# Deep Dive on Recommendation Algorithms

Three Steps: ML Production Framework | Core Concepts | Relevant Algorithms

# Deep Dive on Recommendation Algorithms

## Stage One: ML Production Framework

# Deep Dive on Monolith from ByteDance

Deep Learning Framework for Real Time Recommendation Systems : https://arxiv.org/pdf/2209.07663

- Issues with using existing DL frameworks i.e. Pytorch/Tensorflow:
  - static parameters are not ideal for recommendation systems with dynamic features
  - platforms such as TikTok gain ~30 million videos **daily, upon which recommender must be retrained**

- Pytorch is designed with batch training separated from serving stage
  - **Not ideal for real-time interaction with customers (RLHF)**

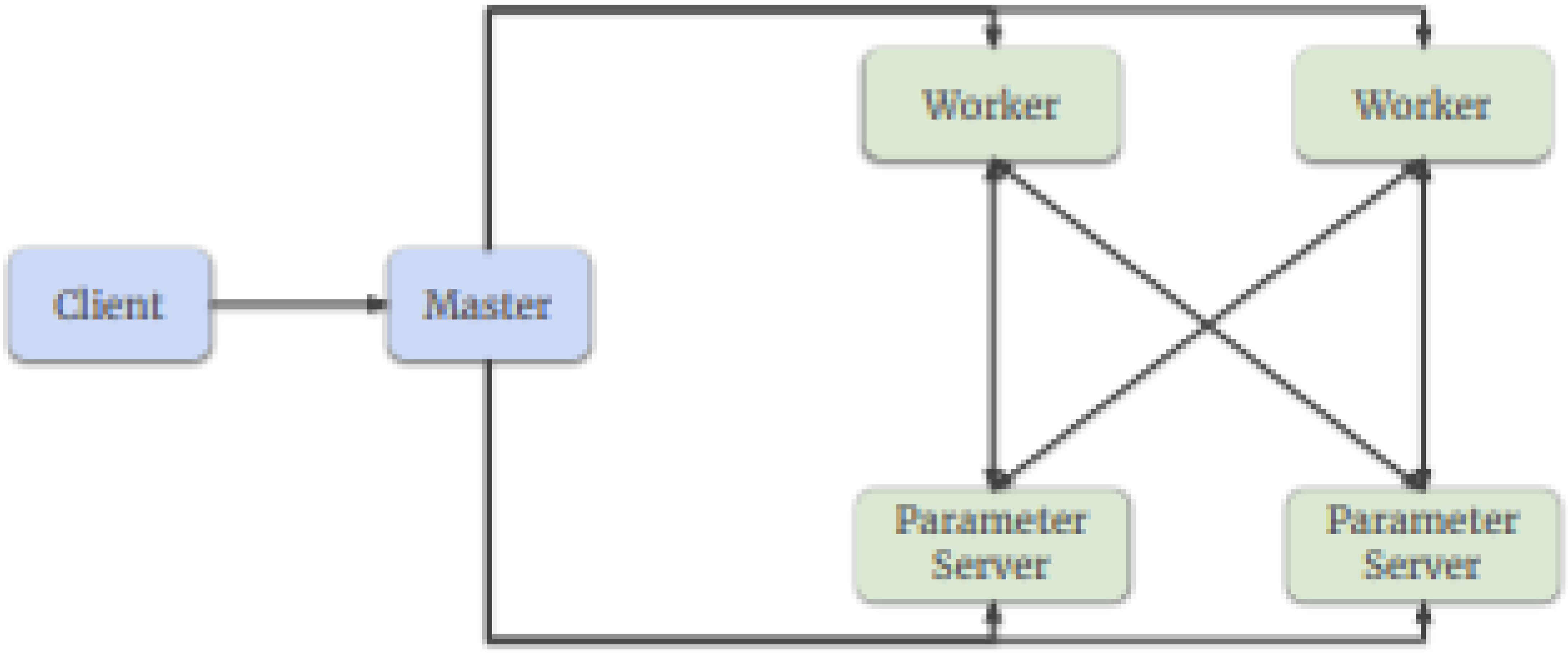
- **Issues Addressed by MONOLITH:**
  - designed for **online** training of recommendation system
  - high fault-tolerance **(collisionless embedding table)**
  - Addresses non-stationary distribution of training data **(Concept Drift)**
  - Size of embedding table scales with users and ranked videos

