Automated Machine Learning for Recommender Systems

Xiangyu Zhao

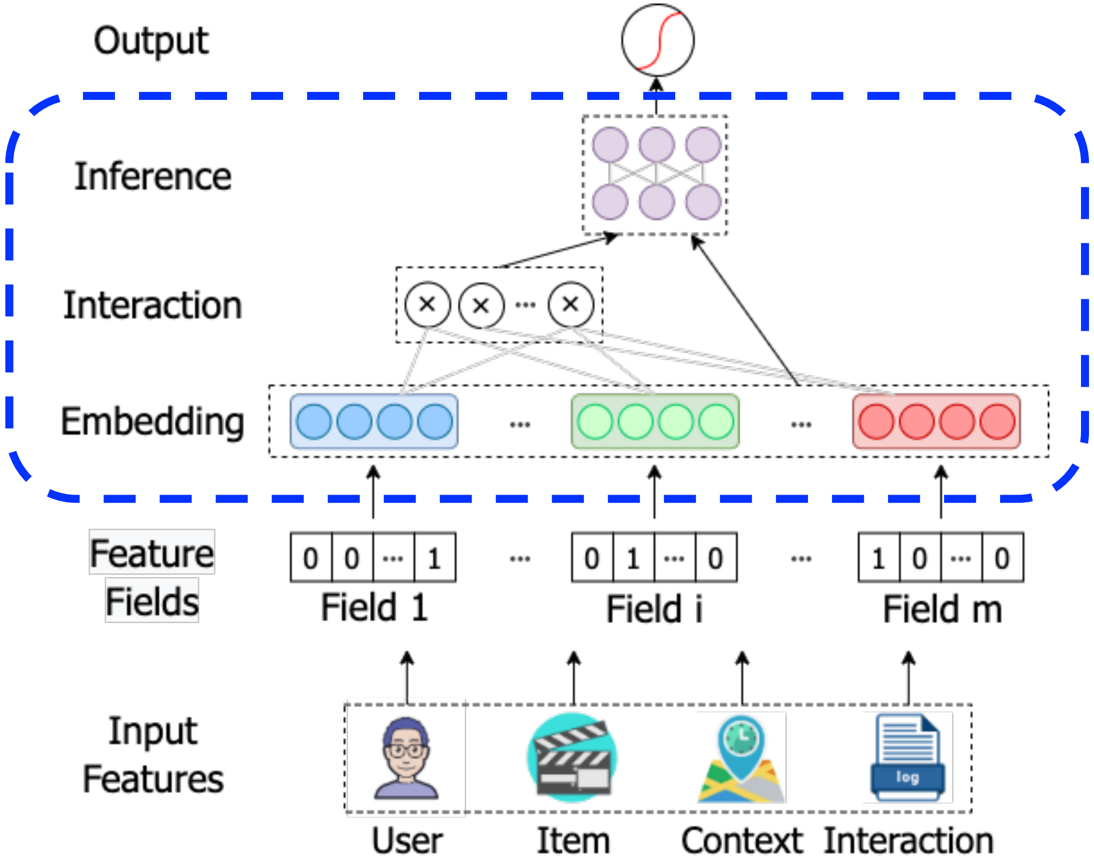
Data Science and Engineering Lab

Michigan State University

www.cse.msu.edu/~zhaoxi35 , zhaoxi35@msu.edu

Deep Recommender Architectures

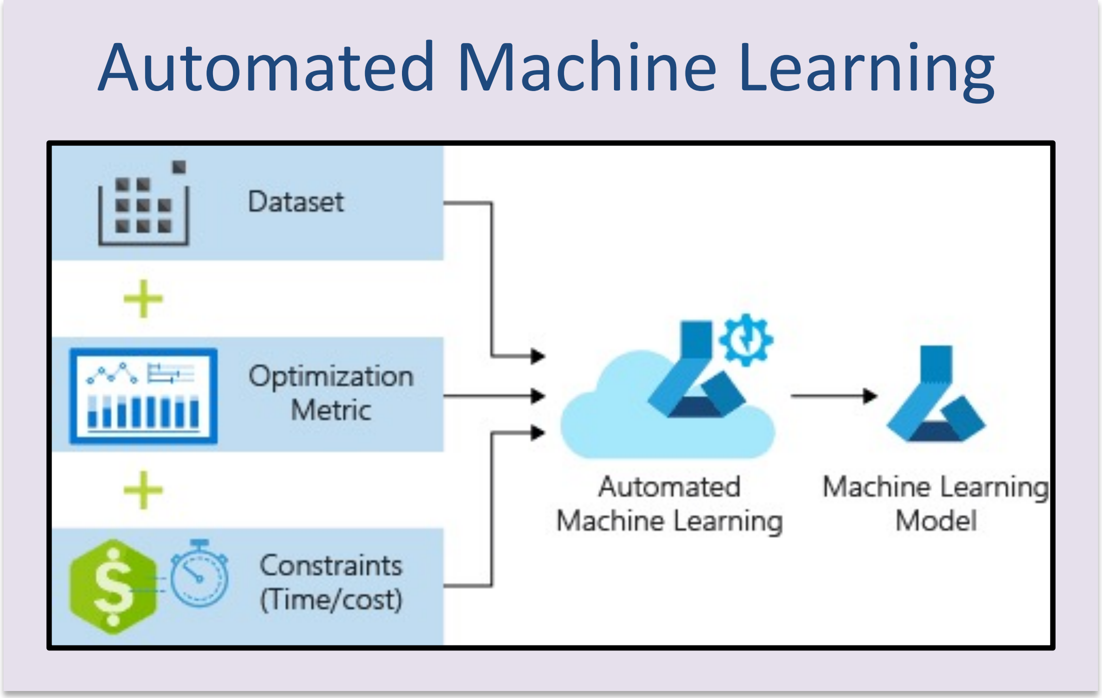
- Advantages
  - Feature representations of users and items
  - Non-linear relationships between users and items
- Typical architecture
  - Embedding layer
  - Interaction layer
  - Inference layer
- Manually designed architecture
  - Expert knowledge
  - Time and engineering efforts
  - Human error and bias → suboptimal architecture



AutoML for Deep Recommender Systems

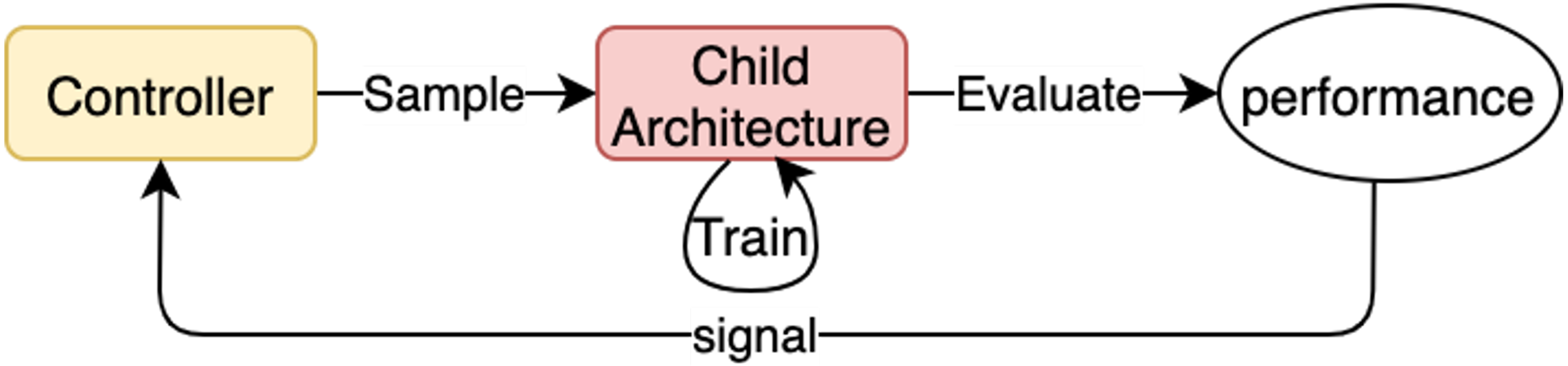
- Deep architectures are designed by the machine automatically

- Advantages
  - Less expert knowledge
  - Saving time and efforts
  - Different data → different architectures



Neural Architecture Search

- Reinforcement Learning-based NAS
  - Controller: learning to select optimal child architecture
  - Child architecture: the DNN with a specific architecture

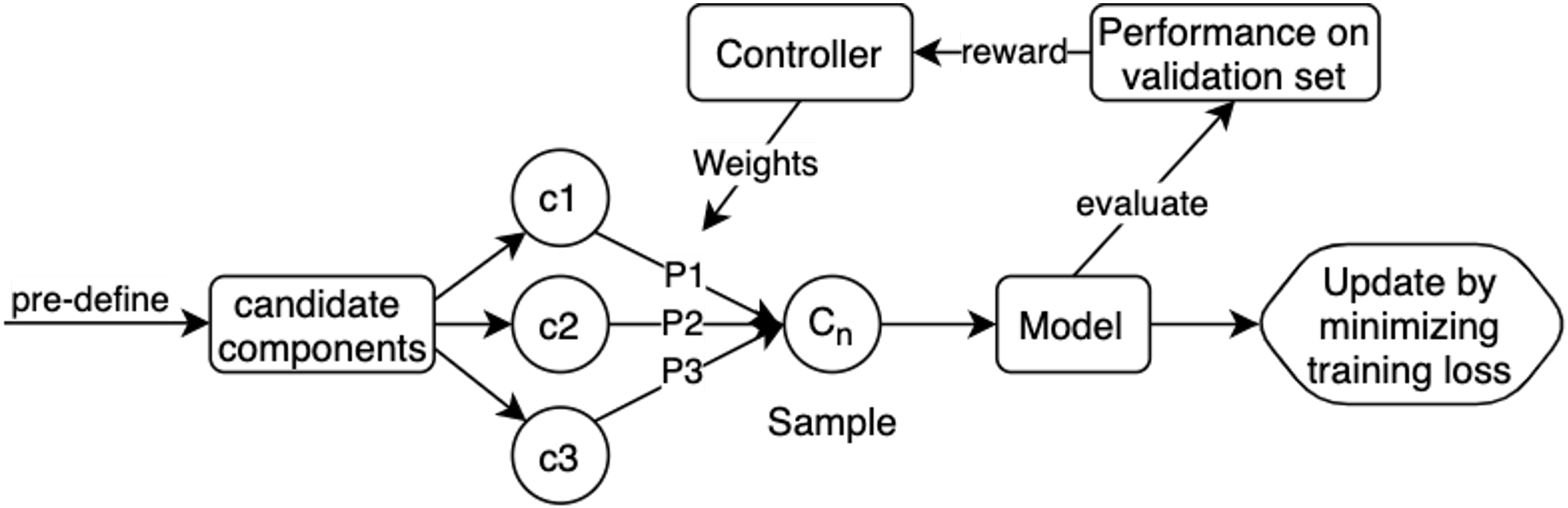


candidate components:

- identity
- 1x7 then 7x1 convolution
- 3x3 average pooling
- 5x5 max pooling
- 1x1 convolution
- 3x3 depthwise-separable conv
- 7x7 depthwise-separable conv
- 1x3 then 3x1 convolution
- 3x3 dilated convolution
- 3x3 max pooling
- 7x7 max pooling
- 3x3 convolution
- 5x5 depthwise-seperable conv

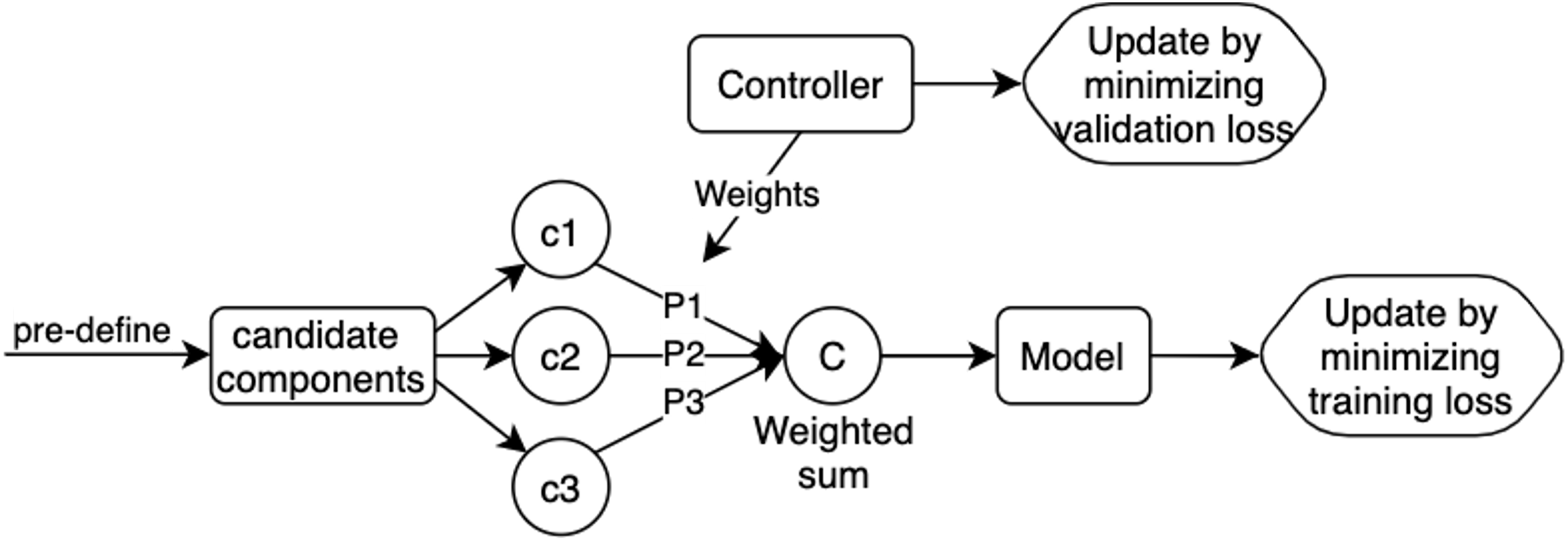
Neural Architecture Search

- Reinforcement Learning-based NAS
  - Hard selection on candidate components
  - The model's performance on validation set are viewed as reward
  - The weights of controller are updated to maximize the reward

