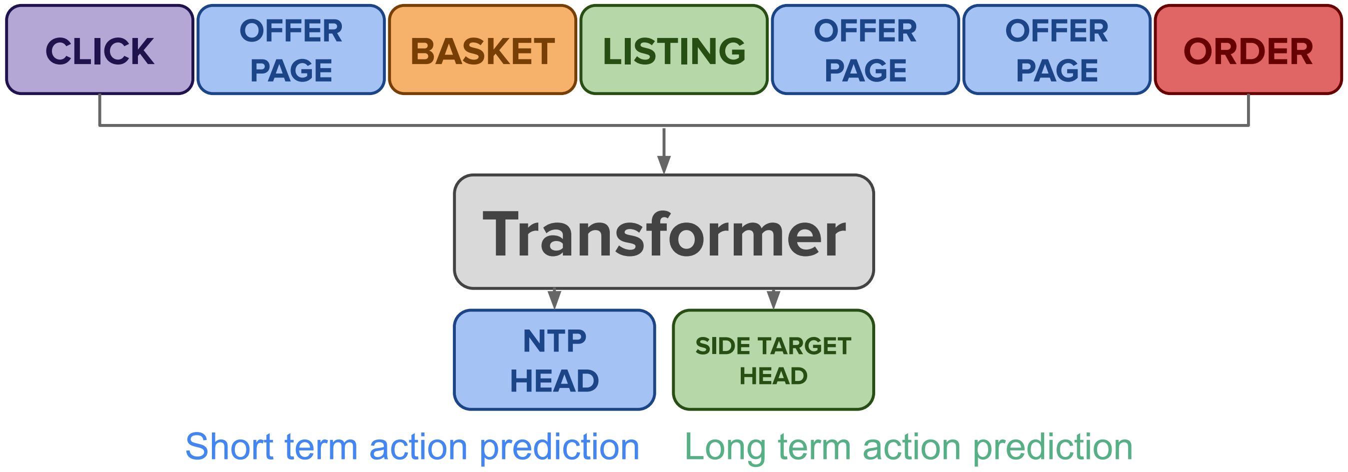
Sequential models have become increasingly popular in powering personalized recommendation systems over the past several years. These approaches traditionally model a user's actions on a website as a sequence to predict the user's next action. While theoretically simplistic, these models are quite challenging to deploy in production, commonly requiring streaming infrastructure to reflect the latest user activity and potentially managing mutable data for encoding a user's hidden state. Here we introduce PinnerFormer, a user representation trained to predict a user's future long-term engagement using a sequential model of a user's recent actions. Unlike prior approaches, we adapt our modeling to a batch infrastructure

(Repin), clicking through to the underlying link, zooming in on one Pin (close-up), hiding irrelevant content, and more. To achieve our mission of bringing everyone the inspiration to create a life they love, we need to personalize our content with our user's interests and context, taking into consideration feedback a user has given on their Pinterest journey; i.e., we need a strong representation of our users.

Learning user embeddings (representations) has become an increasingly popular method of improving recommendations. Such embeddings have been adopted to power ranking and candidate generation in industry, and are used to power personalized recommendations across YouTube [6], Google Play [26], Airbnb search

RTBHOUSE

Long term actions prediction with Side Targets



We combine NTP loss with losses from side targets.

Side Targets:

- Conversion in 30 days
- Log time to next conversion
- Interactions with different categories
- Etc, etc

A TRADEOFF

How often should we update Representations?

Real Time
(evaluation every bid request)

Better, fresher users representations – we know exactly how much time passed since the last user tag

BUT

Much more model evaluations

hundreds of millions per second

Offline
(update on new tag)

Fewer model evaluations

hundreds of thousands per second

(still hundreds of GPUs required)

BUT

Worse users representations – we don’t know when we will get a bid request. There is a big difference between a user who visited a store an hour ago, and one who visited it two weeks ago.