  - **Challenges the assumption that low-collision hashing for embedding table is harmless to model quality**

# Deep Dive on Monolith from ByteDance

## Deep Learning Framework for Real Time Recommendation Systems

- Monolith architecture designed off of TensorFlow's distributed worker system:



# Deep Dive on Monolith from ByteDance

## Deep Learning Framework for Real Time Recommendation Systems

- Monolith architecture designed off of TensorFlow's distributed worker system:

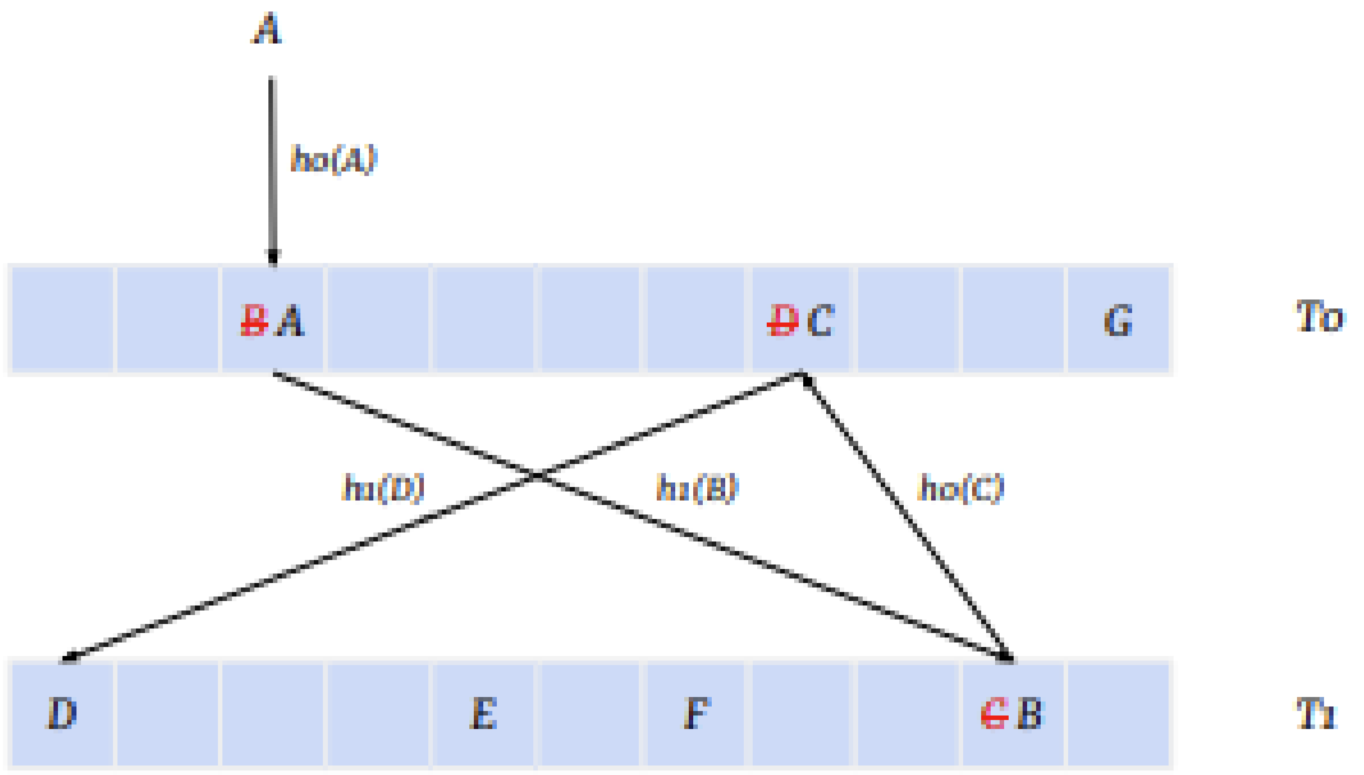


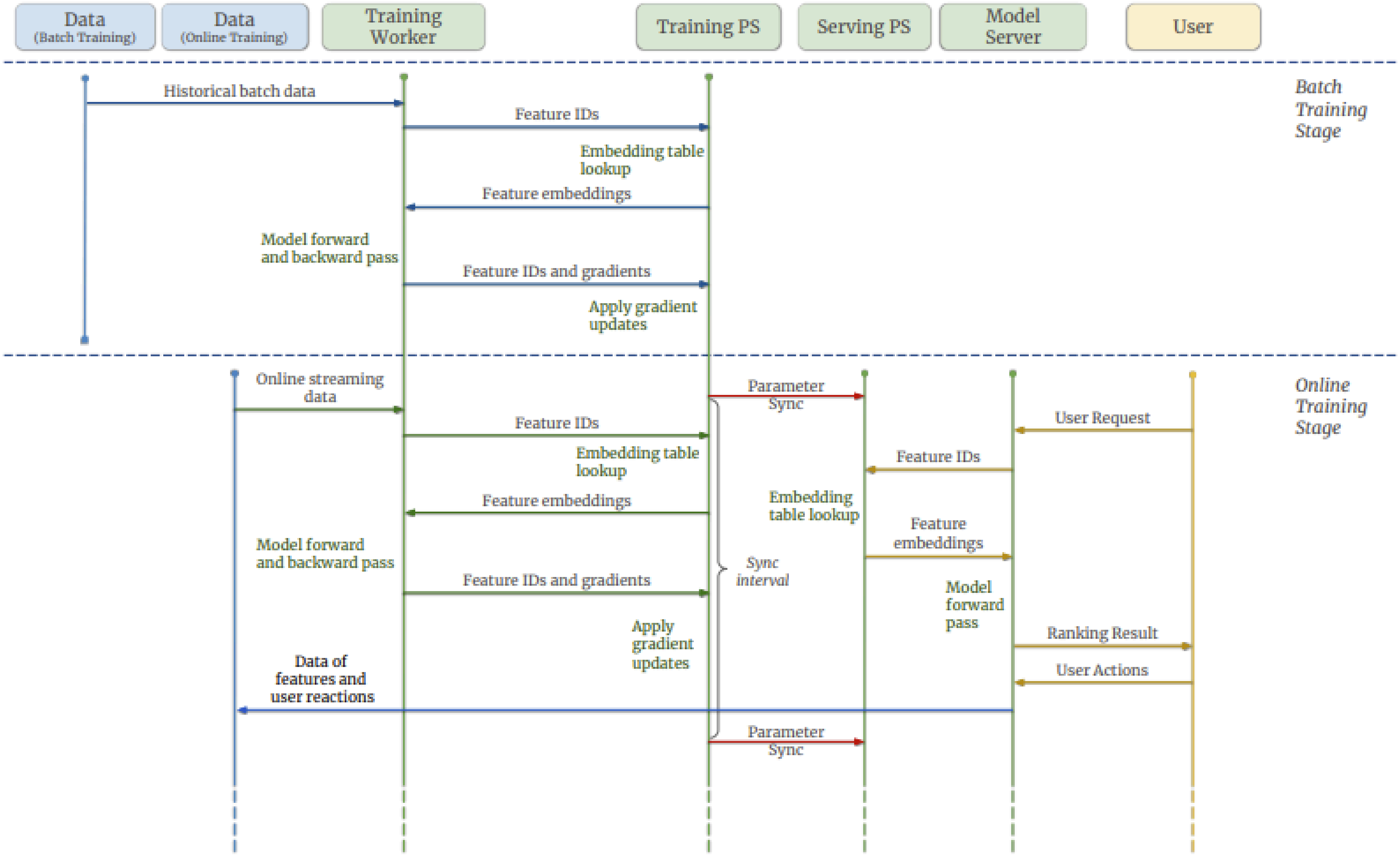

**Figure 3: Cuckoo HashMap.**

  - Cuckoo hashing addresses issue of ID collision associated with Tensorflow
  - Rehashes keys with a suite of hashing functions till elements stabilize

- **Memory Footprint Reduction:**
  - tunable expiry parameter for all IDs
  - Removal of IDs with low occurrence
  - **(how old / relevant is this content)**

# Deep Dive on Monolith from ByteDance

## Deep Learning Framework for Real Time Recommendation Systems




- Cuckoo hashing addresses issue of ID collision associated with Tensorflow
- Rehashes keys with a suite of hashing functions till elements stabilize