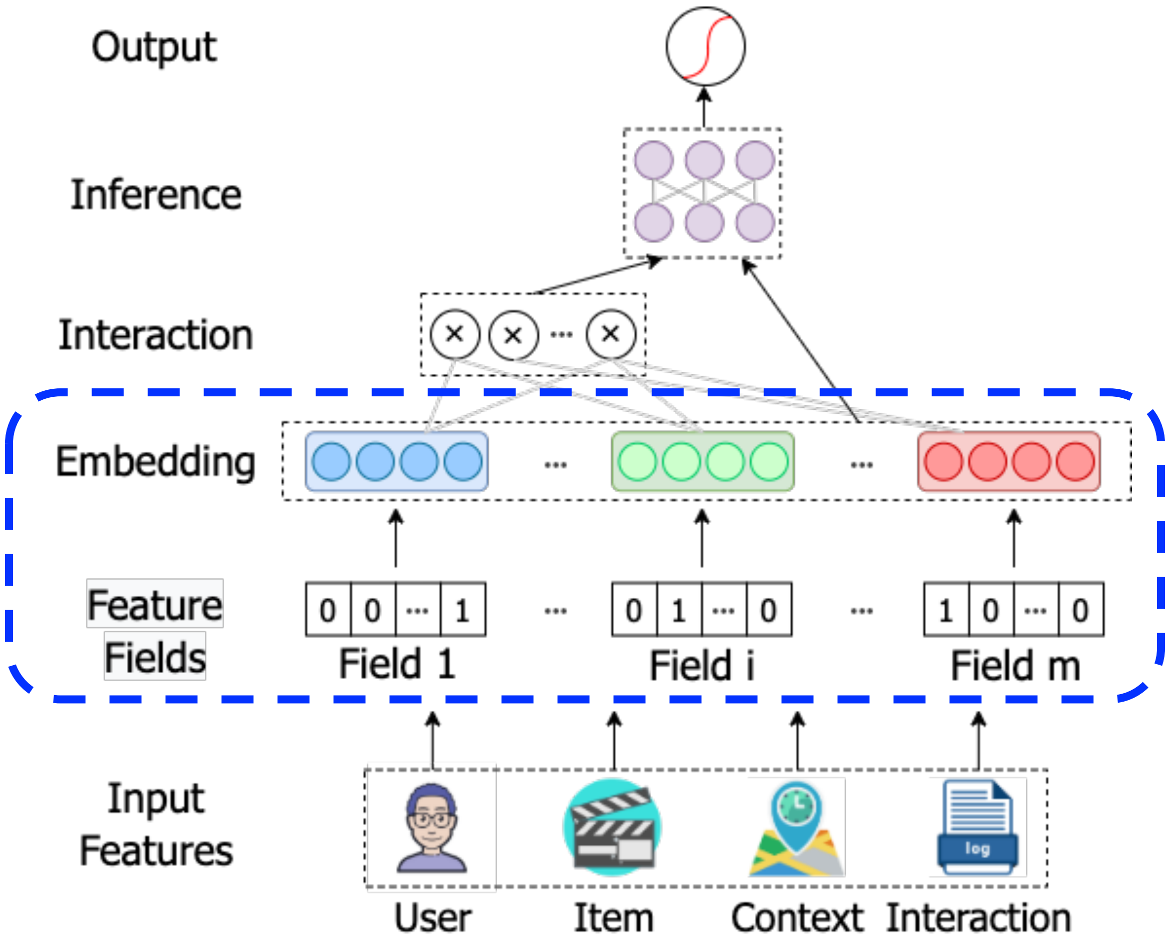


Neural Architecture Search

- Gradient Descent-based NAS
  - Soft selection on candidate components, weighted sum them
  - Directly update the controller weights by minimizing the loss on validation set

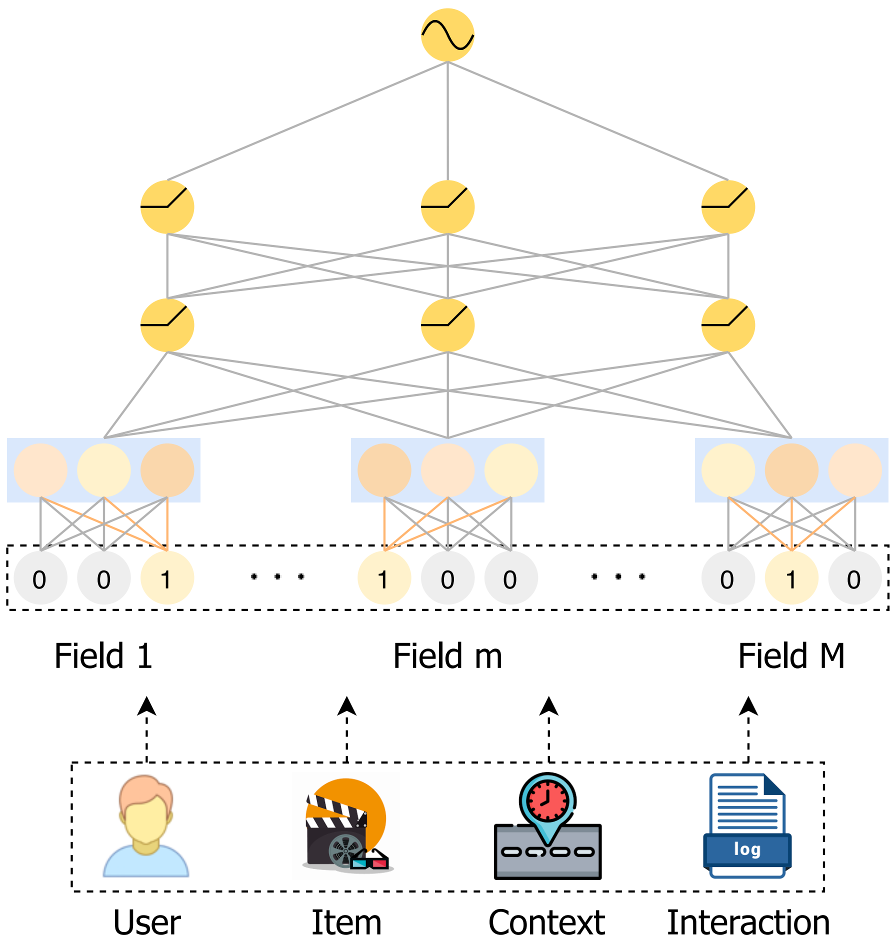


AutoML in Embedding Layer



Embedding Components

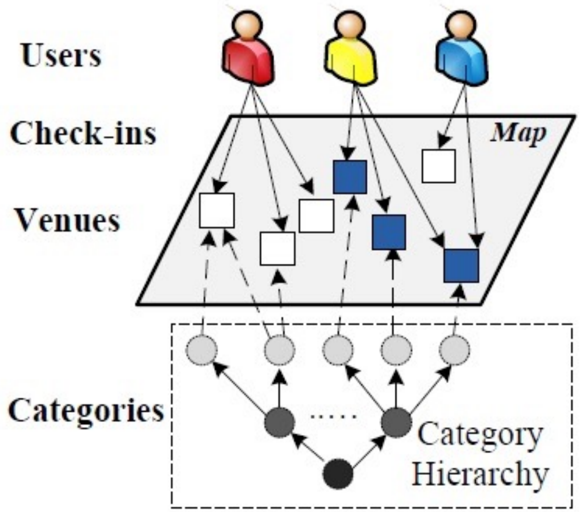
- Real-world recommender systems involve numerous feature fields
  - Users
    - e.g., gender and age
  - Items
    - e.g., category and price
  - Contextual information
    - e.g., time and location
  - Their interactions
    - e.g., ***users***' purchased ***items*** at ***location*** *A*



- Features → Embeddings
  - Unified dimension for all features

Unified Embedding Dimension

- Memory inefficiency problem
  - Embedding dimension → Capacity to encode information
  - Different feature fields have different cardinality
  - Different features have different frequency



Target	Weekday	Gender	User_ID
1	Tuesday	Male	0000001
0	Monday	Female	3495682
1	Thursday	Female	5676562
0	Friday	Male	9231237
	7	2	million

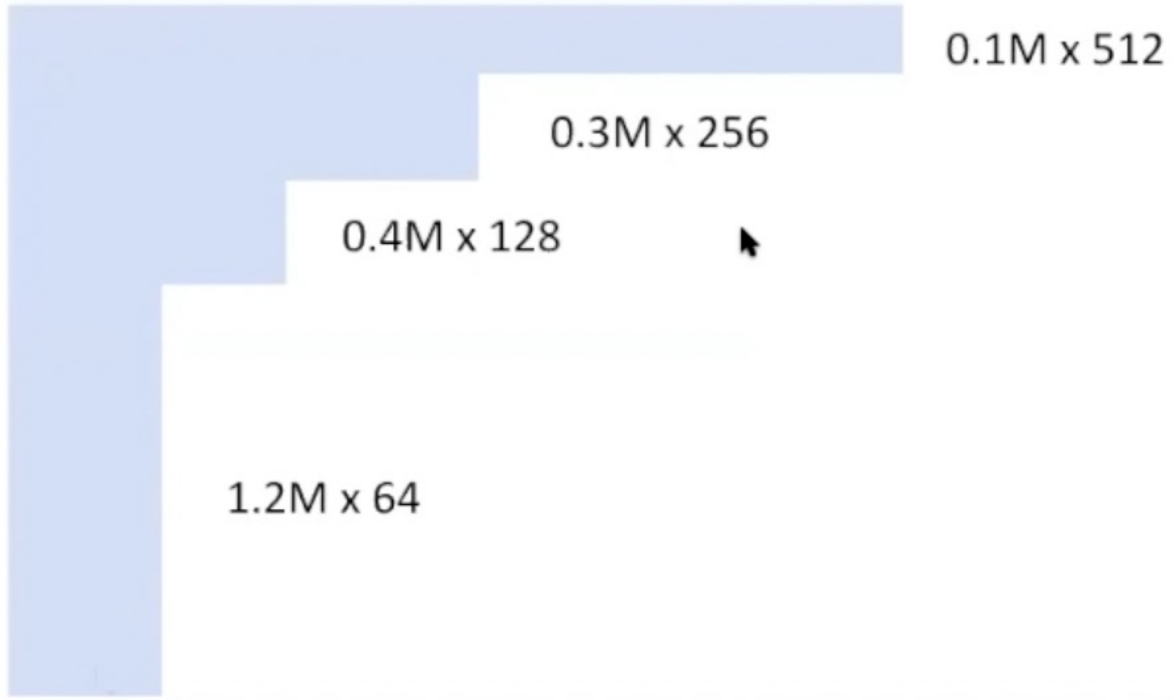
Outline

- **AutoML in Embedding Layer**
  - **NIS** - Neural Input Search for Large Scale Recommendation Models (KDD’2020)
  - **ESAPN** - Automated Embedding Size Search in Deep Recommender Systems (SIGIR’2020)
  - **AutoDim** - Field-aware Embedding Dimension Search in Recommender Systems (WWW’2021)
  - **AutoDis** - Automatic Discretization for Embedding Numerical Features in CTR Prediction (AAAI’2021)

- **AutoML in Interaction Layer**
  - **AutoFIS** - Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction (KDD’2020)
  - **AutoGroup** - Automatic Feature Grouping for Modelling Explicit High-Order Feature Interactions in CTR Prediction (SIGIR’2020)
  - **AutoCTR** - Towards Automated Neural Interaction Discovery for Click-Through Rate Prediction (KDD’2020)

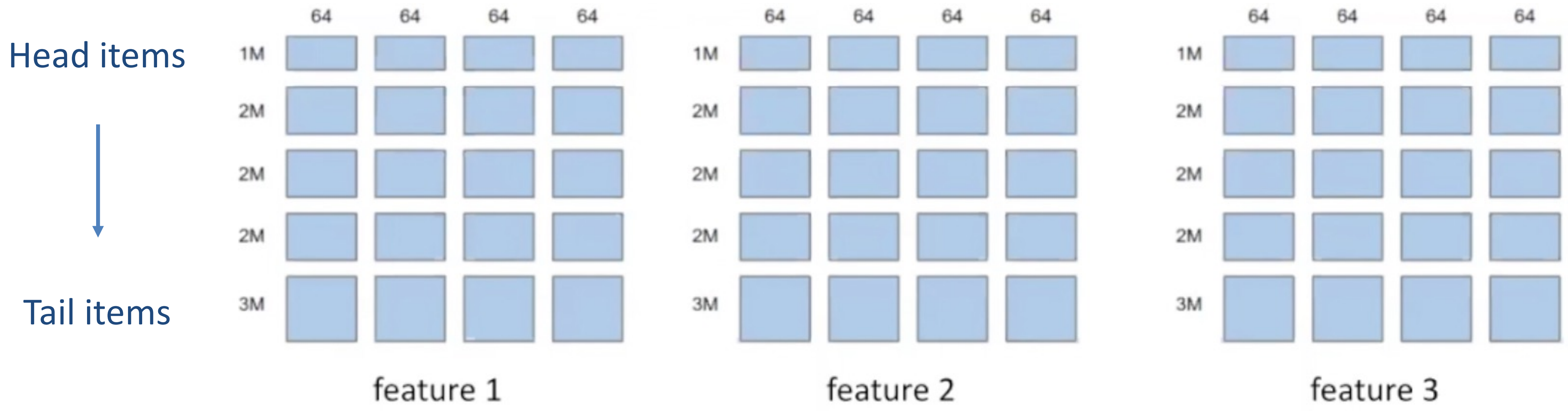
NIS - Motivation

- Head items
  - More data, more information
  - Needing larger embedding size
- Tail items
  - Less data, less information
  - Small embedding size is enough



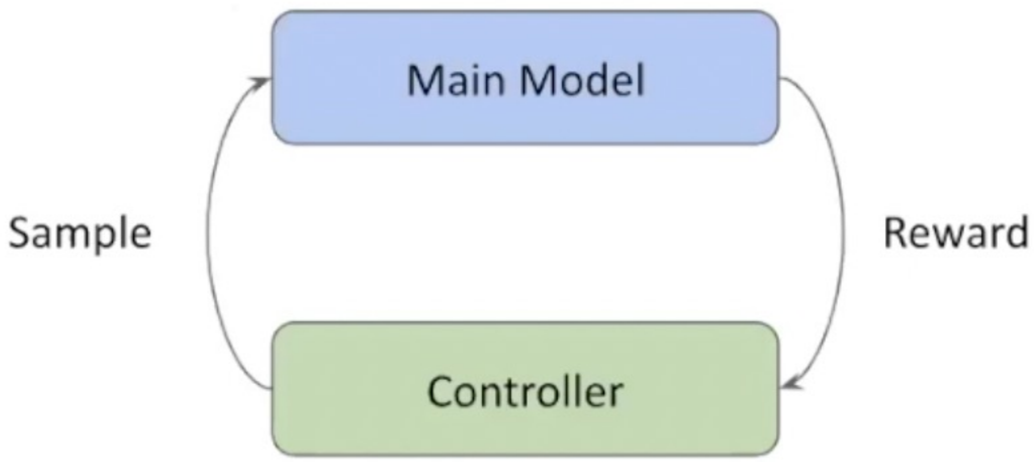
NIS - Search Space

- Assume 3 features, each with largest allowed embedding matrix of size 10M x 256
  - Items should be sorted by their frequency
  - Cutting the embedding matrix into smaller pieces
  - The way to cut the embedding matrix is pre-defined