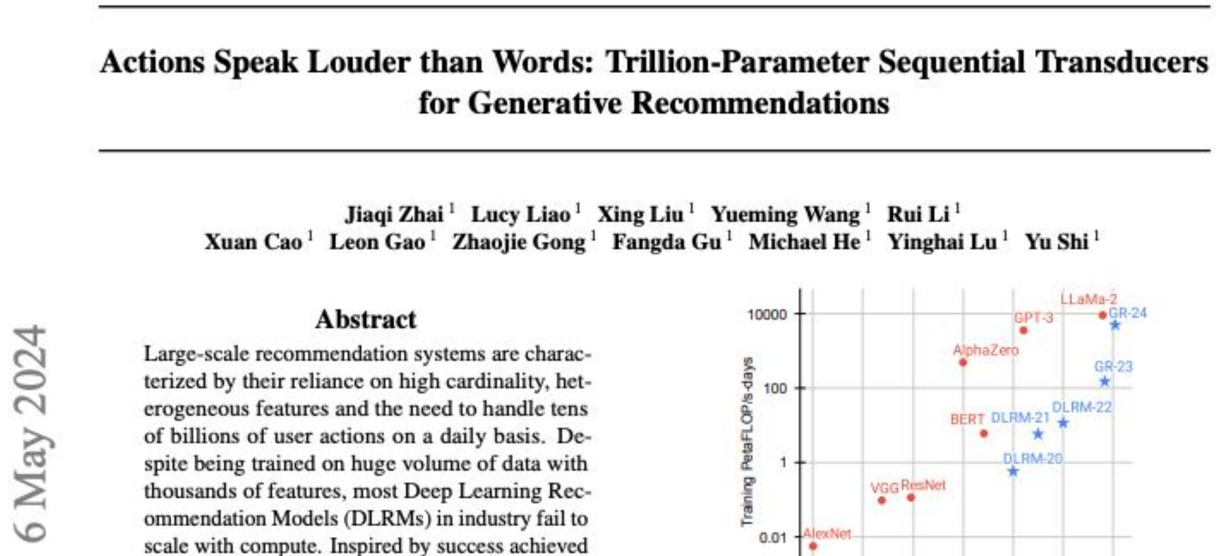
RTBHOUSE

A compromise between freshness and amount of computation

We can simulate a clicks happening after different amounts of time

We mask new “CLICK”s attentions to calculate representations for few different scenarios in one pass

Inspired by this paper ;)



Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations

Jiaqi Zhai[1] Lucy Liao[1] Xing Liu[1] Yueming Wang[1] Rui Li[1]
Xuan Cao[1] Leon Gao[1] Zhaojie Gong[1] Fangda Gu[1] Michael He[1] Yinghai Lu[1] Yu Shi[1]

Abstract

Large-scale recommendation systems are characterized by their reliance on high cardinality, heterogeneous features and the need to handle tens of billions of user actions on a daily basis. Despite being trained on huge volume of data with thousands of features, most Deep Learning Recommendation Models (DLRMs) in industry fail to scale with compute. Inspired by success achieved

6 May 2024

CURRENT RESULTS

We trained a transformer from scratch and integrated it with our CVR model

Offline ROC AUC ↗

1.2%

Legacy Model Context

- Percentage point difference leads to profit increase of 15+% 💸
- Ablation studies of our legacy CVR model:
  - BASKET ➔ 1.0% AUC
  - CLICKS ➔ 0.7% AUC

A/B testing ongoing

Plans for the Future

Scale, Scale, Scale

Scale up to thousands of advertisers. Full scale training will take 20k GPU hours*

*estimated for A100 or similar GPUs

Use representations more broadly

Using our users representations in all of our downstream models:

- Conversion Rate (CR)
- Click-Through Rate (CTR)
- Conversion Value (CV)
- Recommenders

Offer Embeddings

Incorporating Semantic Offer Embeddings into our dataset.

We want to use:

- Images
- Titles & Descriptions
- Categories

THE SCALE OF OUR DATA

Our processed dataset has about 7.5TB
That's 90B of tokens with additional features

More than

4.5x

Of GPT-2 tokens number

About

30%

Of GPT-3 tokens

THE SCALE OF OUR DATA

Creating Semantic Offer Embeddings will be challenging...