- **Memory Footprint Reduction:**
  - tunable expiry parameter for all IDs
  - Removal of IDs with low occurrence
  - **(how old / relevant is this content)**

# Deep Dive on Monolith from ByteDance

## Deep Learning Framework for Real Time Recommendation Systems

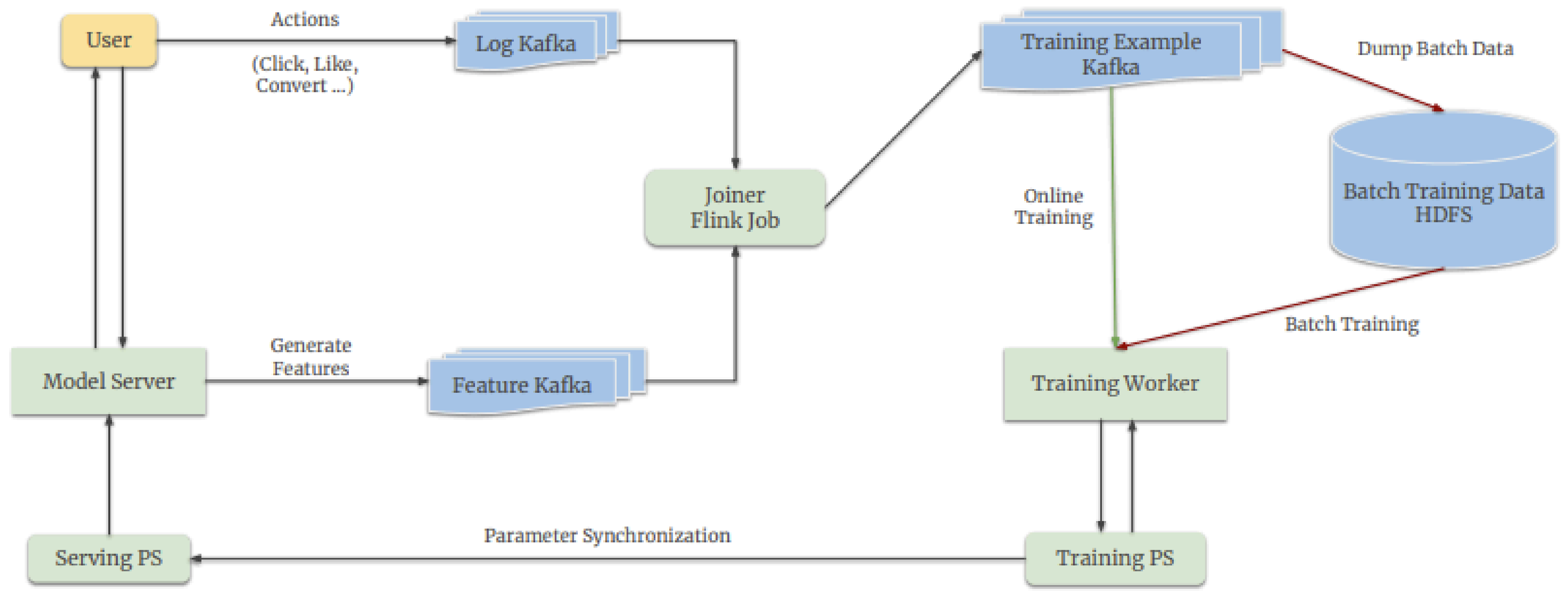



**Figure 4: Streaming Engine.**
*The information feedback loop from [User → Model Server → Training Worker → Model Server → User] would spend a long time when taking the Batch Training path, while the Online Training will close the loop more instantly.*

- **Streaming Engine:**
  - designed for seamless switch between online and offline training
  - Makes use of various open-source fault-tolerant distributed computation libraries:
    - Apache Kafka
    - Apache Flink
    - Apache Hadoop

# Deep Dive on Monolith from ByteDance

## Deep Learning Framework for Real Time Recommendation Systems

- **Performance Considerations:**
  - model quality suffers when parameters acquired from online training are not routinely used to update the actual, deployed model users experience
    - Synchronization steps can involve TB level data transfer
    - Total accuracy vs. network bandwidth