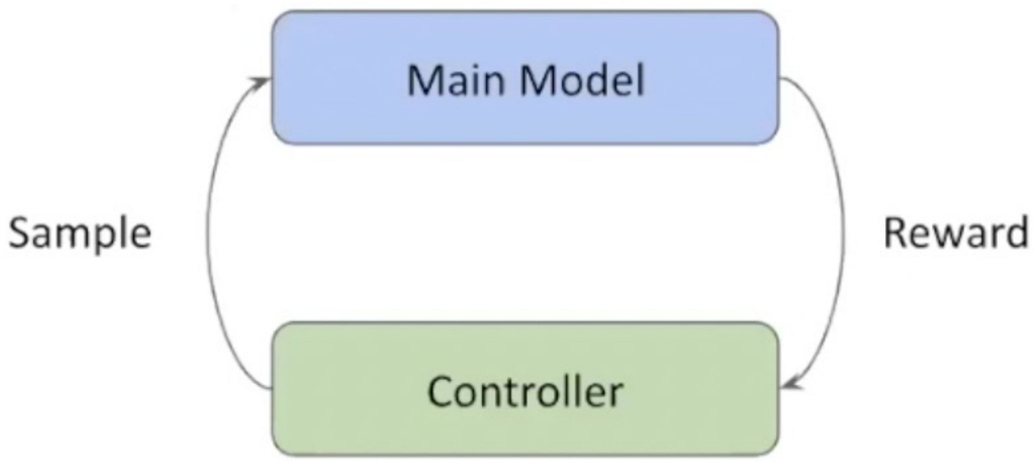
NIS - Multisize Embedding

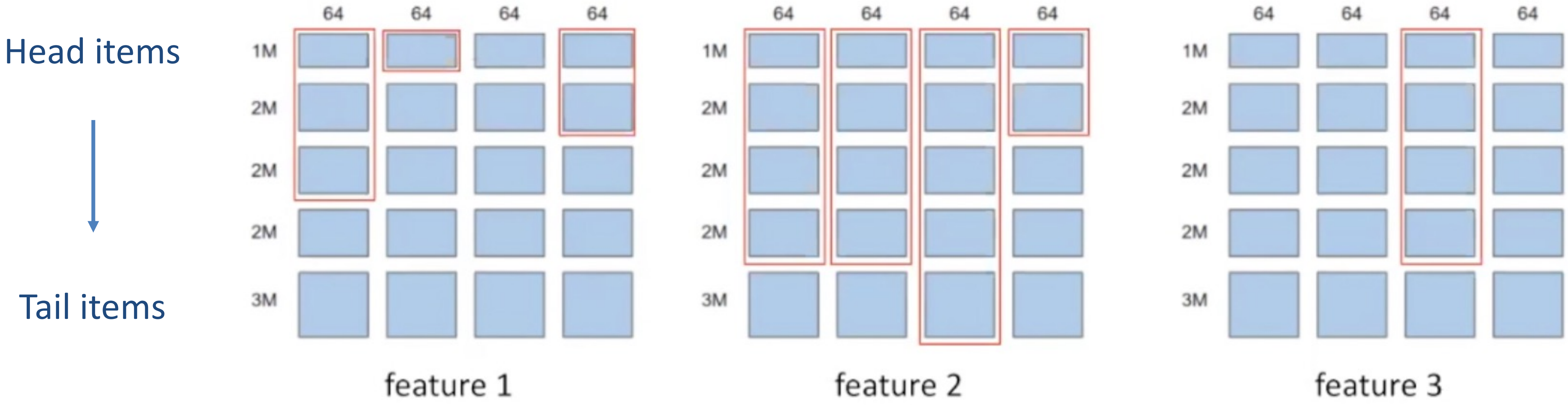
- RL-based AutoML approach
  - Main model is the deep recommendation model
  - Controller learns to sample embedding dimensions that generate higher reward



NIS - Multisize Embedding

- RL-based AutoML approach
  - Main model is the deep recommendation model
  - Controller learns to sample embedding dimensions that generate higher reward
  - E.g. feature 1: 1M x 192 + 2M x 128 + 2M x 64
  - Reward: $R = R_Q - \lambda * C_M$





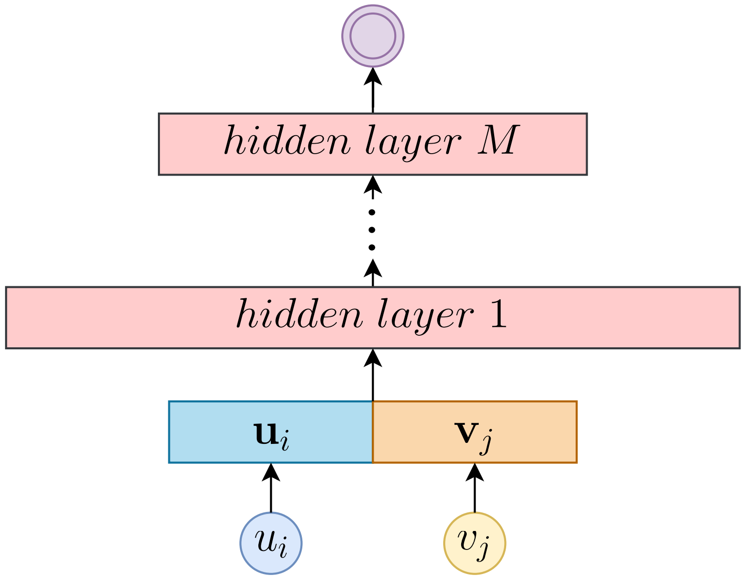
Outline

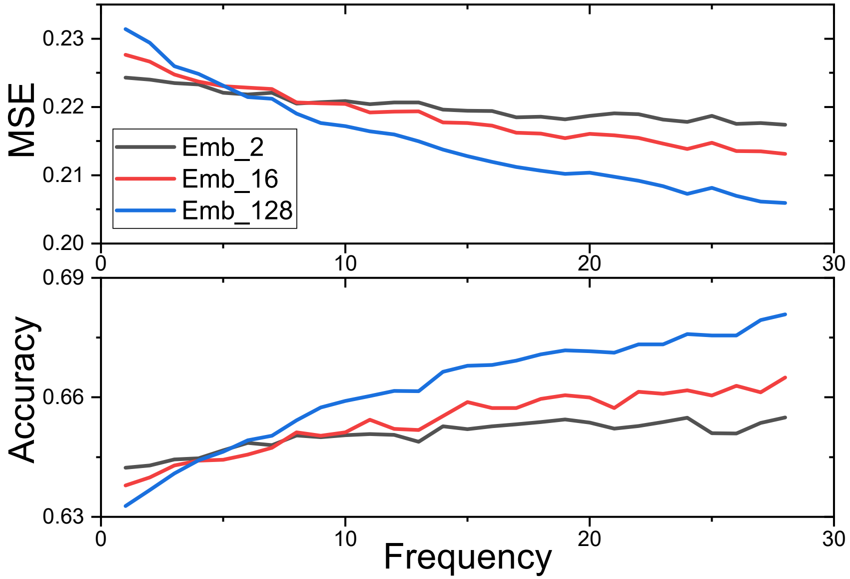
- **AutoML in Embedding Layer**
  - **NIS** - Neural Input Search for Large Scale Recommendation Models (KDD’2020)
  - **ESAPN** - Automated Embedding Size Search in Deep Recommender Systems (SIGIR’2020)
  - **AutoDim** - Field-aware Embedding Dimension Search in Recommender Systems (WWW’2021)
  - **AutoDis** - Automatic Discretization for Embedding Numerical Features in CTR Prediction (AAAI’2021)

- **AutoML in Interaction Layer**
  - **AutoFIS** - Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction (KDD’2020)
  - **AutoGroup** - Automatic Feature Grouping for Modelling Explicit High-Order Feature Interactions in CTR Prediction (SIGIR’2020)
  - **AutoCTR** - Towards Automated Neural Interaction Discovery for Click-Through Rate Prediction (KDD’2020)

Motivations

- Preliminary Experiment
  - Frequency: # interactions a user/item

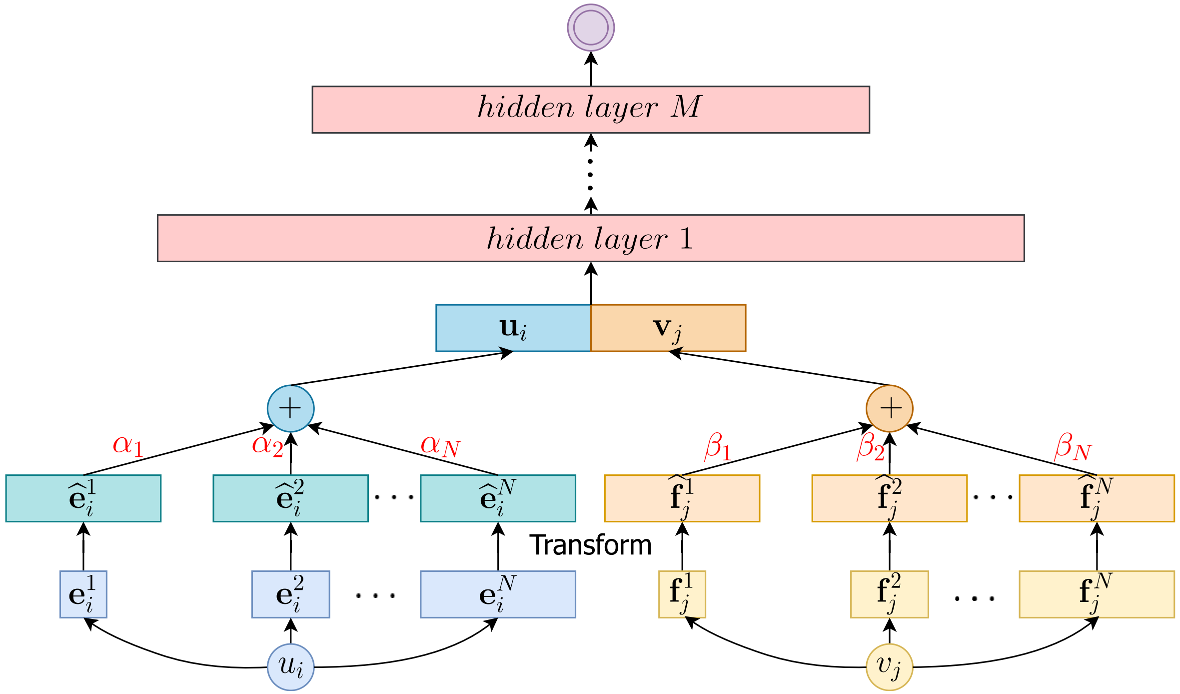




- Embedding dimension often determines the capacity to encode information

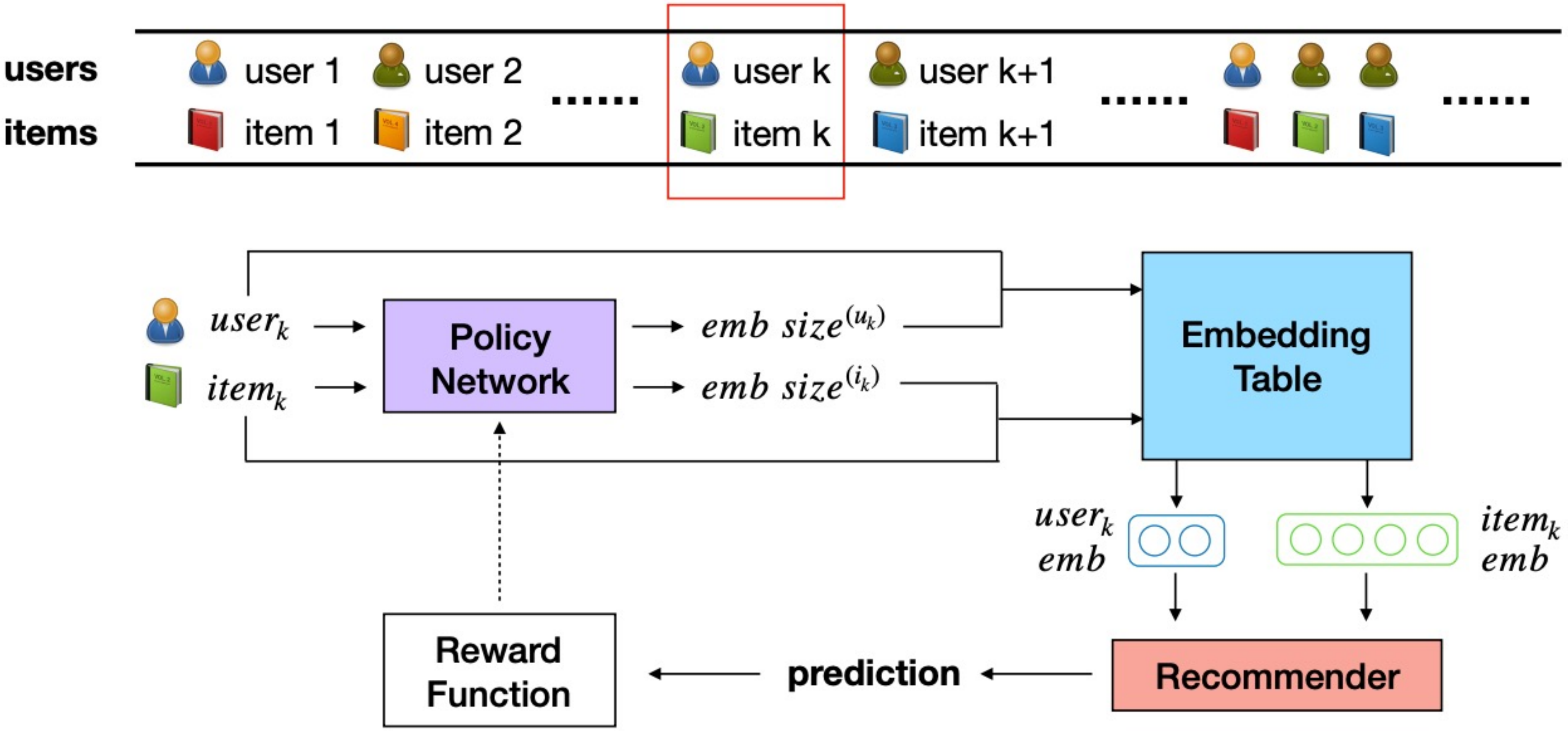
Motivations

- Dynamically search the embedding sizes for different users and items
  - Optimal recommendation quality all the time
  - More efficient in memory

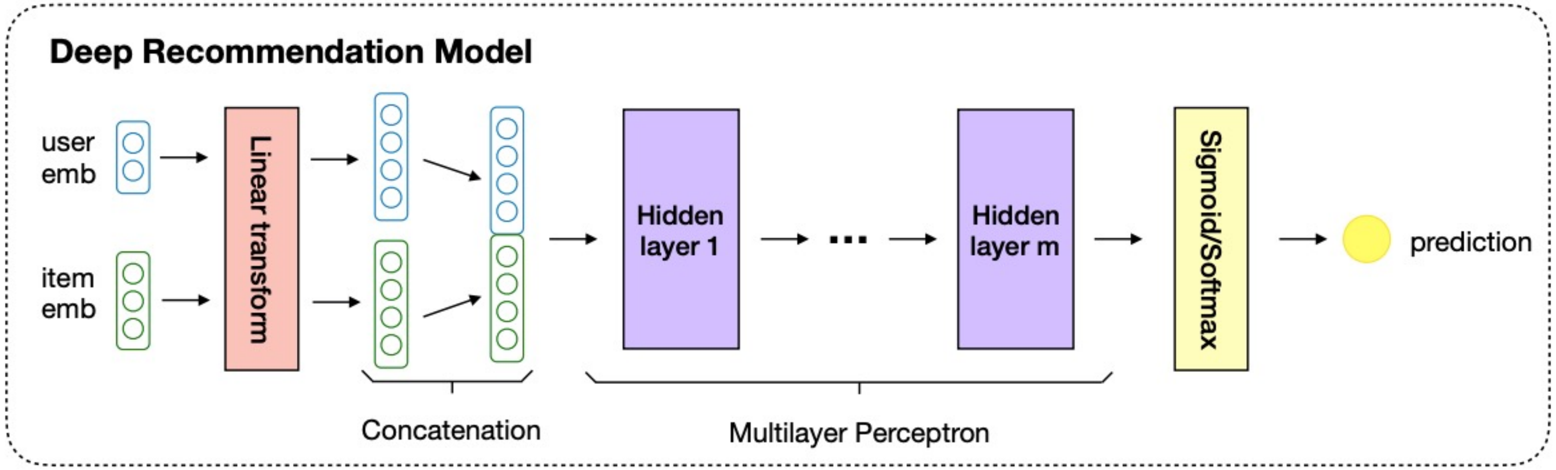


Overview

- Two Components
  - Deep recommendation model
  - Embedding Size Adjustment Policy Network (ESAPN): hard selection via RL