We have over

5 Billions

Individual Offers

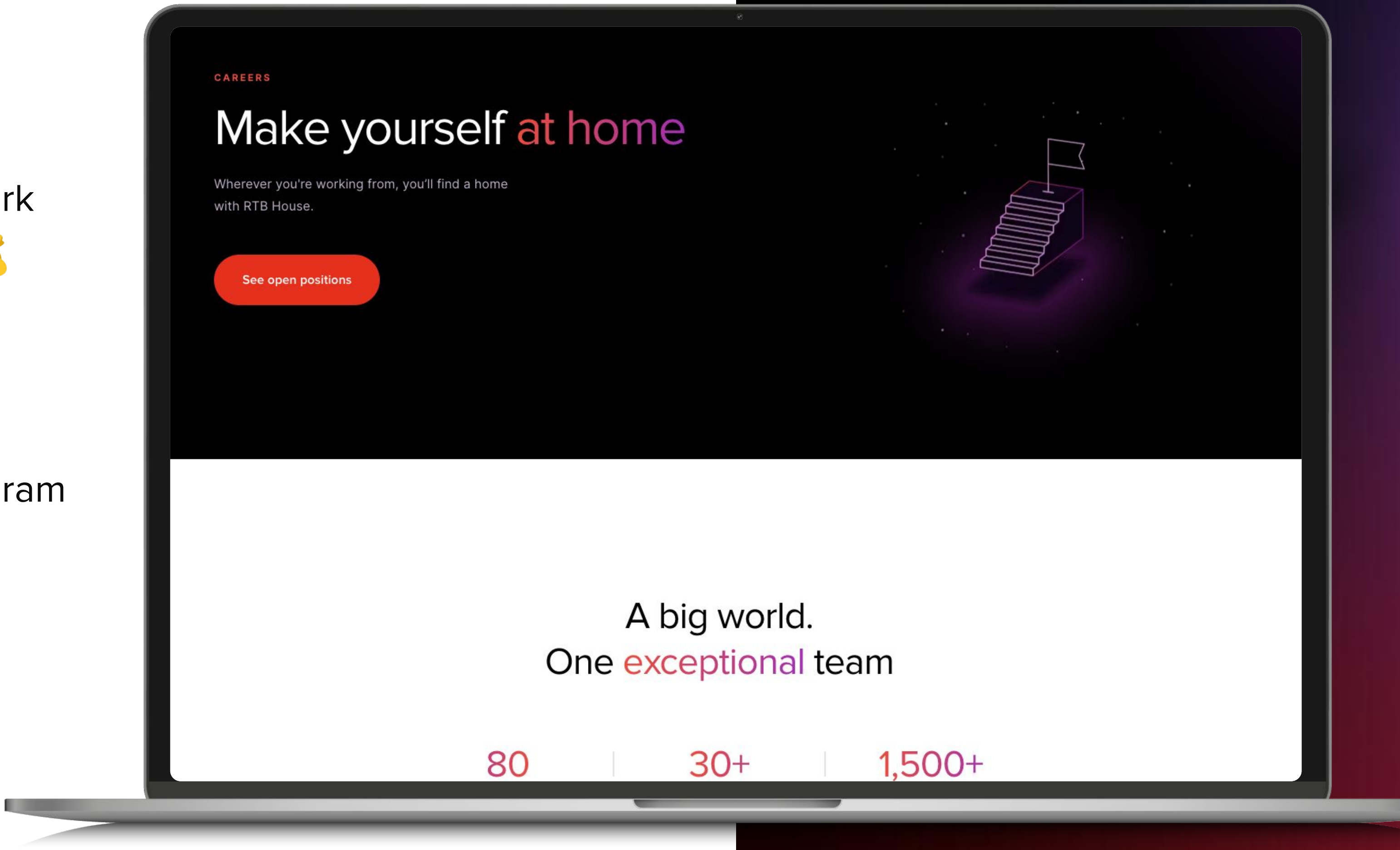
RTB HOUSE

Does that sound like an interesting challenge?

Sequential Representation Learning for Real-Time Bidding

Open positions

- Ownership of meaningful parts of work
- Impact measured in hundreds mln 💰
- Very competitive salary
- Cool get-together destinations 🌋🌴
- Competent and bright colleagues
  - OI, OM, PhDs, ex-Google/Instagram



https://www.rtbhouse.com/careers

RTB HOUSE

Thank you

Get in touch with us!

Mateusz Błajda
ML Researcher
mateusz.blajda@rtbhouse.com

Maciej Zdanowicz
ML Research Team Lead
maciej.zdanowicz@rtbhouse.com