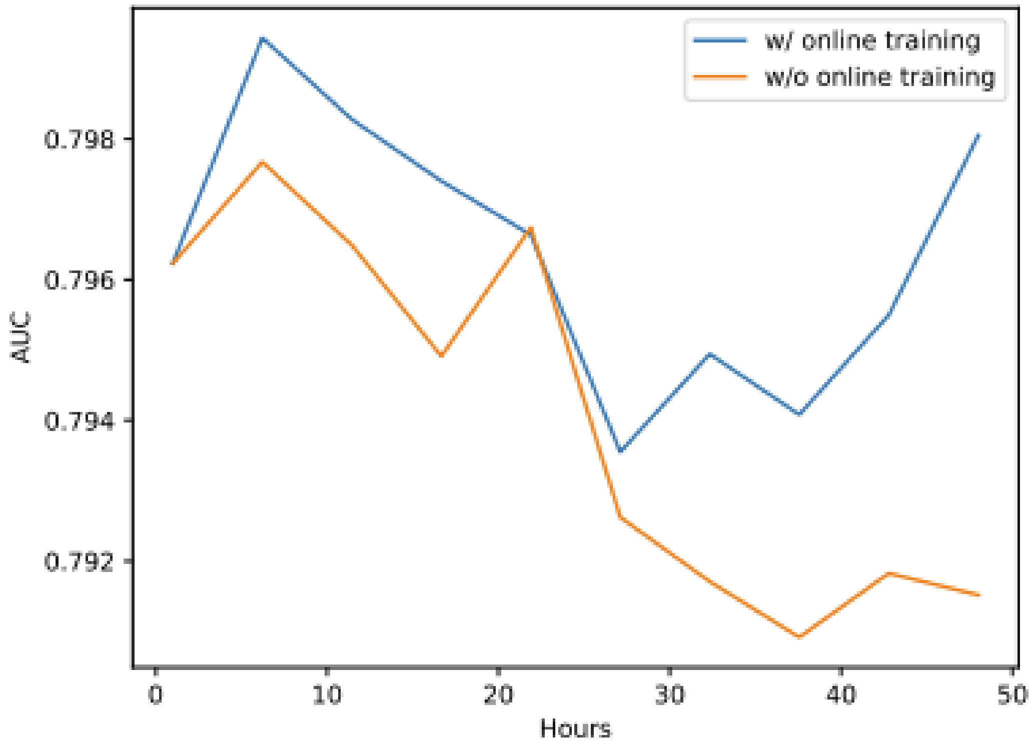
- **Parameter Synchronization:**
  - update process may be long, and client service must not stop during this process
  - synchronization schedules may be different for sparse vs. dense parameters
  - Current recommend synchronization:
    - 1/day for dense parameters
    - 1/minute for sparse parameters (or as much as computation overhead can handle)

# Deep Dive on Monolith from ByteDance

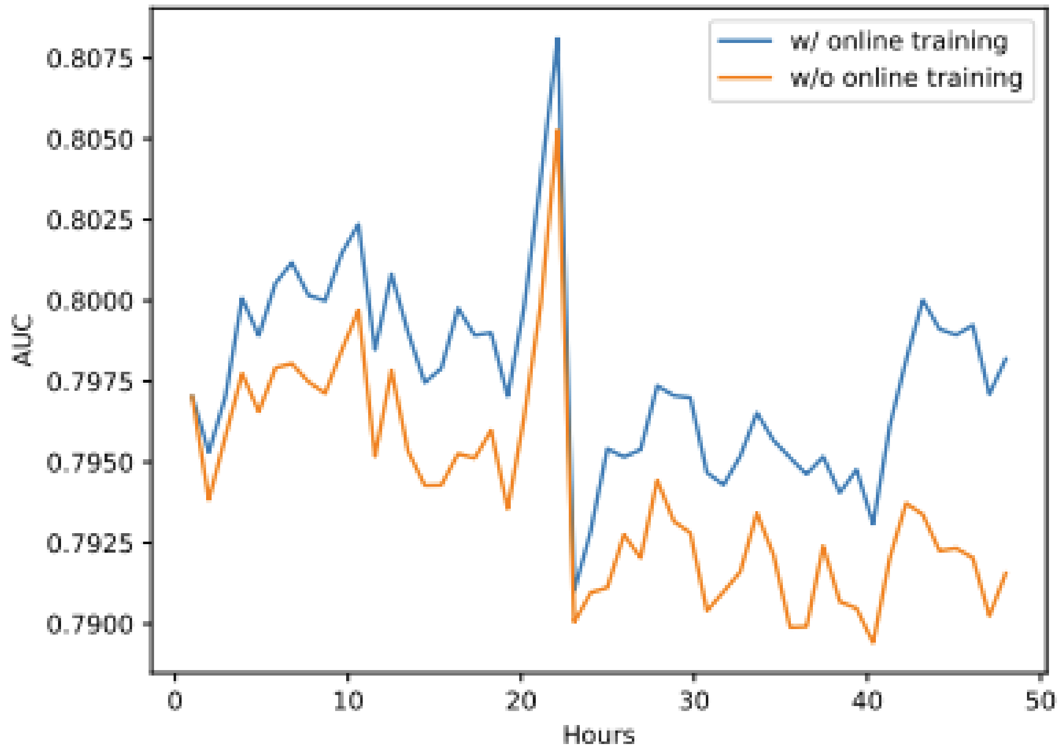
## Deep Learning Framework for Real Time Recommendation Systems