Deep Recommendation Model



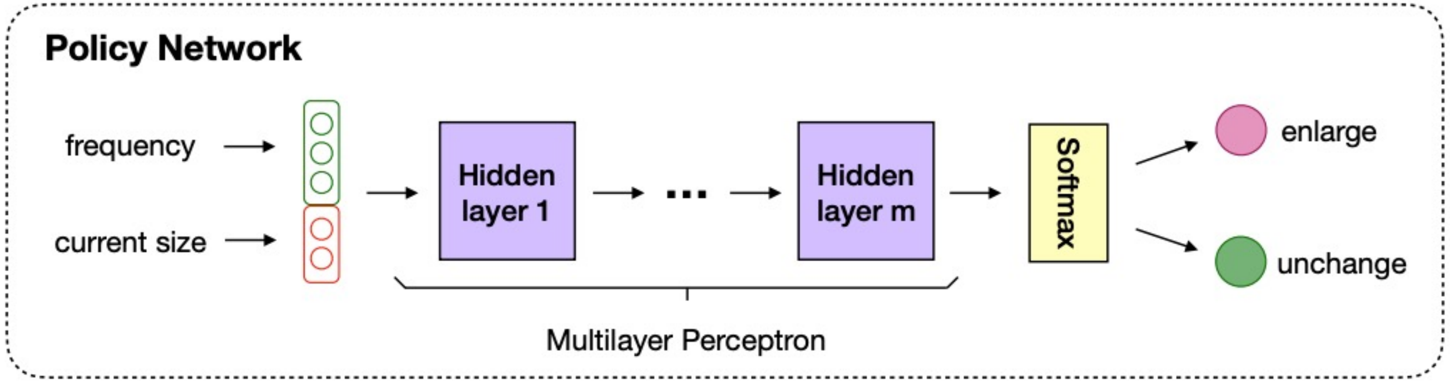
- Candidate embedding sizes

$$D = \{d_1, d_2, \ldots, d_n\} \qquad d_1 < d_2 < \cdots < d_n$$

- Linear transformations

$$\begin{aligned} \mathbf{e_2} &= W_{1\to2}\mathbf{e_1} + b_{1\to2} \\ \mathbf{e_3} &= W_{2\to3}\mathbf{e_2} + b_{2\to3} \\ \cdots \\ \mathbf{e_n} &= W_{n-1\to n}\mathbf{e_{n-1}} + b_{n-1\to n} \end{aligned}$$

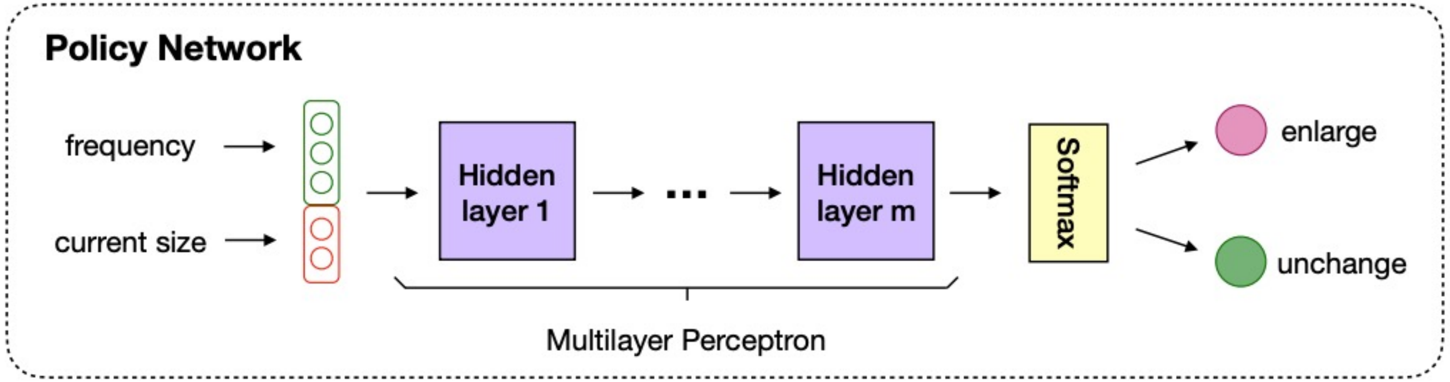
Policy Network



- Environment
  - The deep recommendation model

- State
  - *s = (f, e)*
  - *f*: frequency   *e:* current embedding size

Policy Network



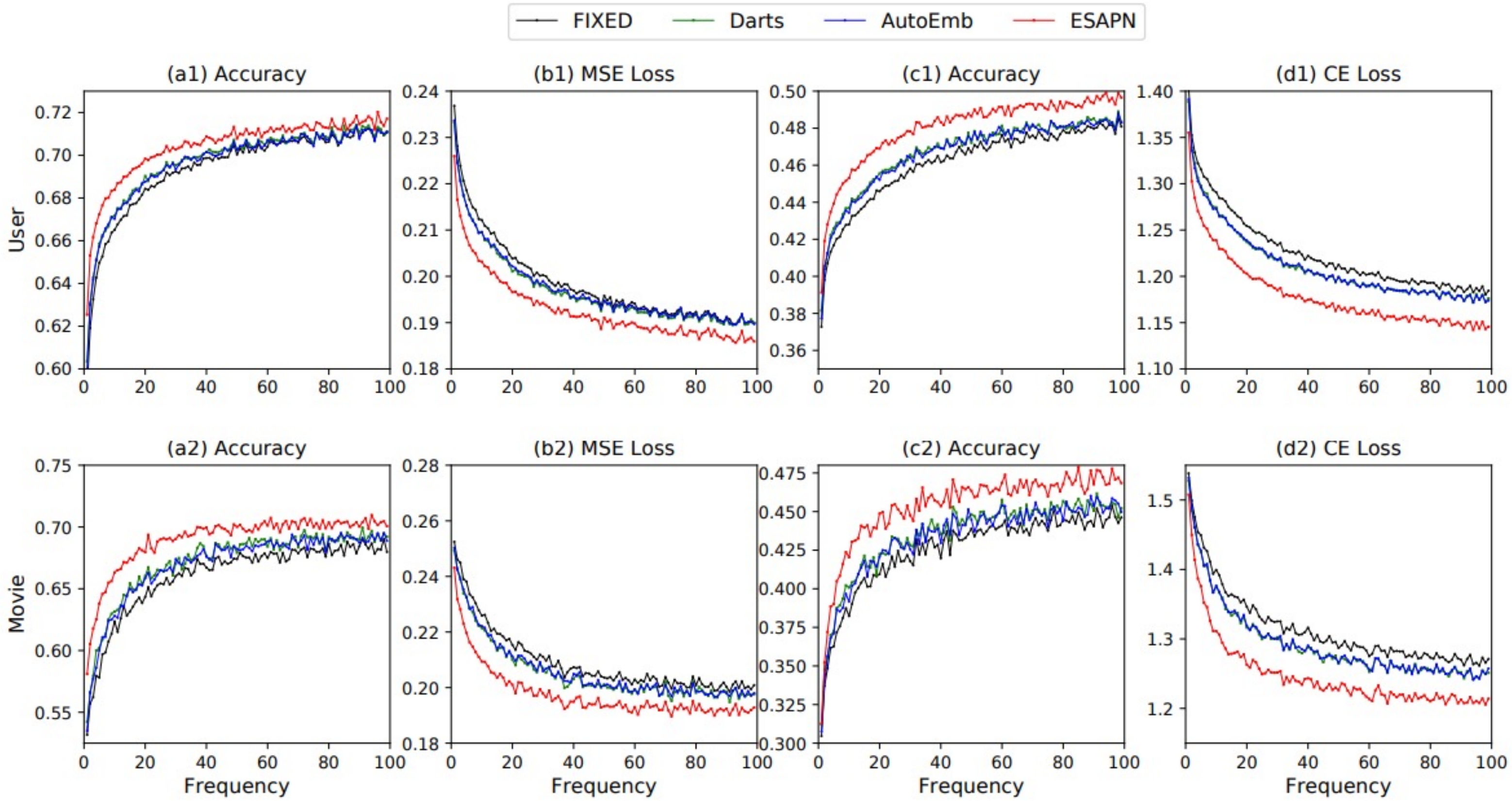
- Action
  - Enlarge or Unchange

- Reward

$$L^{(u)} = (L_1^{(u)}, \ldots, L_T^{(u)}) \qquad L^{(i)} = (L_1^{(i)}, \ldots, L_T^{(i)})$$

$$R^{(u)} = \frac{1}{T}\sum_{t=1}^{T} L_t^{(u)} - L \qquad R^{(i)} = \frac{1}{T}\sum_{t=1}^{T} L_t^{(i)} - L$$

Performance with Frequency



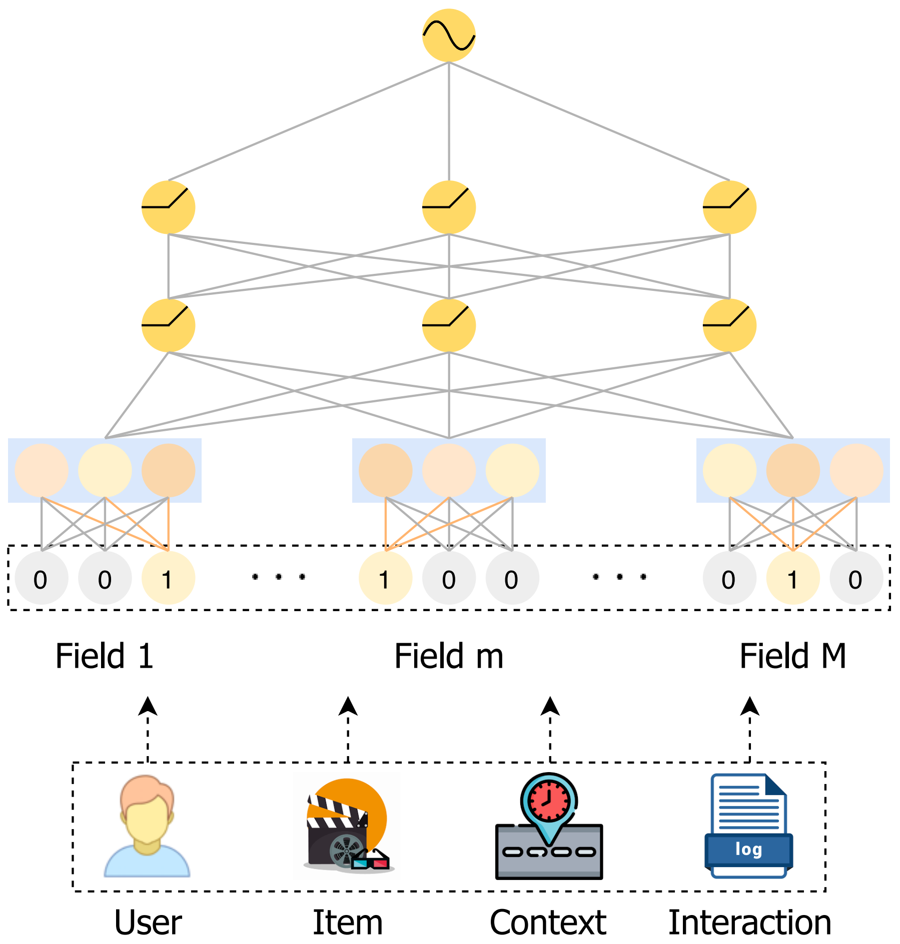
Outline

- **AutoML in Embedding Layer**
  - **NIS** - Neural Input Search for Large Scale Recommendation Models (KDD'2020)
  - **ESAPN** - Automated Embedding Size Search in Deep Recommender Systems (SIGIR'2020)
  - **AutoDim** - Field-aware Embedding Dimension Search in Recommender Systems (WWW'2021)
  - **AutoDis** - Automatic Discretization for Embedding Numerical Features in CTR Prediction (AAAI'2021)

- **AutoML in Interaction Layer**
  - **AutoFIS** - Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction (KDD'2020)
  - **AutoGroup** - Automatic Feature Grouping for Modelling Explicit High-Order Feature Interactions in CTR Prediction (SIGIR'2020)
  - **AutoCTR** - Towards Automated Neural Interaction Discovery for Click-Through Rate Prediction (KDD'2020)

AutoDim - Motivation

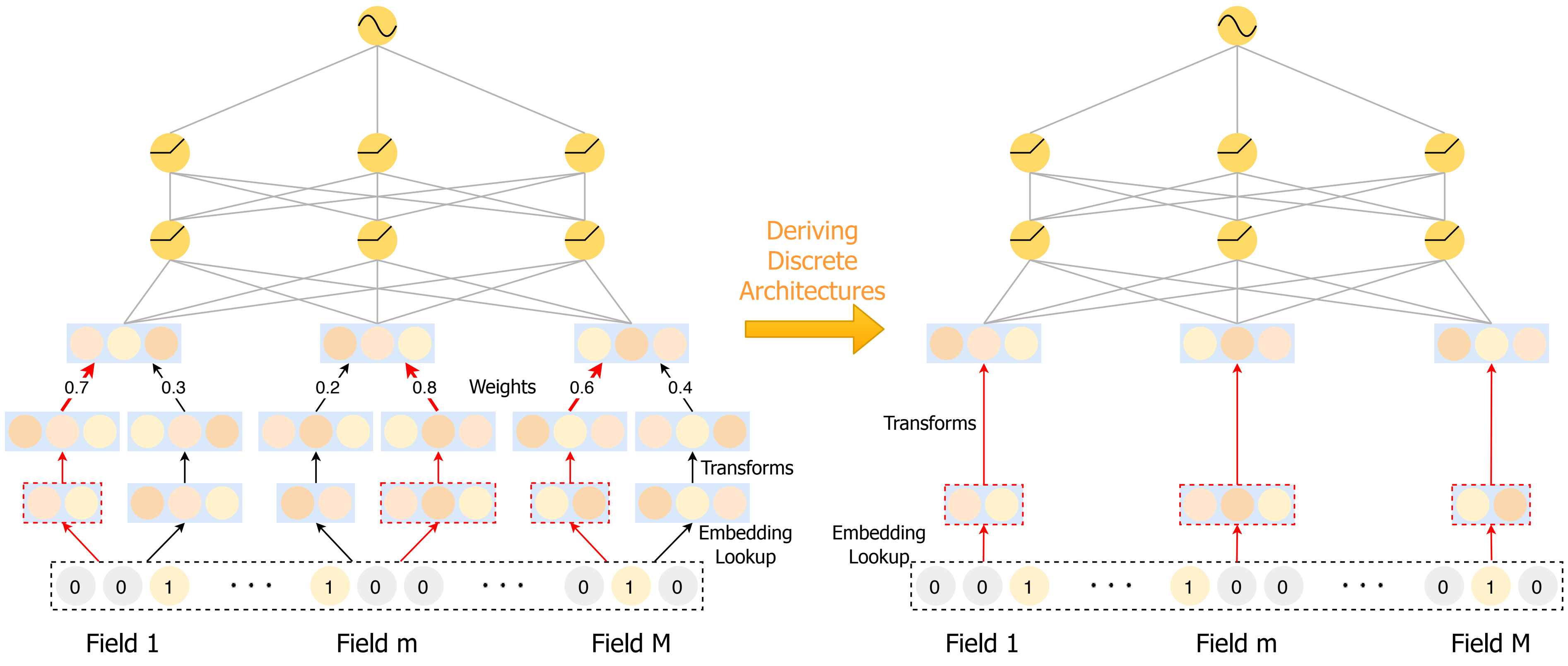
- Complex relationship
  - Embedding dimensions
  - Feature distributions
  - Neural network architectures
- Large search space
  - M feature field (M > 100)
  - K candidate dimensions
  - K^M selecion space