- **Note**: decreased performance gap between online training and offline training as synchronization frequency between training PS and PS servers is increased for parameters

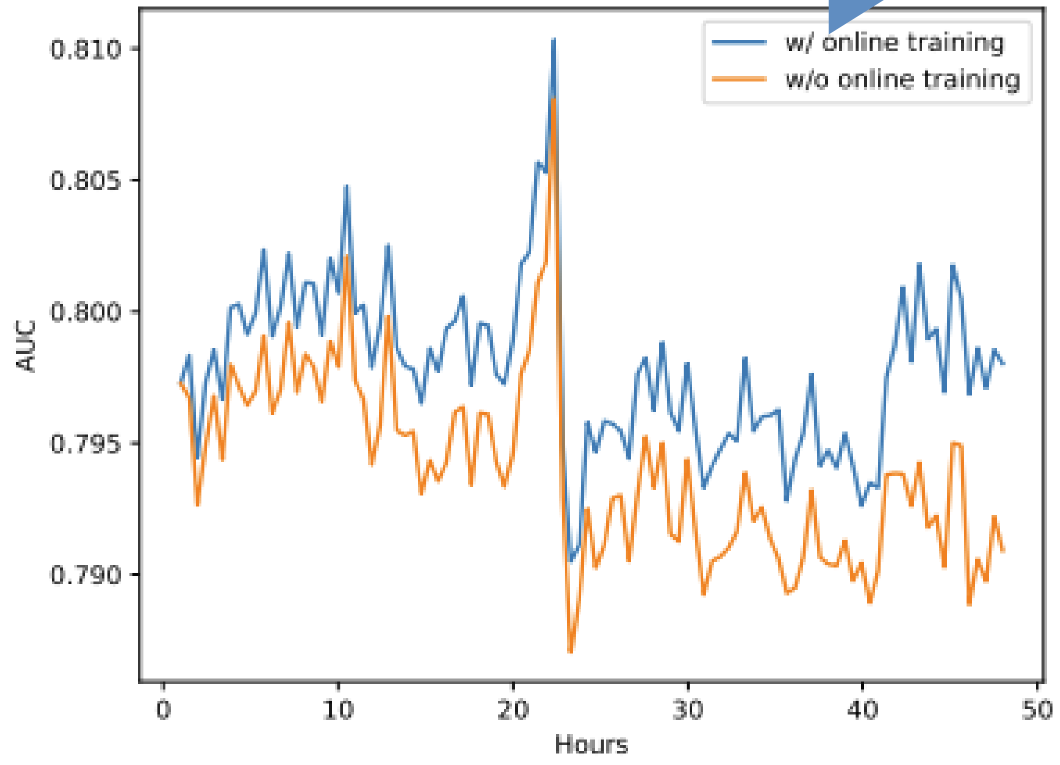
Sync Interval 5 hr
Sync Interval 1 hr
Sync Interval 30 min