- Goal: Selecting embedding dimensions to different feature fields automatically

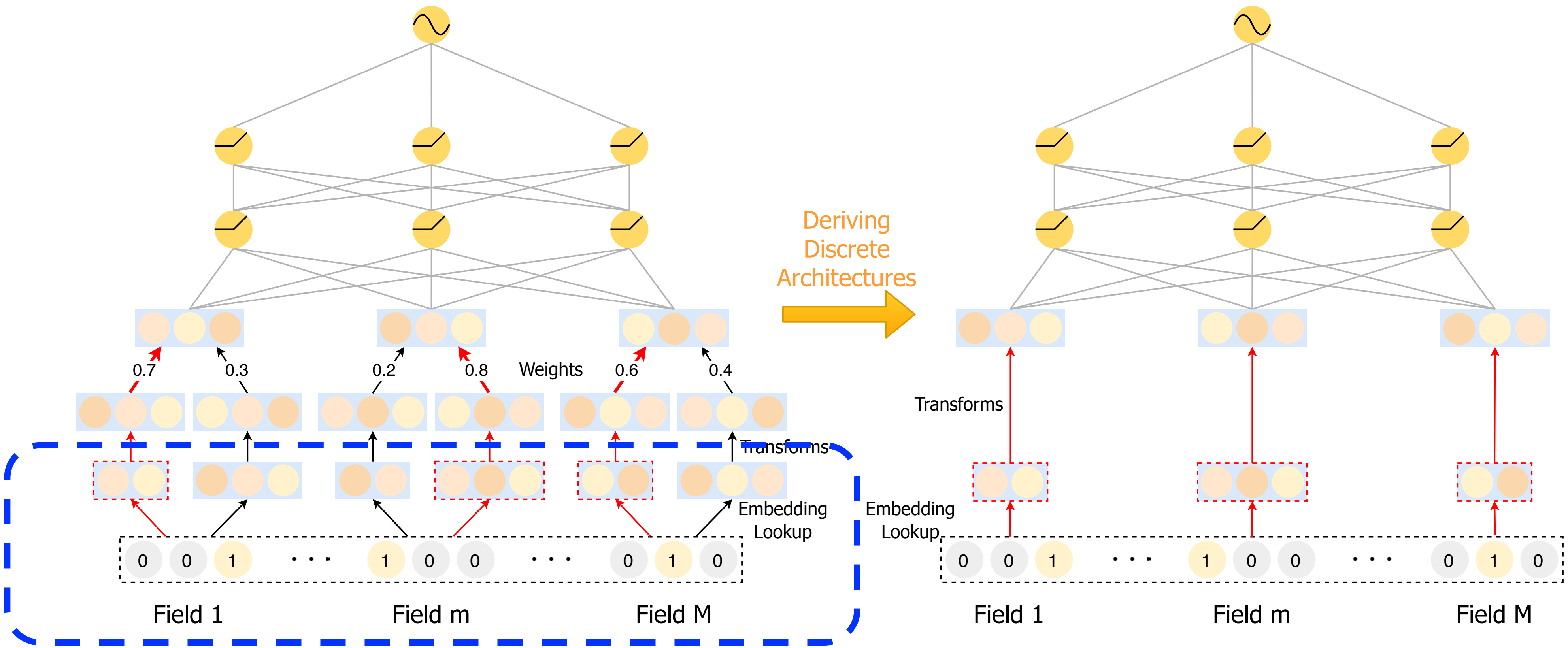
AutoDim - Overview

- Two-stage framework



(a) Dimension Search　　(b) Parameter Retraining

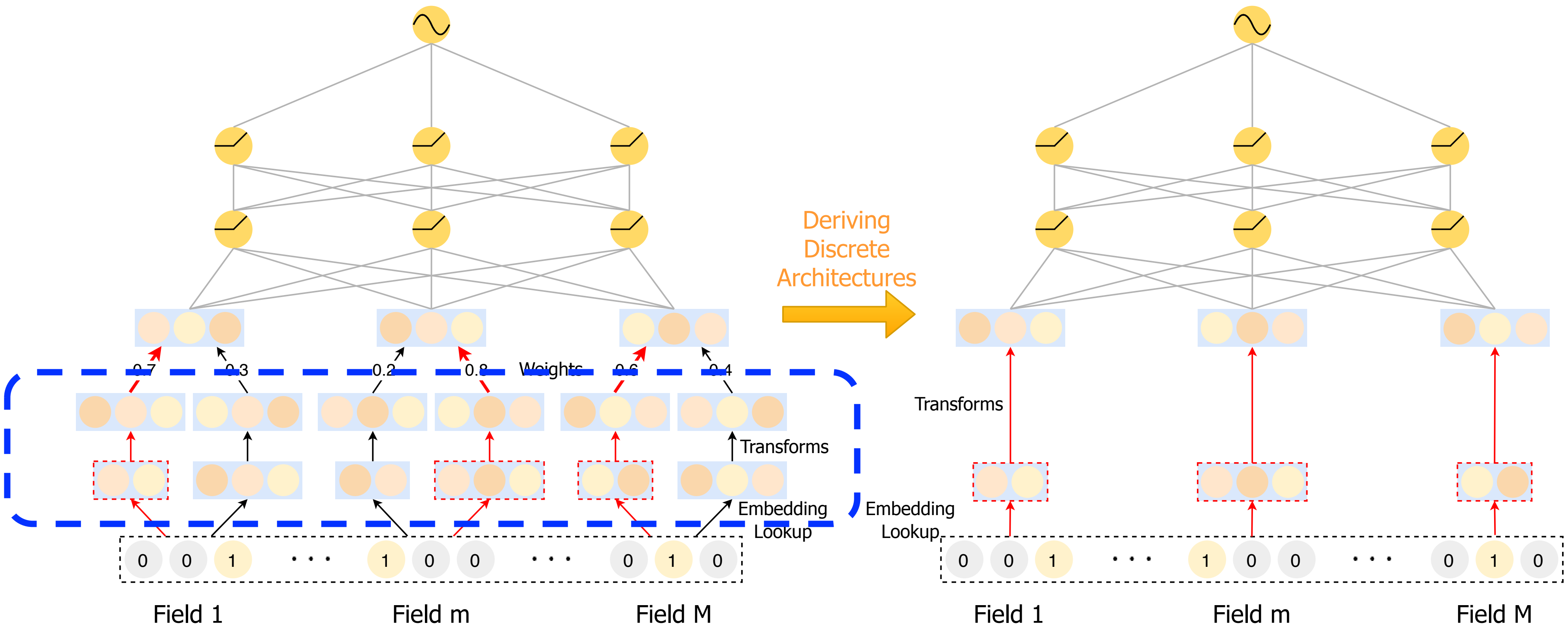
AutoDim - Dimension Search Stage



(a) Dimension Search

(b) Parameter Retraining

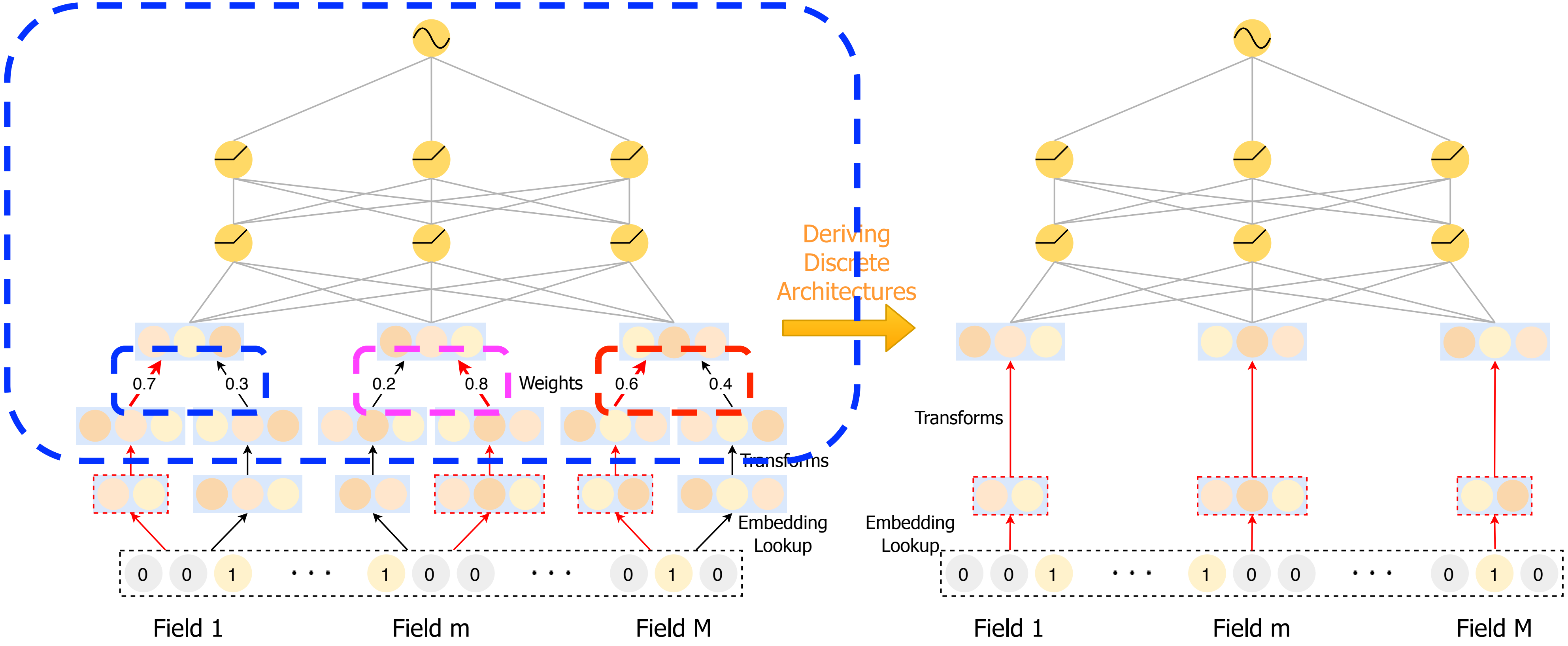
AutoDim - Dimension Search Stage



(a) Dimension Search

(b) Parameter Retraining

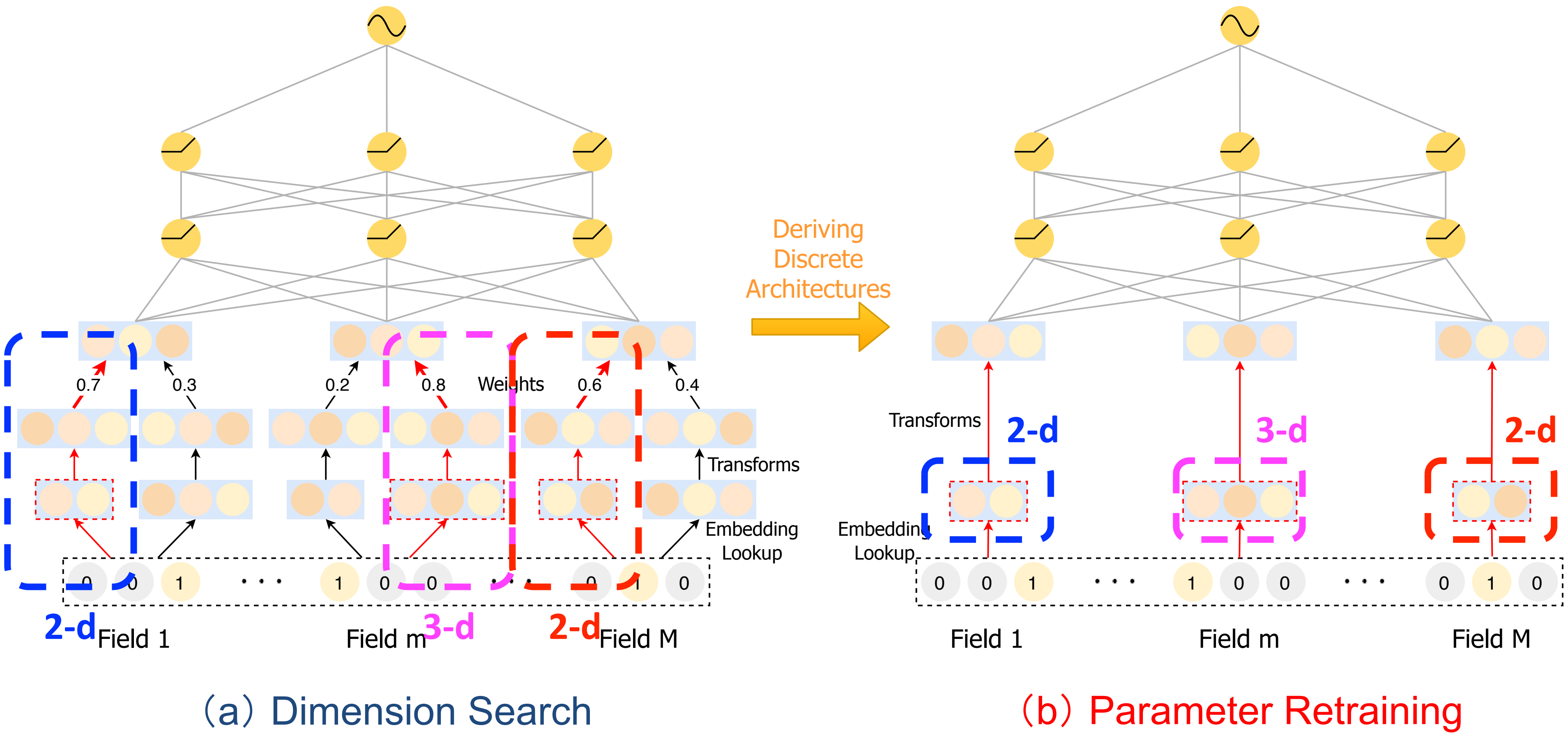
AutoDim - Dimension Search Stage



（a）Dimension Search

（b）Parameter Retraining

AutoDim - Parameter Retraining Stage



Overall Performance

Dataset	Model	Metrics	Search Methods								
			FDE	MDE	DPQ	NIS	MGQE	AEmb	RaS	AD-s	**AutoDim**
Criteo	FM	AUC	0.8020	0.8027	0.8035	0.8042	0.8046	0.8049	0.8056	0.8063	**0.8078***
		Logloss	0.4487	0.4481	0.4472	0.4467	0.4462	0.4460	0.4457	0.4452	**0.4438***
		EP (M)	34.778	15.520	20.078	13.636	12.564	13.399	16.236	31.039	**11.632***
Criteo	W&D	AUC	0.8045	0.8051	0.8058	0.8067	0.8070	0.8072	0.8076	0.8081	**0.8098***
		Logloss	0.4468	0.4464	0.4457	0.4452	0.4446	0.4445	0.4443	0.4439	**0.4419***
		EP (M)	34.778	18.562	22.628	14.728	15.741	15.987	18.233	30.330	**12.455***
Criteo	DeepFM	AUC	0.8056	0.8060	0.8067	0.8076	0.8080	0.8082	0.8085	0.8089	**0.8101***
		Logloss	0.4457	0.4456	0.4449	0.4442	0.4439	0.4438	0.4436	0.4432	**0.4416***
		EP (M)	34.778	17.272	25.737	12.955	13.059	13.437	17.816	31.770	**11.457***

"*" indicates the statistically significant improvements (i.e., two-sided t-test with $p < 0.05$) over the best baseline. (M=Million)

- Metrics: AUC ↑, Logloss ↓, EP ↓ (embedding parameters)
- AutoDim is general for any deep recommender systems with embedding layer
- Small serach space: 5 candiate for each feature field
- AutoDim → Best AUC and Logloss, and saving 70~80% embedding parameters

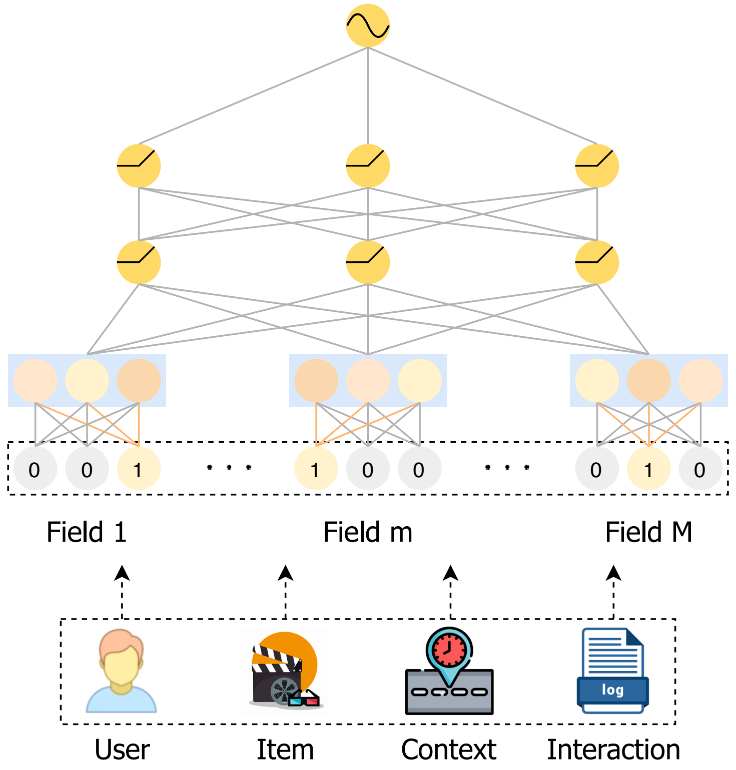
Outline

- **AutoML in Embedding Layer**
  - **NIS** - Neural Input Search for Large Scale Recommendation Models (KDD’2020)
  - **ESAPN** - Automated Embedding Size Search in Deep Recommender Systems (SIGIR’2020)
  - **AutoDim** - Field-aware Embedding Dimension Search in Recommender Systems (WWW’2021)
  - **AutoDis** - Automatic Discretization for Embedding Numerical Features in CTR Prediction (AAAI’2021)

- **AutoML in Interaction Layer**
  - **AutoFIS** - Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction (KDD’2020)
  - **AutoGroup** - Automatic Feature Grouping for Modelling Explicit High-Order Feature Interactions in CTR Prediction (SIGIR’2020)
  - **AutoCTR** - Towards Automated Neural Interaction Discovery for Click-Through Rate Prediction (KDD’2020)

AutoDis

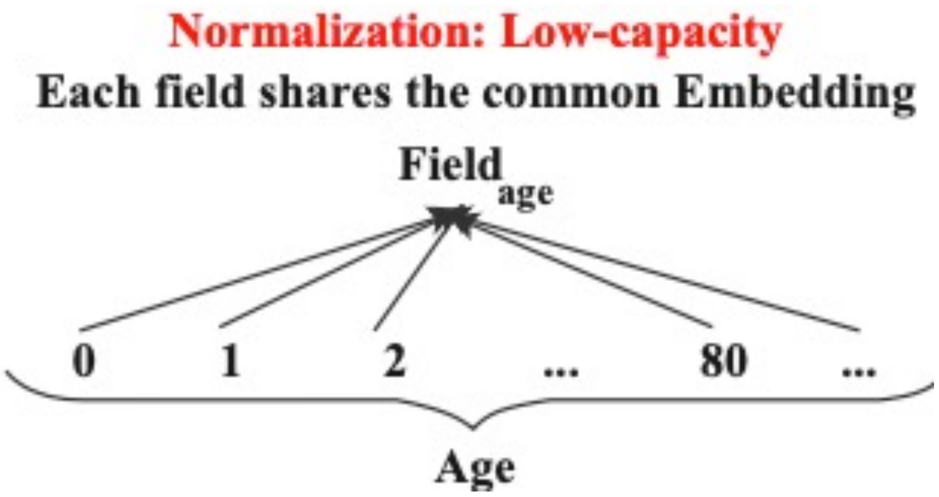
- Real-world recommender systems involve numerous feature fields
  - Users
    - e.g., gender and age
  - Items
    - e.g., category and price
  - Contextual information
    - e.g., time and location
  - Their interactions
    - e.g., ***users***' purchased ***items*** at ***location*** *A*



- E.g., Gender=Male, Day=Tuesday, Height=175.6, Age=18
  - Categorical field Gender v.s. Numerical field Height

Existing Methods for Numerical features

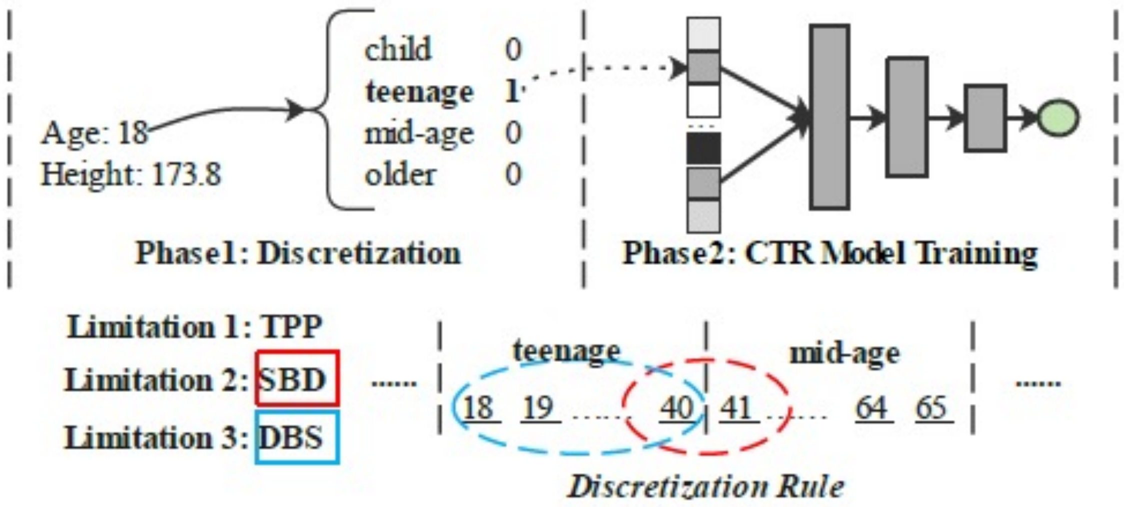
- Normalization
  - All the numerical features in the same field share a single embedding and scalar multiply with their values



- Disadvantage
  - Assuming embeddings of different features in the same field are linearly related to each other

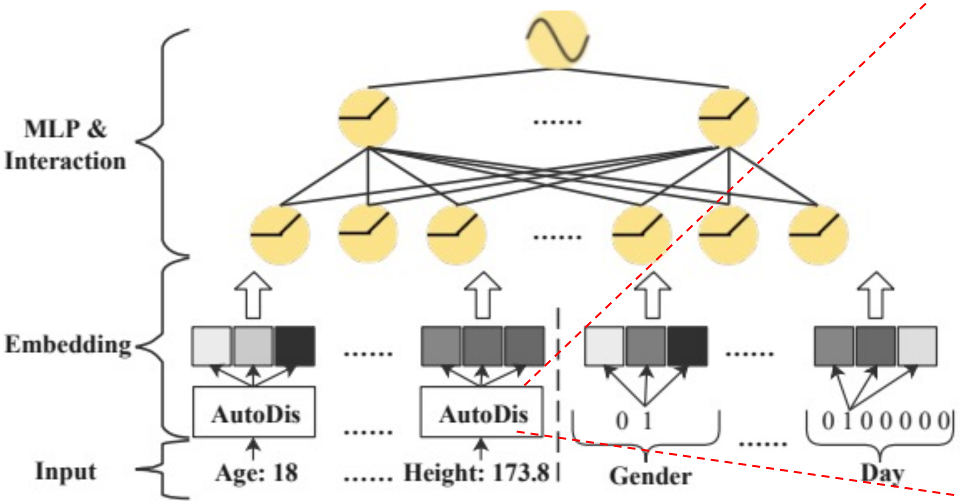
Existing Methods for Numerical features

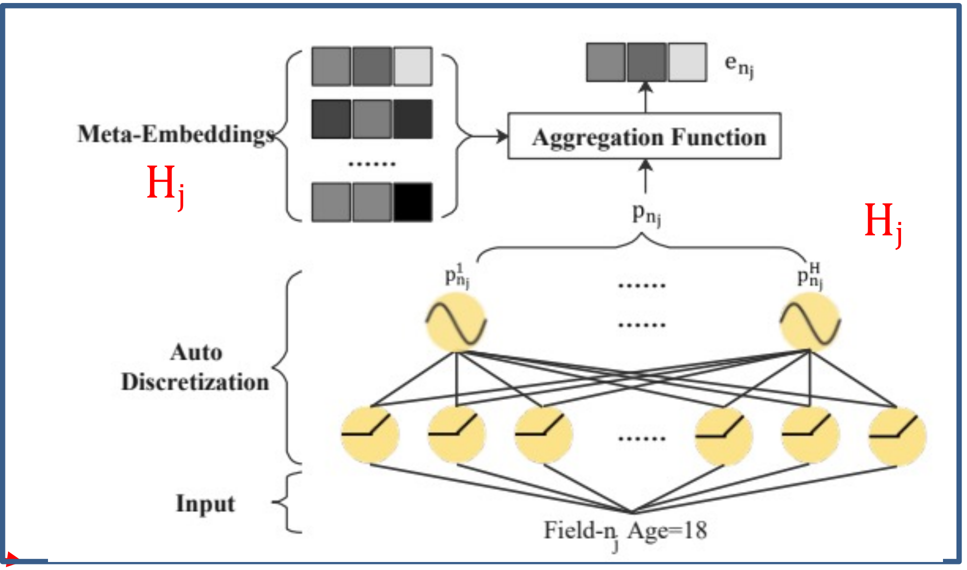
- Discretization
  - E.g., partitioning the range of the feature values into k buckets



- Disadvantages
  - TPP: cannot be optimized together with main model
  - SBD: different embeddings for similar numerical value 40 and 41
  - DBS: same embeddings for very different numerical value 18 and 40

Aggregation Function





Automatic Discretization

Continuous value mapping：

$$\widehat{x}^h_{n_j} = \mathbf{w}^h_{n_j} \cdot x_{n_j}$$

Softmax：

$$p^h_{n_j} = \frac{e^{\frac{1}{\tau}\widehat{x}^h_{n_j}}}{\sum_{l=1}^{H_j} e^{\frac{1}{\tau}\widehat{x}^l_{n_j}}},$$

Soft discretization output:

$$g(x_{n_j}) = [p^1_{n_j}, ..., p^h_{n_j}, ..., p^{H_j}_{n_j}].$$

Aggregation Function

Max-Pooling: $\mathbf{e}_{n_j} = \mathbf{ME}^k_{n_j}$, where $k = \arg max_{h\in\{1,2,...,H_j\}} p^h_{n_j}$,

Top-K-Sum: $\mathbf{e}_{n_j} = \sum_{l=1}^{K} \mathbf{ME}^{k_l}_{n_j}$, where $k_l = \arg \underset{l}{top}\ _{h\in\{1,2,...,H\}} p^h_{n_j}$,

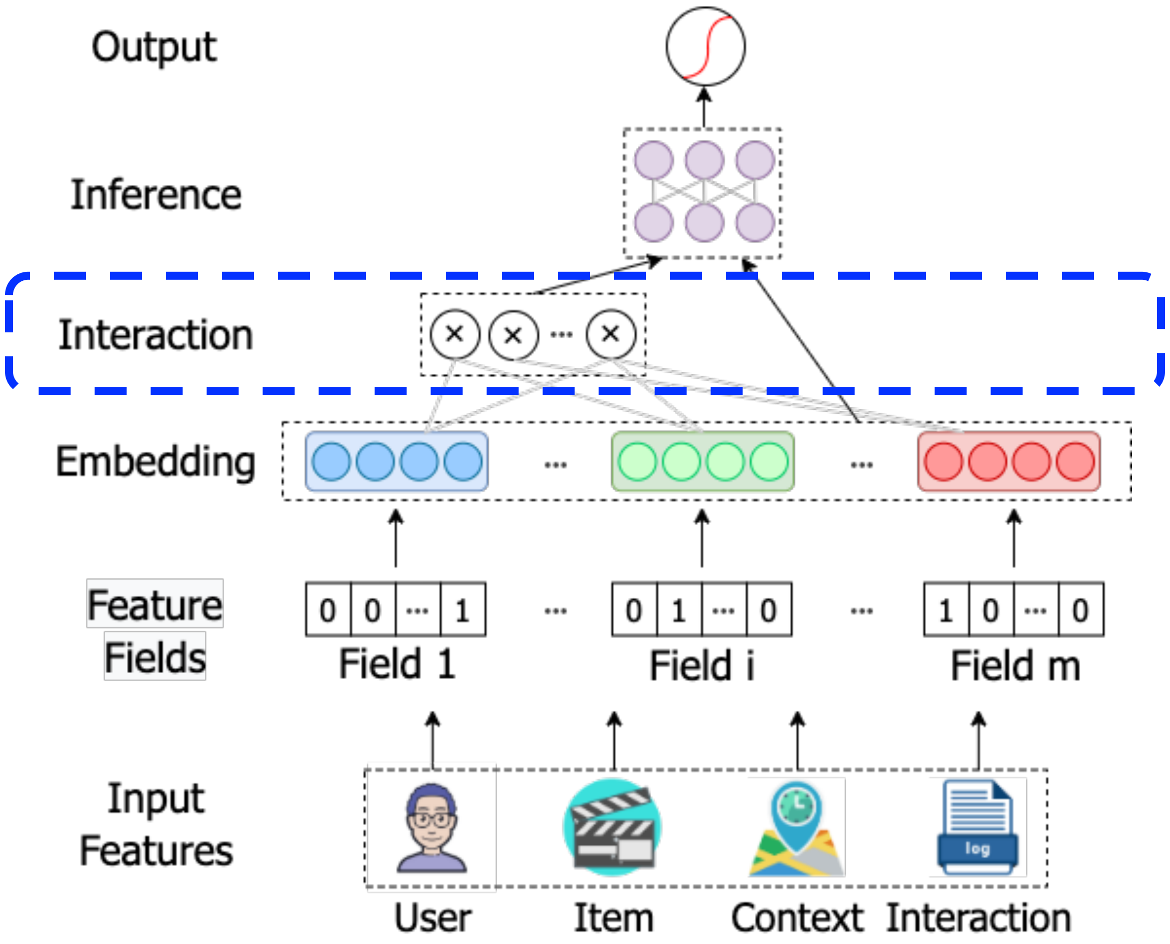
Weighted-Average: $\mathbf{e}_{n_j} = \sum_{l=1}^{H_j} p^l_{n_j} \cdot \mathbf{ME}^l_{n_j}.$

Advantages

- End-To-End:
  - The discretization of numerical features can be optimized jointly with the main model

- Continuous-But-Different
  - Different feature values are assigned with different embeddings
  - Closer the feature values have more similar the embeddings

AutoML in Interaction Layer



Background

- Multi-field data

Target	Weekday	Gender	City	Product Category
1	Tuesday	Male	London	Sports
0	Monday	Female	New York	Cosmetics
1	Thursday	Female	Beijing	Clothing
0	Friday	Male	Tokyo	Food