(a) Online training with 5 hrs sync interval




(b) Online training with 1 hr sync interval




(c) Online training with 30 min sync interval

# Deep Dive on Recommendation Algorithms

Stage Two: Core Concepts

# Deep Dive on Recommendation Algorithms

## Stage Two: Core Concepts

- Satiation dynamics for multi-arm contextual bandit: **rebounding bandits**
  - https://arxiv.org/pdf/2011.06741
  - **Models a more natural decay in bandit selection over time**

- Real-time bidding mechanisms for recommendation systems:
  - Facebook AI Research
  - Utilizes a contextual bandit to solve **Bidding and Ranking Together (BART)**
  - **General assumption:** outcome for one user session has no consequence on future

- Assumptions on Human Preferences in Multi-armed Bandits (MAB)
  - https://dl.acm.org/doi/pdf/10.1145/3544548.3580670
  - Provides a flexible framework for experimental A/B testing of MAB algorithms with humans

- Deep factorization models for CTR (click-through rate) Prediction
  - https://arxiv.org/pdf/1703.04247
  - Matrix factorization model used to learn feature interactions (high & low level)

# Deep Dive on Recommendation Algorithms

Stage Two: Core Concepts

- Deep factorization models for CTR (click-through rate) Prediction
  - https://arxiv.org/pdf/1703.04247
  - Matrix factorization model used to learn feature interactions (high & low level)

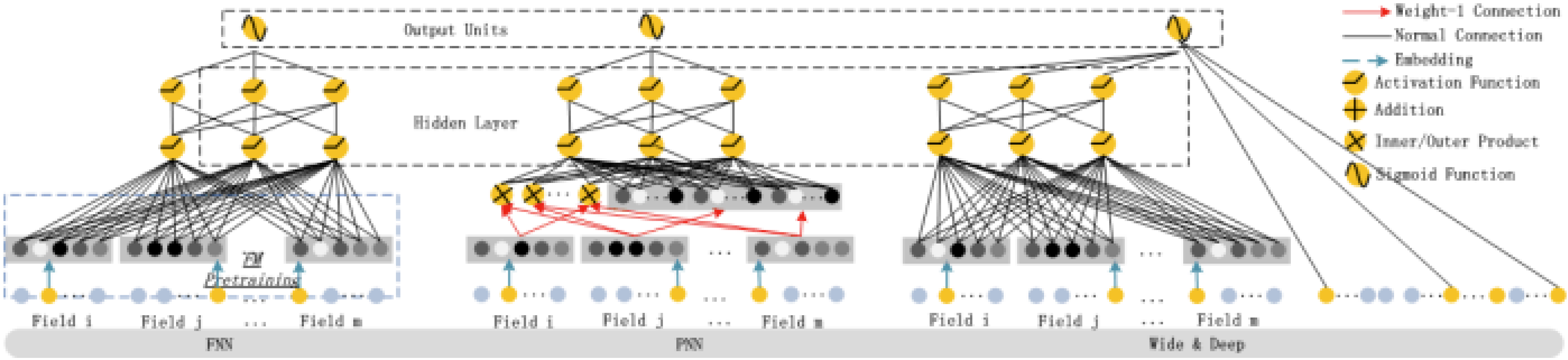



Figure 5: The architectures of existing deep models for CTR prediction: FNN, PNN, Wide & Deep Model