- High dimensional and sparse

Effectively Modelling Feature Interactions Is Important

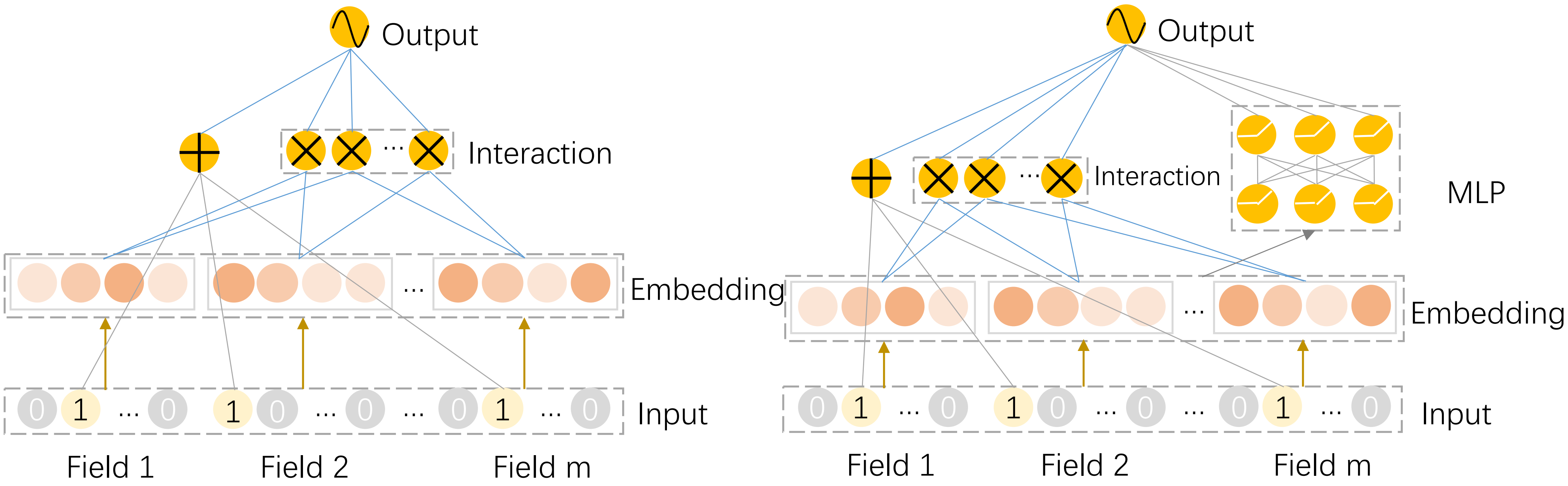
- User behavior is complicated to model

- Both low-order and high-order feature interactions play important roles to model user behavior.
  - People like to download popular apps → id of an app may be a signal
  - People often download apps for food delivery at meal time → interaction between app category and time-stamp may be a signal
  - Male teenagers like shooting game or RPG → interaction of app category, user gender and age may be a signal

- Most feature interactions are hidden in data and difficult to identify (e.g., “diaper and beer" rule)

Background

- **Factorization models** are the models where the interaction of several embeddings from different features is modeled into a real number by some operation such as inner product or neural network



(a) FM (b) DeepFM

Challenges

- Enumerate all feature interactions
  - Large memory and computation cost and difficult to be extended into high-order interactions
  - Useless interaction

- Require human efforts to identify important feature interactions
  - high labor cost
  - risks missing some counterintuitive (but important) interactions

$$y_{\mathrm{FM}}(x) = sigmoid\left(\boxed{\sum_{i=1}^{N} \boldsymbol{\omega_i} x_i} + \boxed{\sum_{i=1}^{N}\sum_{j=i+1}^{N} \langle \boldsymbol{v_i}, \boldsymbol{v_j} \rangle x_i x_j}\right)$$

$$y_{\mathrm{FFM}}(x) = sigmoid\left(\boxed{\sum_{i=1}^{N} \boldsymbol{\omega_i} x_i} + \boxed{\sum_{i=1}^{N}\sum_{j=i+1}^{N} \langle \boldsymbol{v_{i,f(j)}}, \boldsymbol{v_{j,f(i)}} \rangle x_i x_j}\right)$$

$$y_{\mathrm{FwFM}}(x) = sigmoid\left(\boxed{\sum_{i=1}^{N} \boldsymbol{\omega_i} x_i} + \boxed{\sum_{i=1}^{N}\sum_{j=i+1}^{N} A_{f(i),f(j)} \langle \boldsymbol{v_i}, \boldsymbol{v_j} \rangle x_i x_j}\right)$$

Outline

- **AutoML in Embedding Layer**
  - **NIS** - Neural Input Search for Large Scale Recommendation Models (KDD’2020)
  - **ESAPN** - Automated Embedding Size Search in Deep Recommender Systems (SIGIR’2020)
  - **AutoDim** - Field-aware Embedding Dimension Search in Recommender Systems (WWW’2021)
  - **AutoDis** - Automatic Discretization for Embedding Numerical Features in CTR Prediction (AAAI’2021)

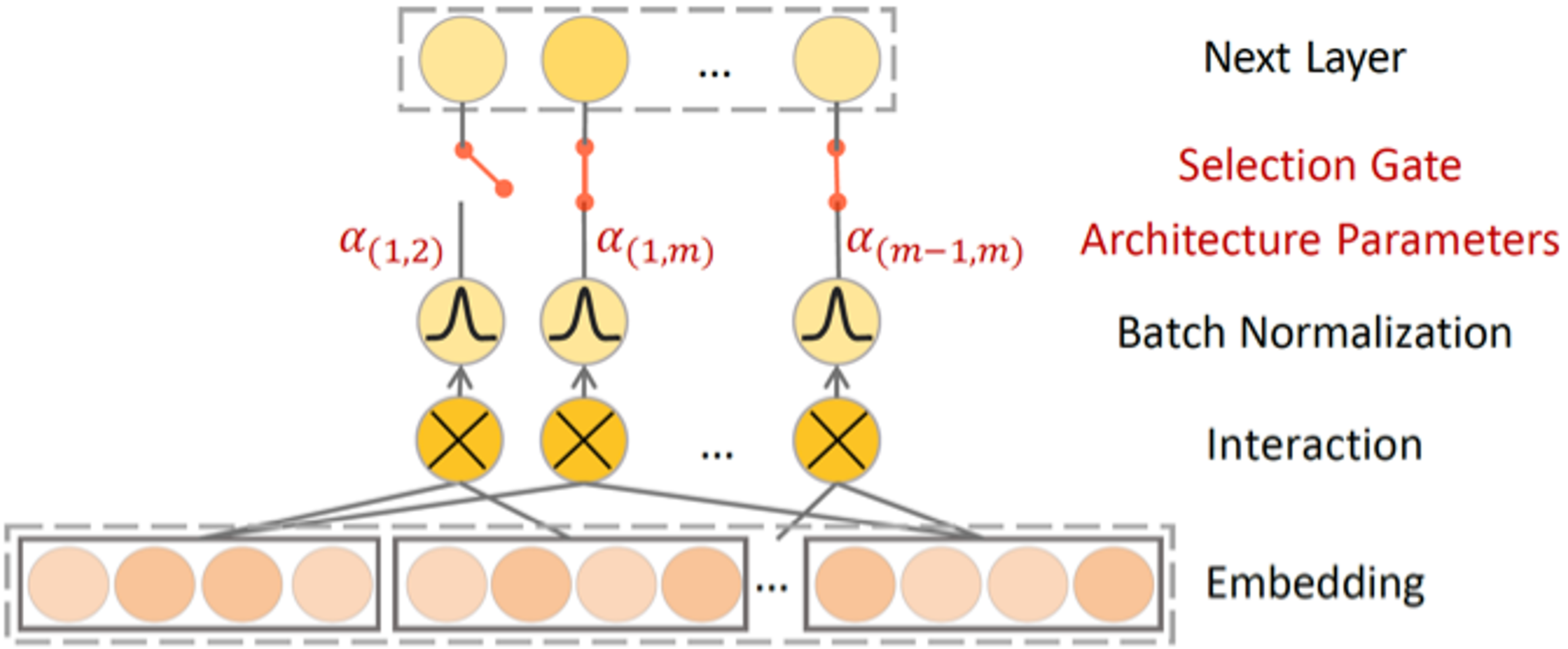
- **AutoML in Interaction Layer**
  - **AutoFIS** - Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction (KDD’2020)
  - **AutoGroup** - Automatic Feature Grouping for Modelling Explicit High-Order Feature Interactions in CTR Prediction (SIGIR’2020)
  - **AutoCTR** - Towards Automated Neural Interaction Discovery for Click-Through Rate Prediction (KDD’2020)

AutoFIS

- Search Stage
  - Detect useful feature interactions
- Retrain Stage
  - Retrain model with selected feature interactions

Search Stage

$$l_{\mathtt{AutoFIS}} = \langle w, x \rangle + \sum_{i=1}^{m} \sum_{j>i}^{m} \alpha_{(i,j)} \langle \boldsymbol{e}_i, \boldsymbol{e}_j \rangle$$ Indicator α = 0 or 1

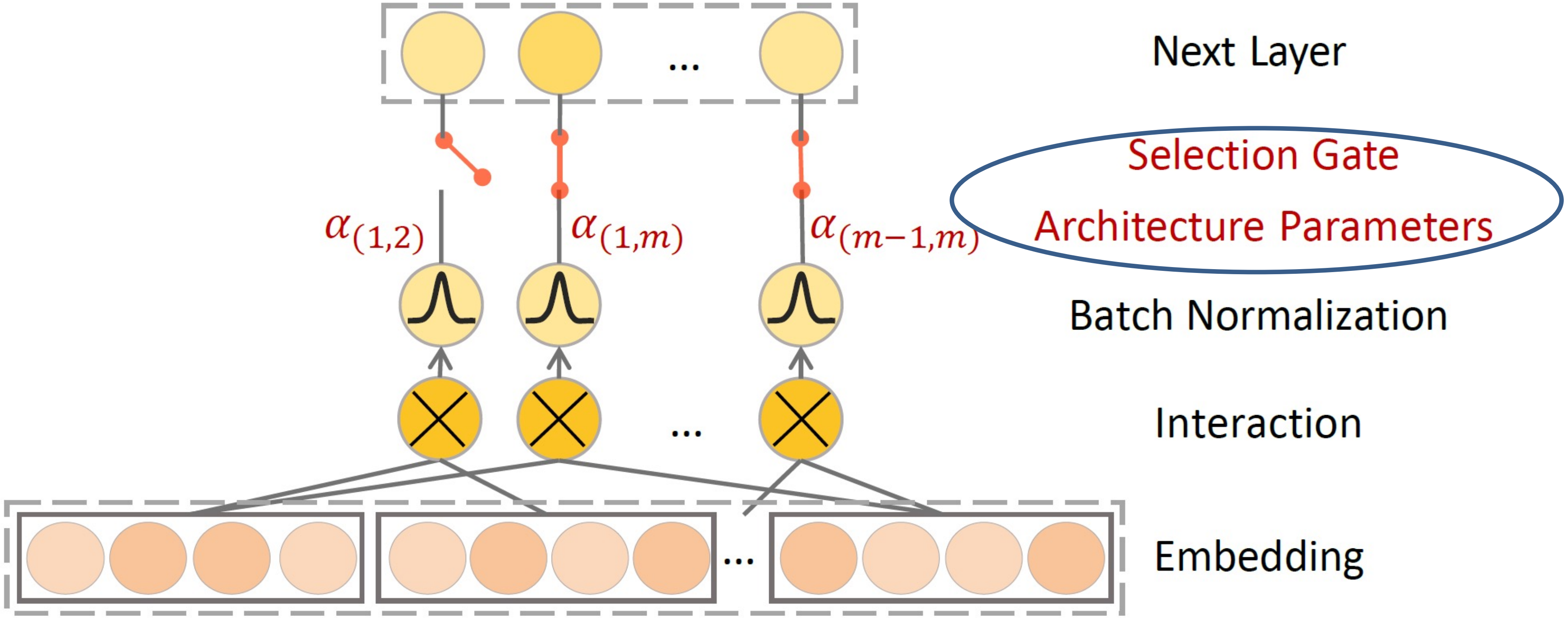


Search Stage

- Gate for each feature interaction
  - Huge search space $2^{C_m^2}$
- Discrete search space -> Continuous search space
  - Architecture parameters α

Search Stage

$$l_{\texttt{AutoFIS}} = \langle w, x \rangle + \sum_{i=1}^{m} \sum_{j>i}^{m} \alpha_{(i,j)} \langle \boldsymbol{e}_i, \boldsymbol{e}_j \rangle$$



Experiment Results in Huawei Dataset

Model	AUC	log loss	top	Rel. Impr
FM	0.8880	0.08881	100%	0
FwFM	0.8897	0.08826	100%	0.19%
AFM	0.8915	0.08772	100%	0.39%
FFM	0.8921	0.08816	100%	0.46%
DeepFM	0.8948	0.08735	100%	0.77%
AutoFM(2nd)	0.8944*	0.08665*	37%	0.72%
AutoDeepFM(2nd)	**0.8979***	**0.08560***	**15%**	1.11%

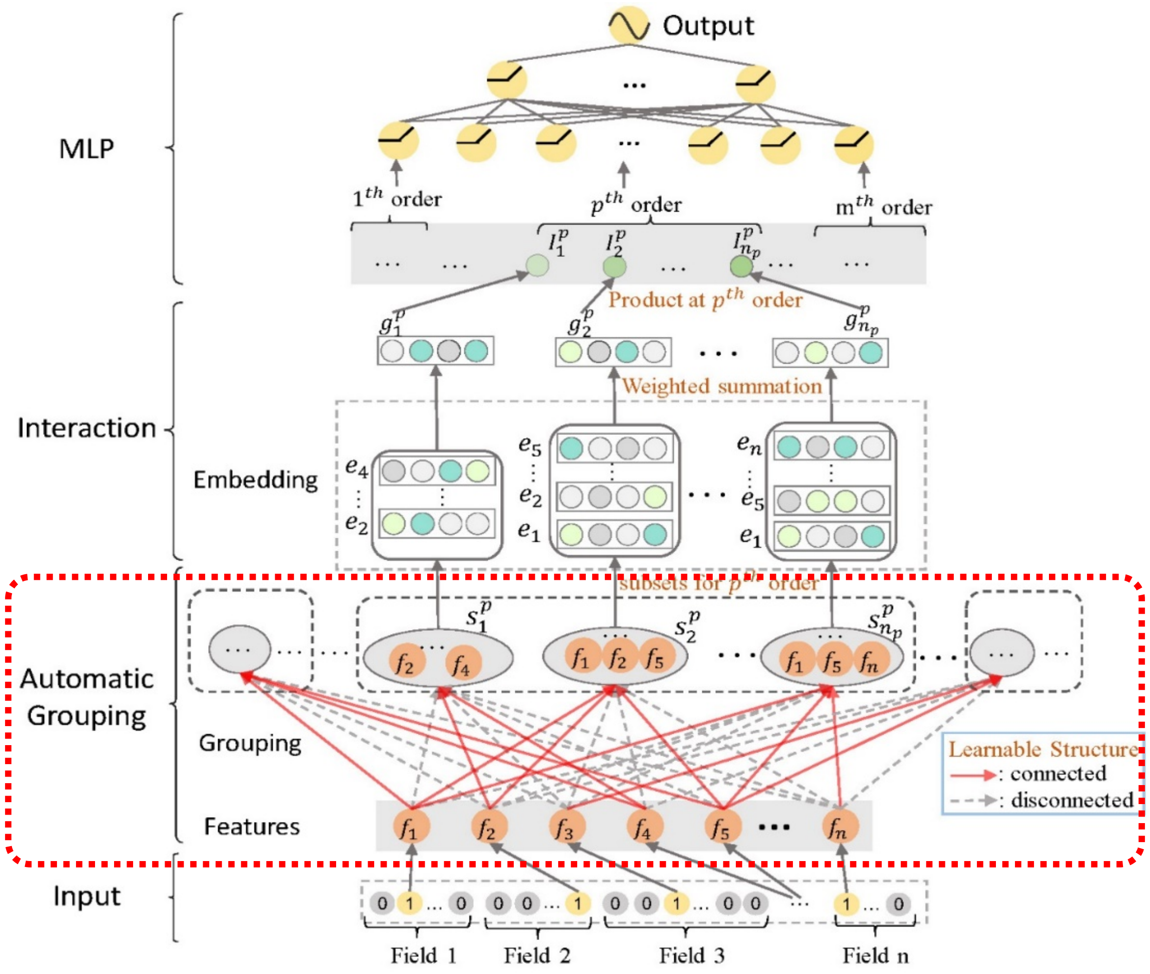
$*$ denotes statistically significant improvement (measured by t-test with p-value<0.005). AutoFM compares with FM and AutoDeepFM compares with all baselines.