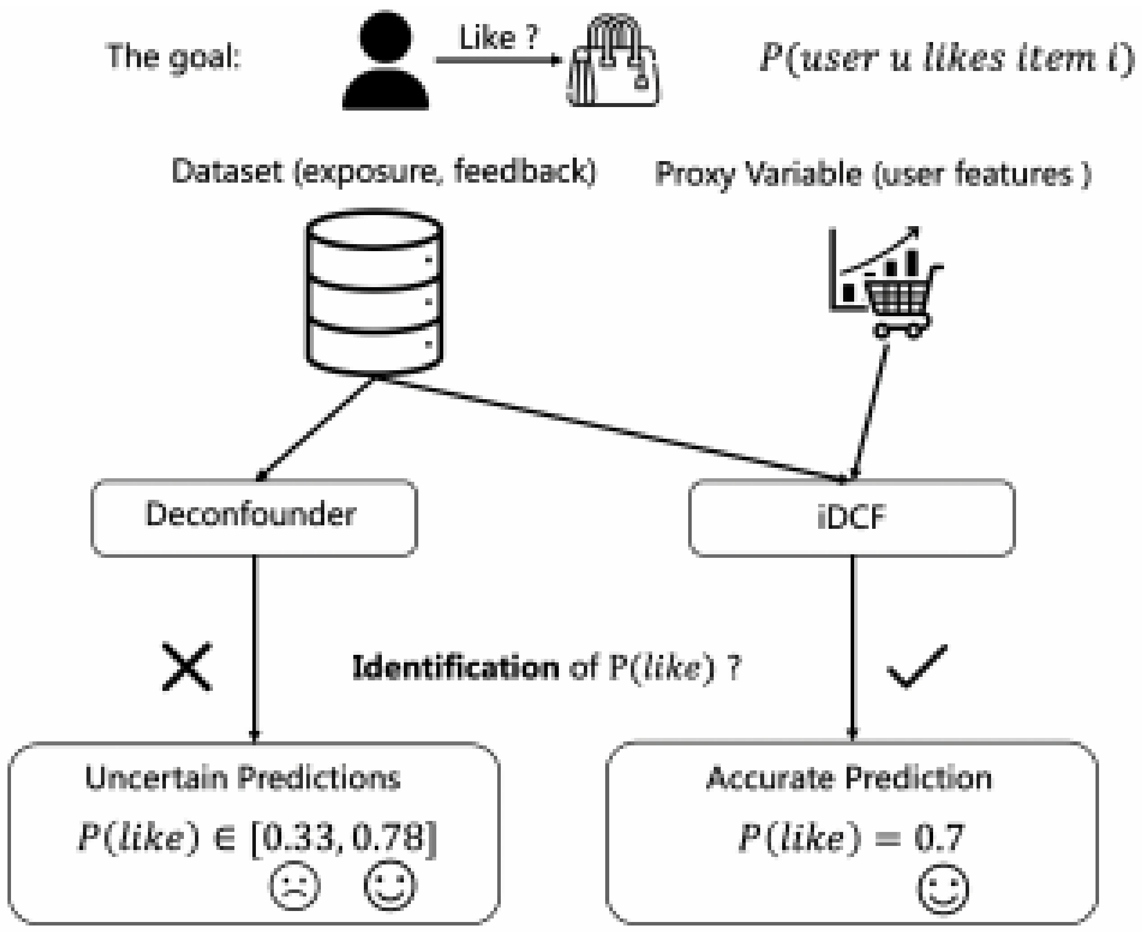
# Deep Dive on Recommendation Algorithms

## Stage Three: State of the Art in Recommendation Algorithms, from Bytedance and Broader Community

# Recent Publications from Bytedance Research Team

## Stage Three: State of the Art in Recommendation Algorithms

- Debiasing Recommendation by Learning Identifiable Latent Confounders **(2023)**
  - https://arxiv.org/pdf/2302.05052
- Disentangled representation for diversified recommendations **(2023)**
  - https://arxiv.org/pdf/2301.05492
- Confidence-aware Fine-tuning of Sequential Recommendation Systems via Conformal Prediction **(2024)**
  - Google Scholar link
- SMLP4Rec: An Efficient all-MLP Architecture for Sequential Recommendations **(2024)**
  - Google Scholar link



# Recent Publications from Bytedance Research Team

## Stage Three: State of the Art in Recommendation Algorithms

- Embedding in Recommender Systems: A Survey **(2023)**
  - Google Scholar Link
- Graph-Based Model-Agnostic Data Subsampling for Recommendation Systems **(2023)**
  - Google Scholar Link

# Recent Publications in Broader Research Community

## Stage Three: State of the Art in Recommendation Algorithms

- Ranking Enhanced Fine-Grained Contrastive Learning for Recommendation **(2024)**
  - https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10446207

- MCRPL: A Pretrain, Prompt, and Fine-tune Paradigm for Non-overlapping Many-to-one Cross-domain Recommendation **(2024)**
  - https://dl.acm.org/doi/pdf/10.1145/3641860

- Entire Chain Uplift Modeling with Context-Enhanced Learning for Intelligent Marketing **(2024)**
  - https://arxiv.org/abs/2402.03379

- Point-of-interest Recommendation using Deep Semantic Model **(2023)**
  - https://dl.acm.org/doi/10.1016/j.eswa.2023.120727