Outline

- **AutoML in Embedding Layer**
  - **NIS** - Neural Input Search for Large Scale Recommendation Models (KDD’2020)
  - **ESAPN** - Automated Embedding Size Search in Deep Recommender Systems (SIGIR’2020)
  - **AutoDim** - Field-aware Embedding Dimension Search in Recommender Systems (WWW’2021)
  - **AutoDis** - Automatic Discretization for Embedding Numerical Features in CTR Prediction (AAAI’2021)

- **AutoML in Interaction Layer**
  - **AutoFIS** - Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction (KDD’2020)
  - **AutoGroup** - Automatic Feature Grouping for Modelling Explicit High-Order Feature Interactions in CTR Prediction (SIGIR’2020)
  - **AutoCTR** - Towards Automated Neural Interaction Discovery for Click-Through Rate Prediction (KDD’2020)

Automatic Feature Grouping Stage



Each feature is possible to be selected into the feature sets of each order.

- $\Pi_{i,j}^{p} \in \{0,1\}$: whether select feature f_i into the j^{th} set of order-p.

To make the selection differentiable, we relax the binary discrete value to a softmax over the two possibilities:

$$\overline{\Pi}_{i,j}^{p} = \frac{1}{1+\exp(-\alpha_{i,j}^{p})}\Pi_{i,j}^{p} + \frac{\exp\left(-\alpha_{i,j}^{p}\right)}{1+\exp\left(-\alpha_{i,j}^{p}\right)}(1-\Pi_{i,j}^{p}).$$

To learn a less-biased selection probability, we use Gumbel-Softmax:

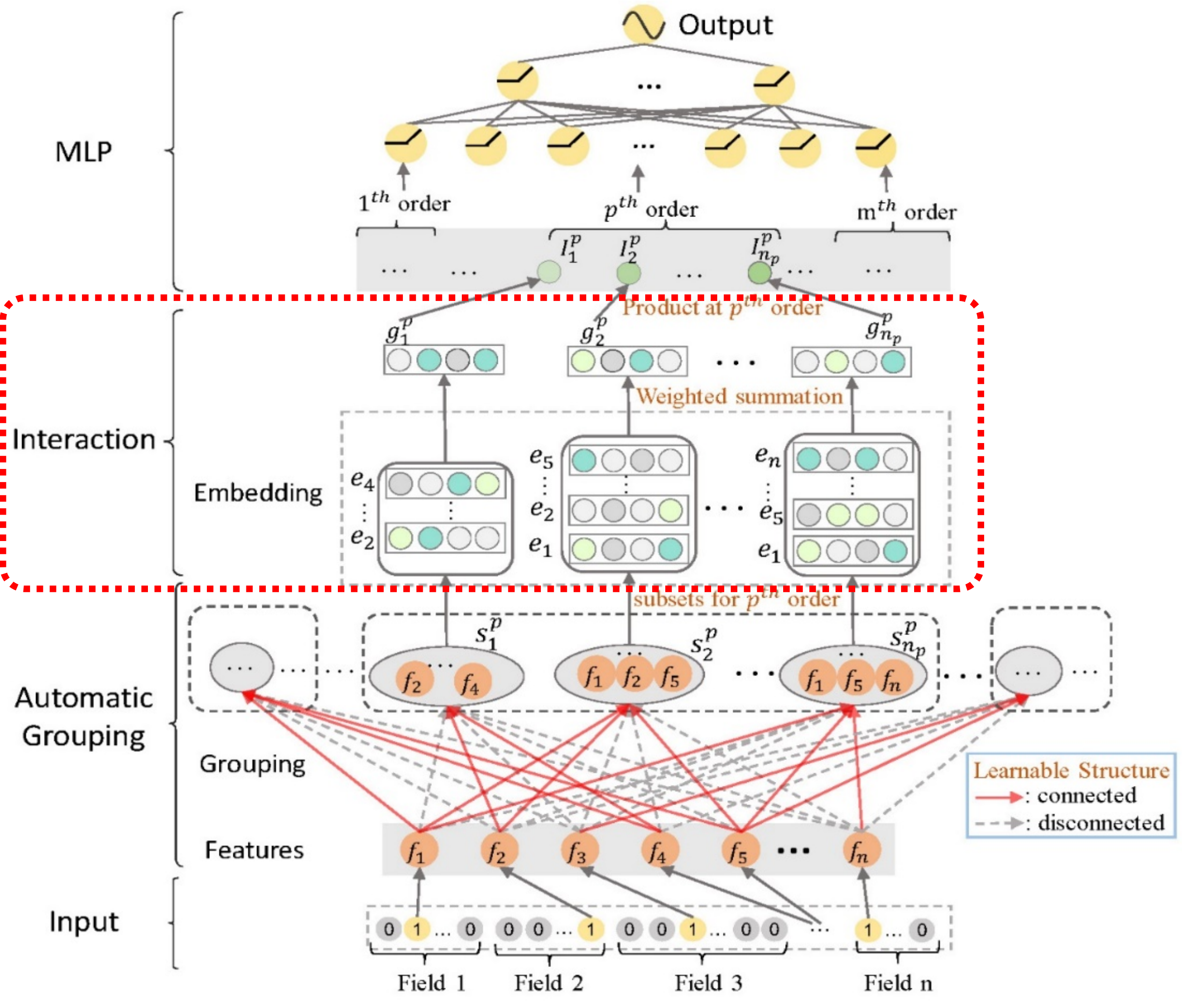
$$\left(\overline{\Pi}_{i,j}^{p}\right)_o = \frac{\exp(\frac{\log\alpha_o + G_o}{\tau})}{\sum_{o'\in\{0,1\}}\exp(\frac{\log\alpha_{o'} + G_{o'}}{\tau})} \text{ where } o \in \{0,1\}.$$

$$\alpha_0 = \frac{1}{1+\exp(-\alpha_{i,j}^{p})} \qquad \alpha_1 = \frac{\exp\left(-\alpha_{i,j}^{p}\right)}{1+\exp\left(-\alpha_{i,j}^{p}\right)}$$

$$G_o = -\log(-\log u) \text{ where } u \sim Uniform(0,1)$$

Trainable Parameters: $\{\alpha_{i,j}^{p}\}$

Interaction Stage

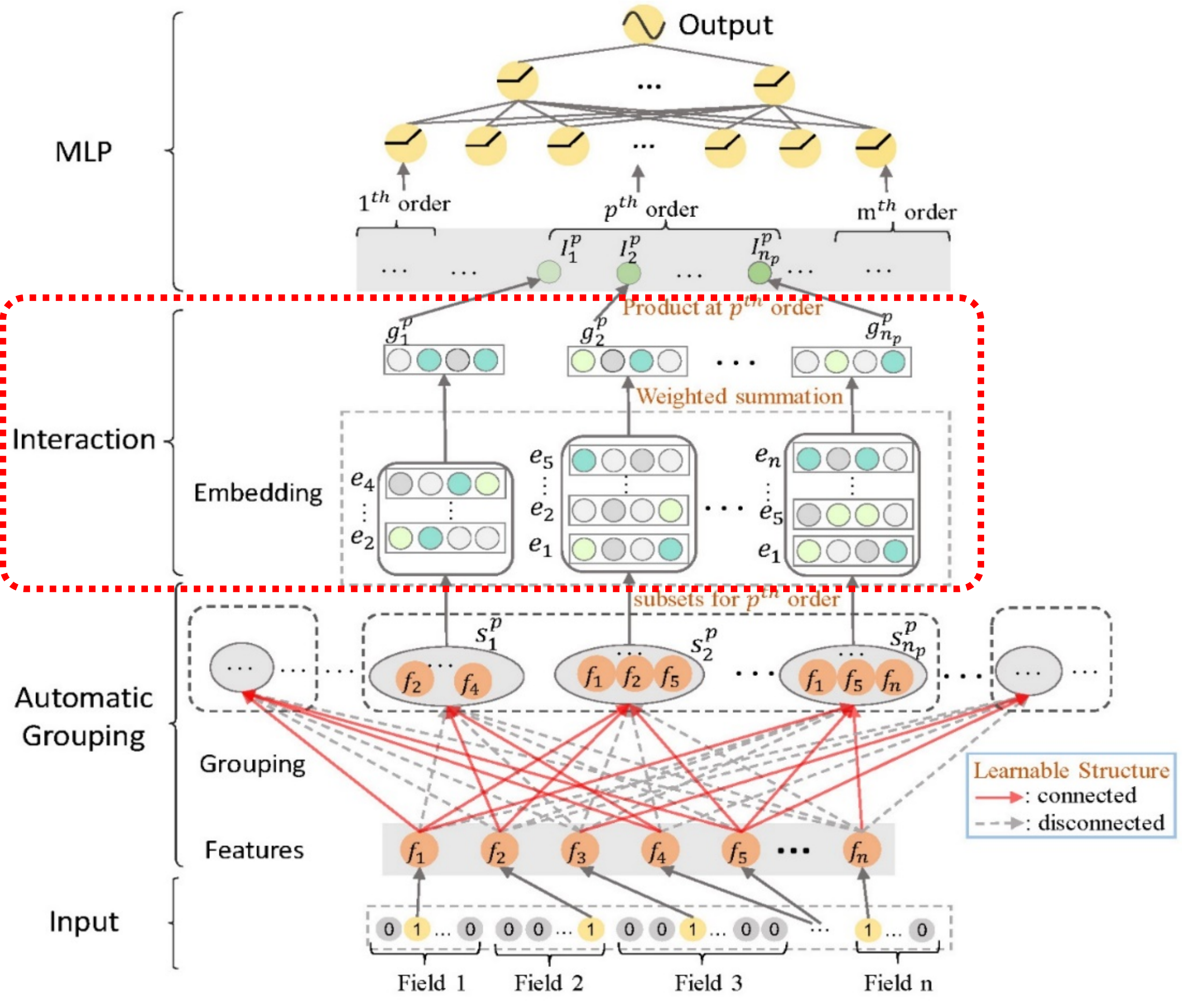


Feature set representation:

$$g_j^p = \sum_{f_i \in s_j^p} w_i^p \, e_i$$

s_j^p: the j^{th} feature set for order-p feature interactions.
e_i: embedding for feature f_i
w_i^p: weights of embeddings in feature set s_j^p.

Interaction Stage



Interaction at a given order:

- Inspired by the reformulation of FM:

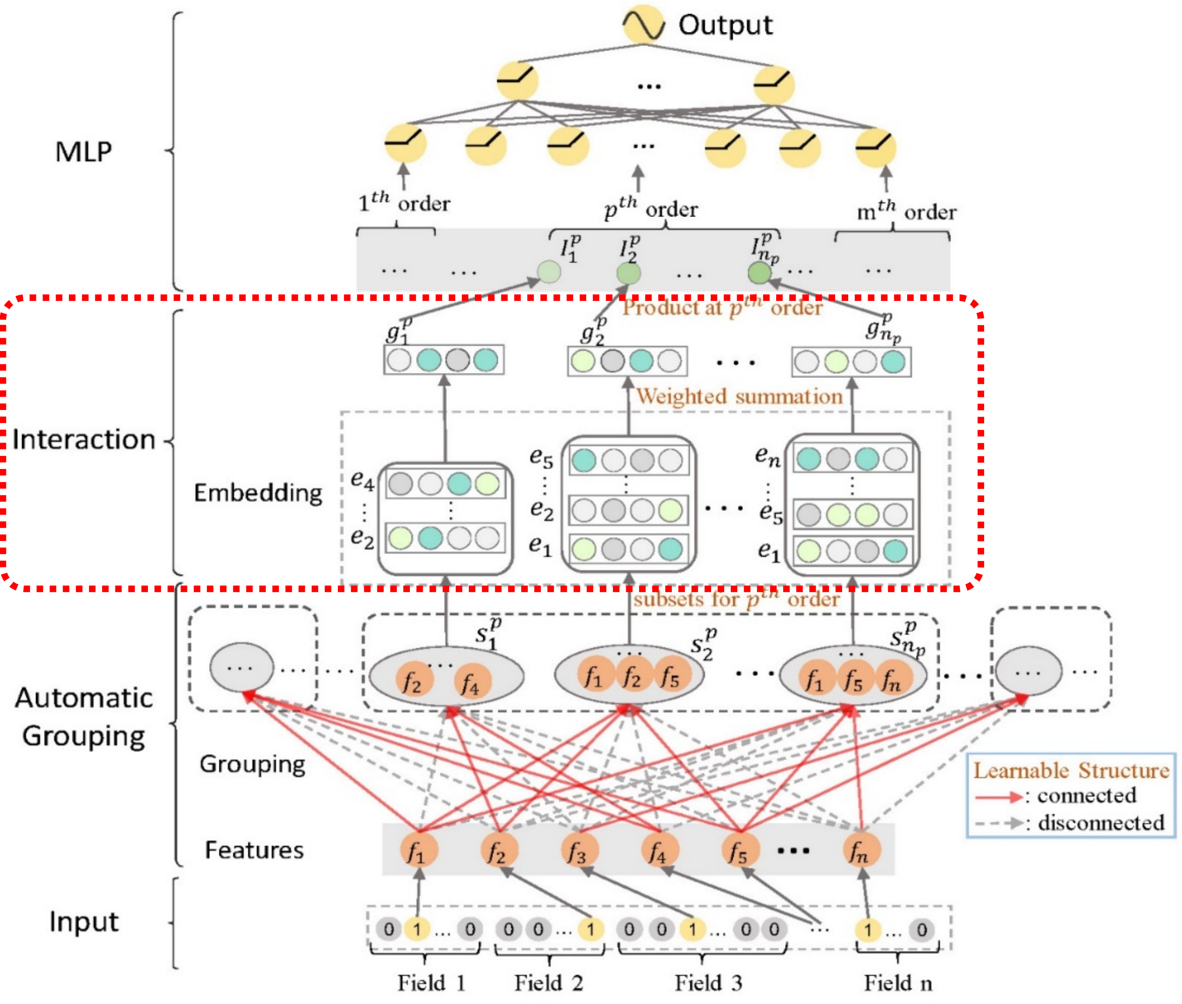
$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < e_i, e_j > x_i x_j$$

$$= w_0 + \sum_{i=1}^{n} w_i x_i + \frac{1}{2}\boxed{\left((\sum_{i=1}^{n} x_i e_i)^2 - \sum_{i=1}^{n} (x_i e_i)^2 \right)}$$

- The order-p interaction in a given set s_j^p is defined as:

$$I_j^p = \begin{cases} \boxed{\left(g_j^p\right)^p - \sum_{f_i \in s_j^k} \left(w_i^p e_i\right)^p} \in R, p \geq 2 \\ g_j^p \in R^{\kappa}, \qquad p = 1 \end{cases}$$

Interaction Stage



Parameter Training:
The structural parameters $\{\alpha_{i,j}^{p}\}$ and other normal parameters (embedding parameters and network parameters) are optimized alternatively in bi-level optimization (DARTS).

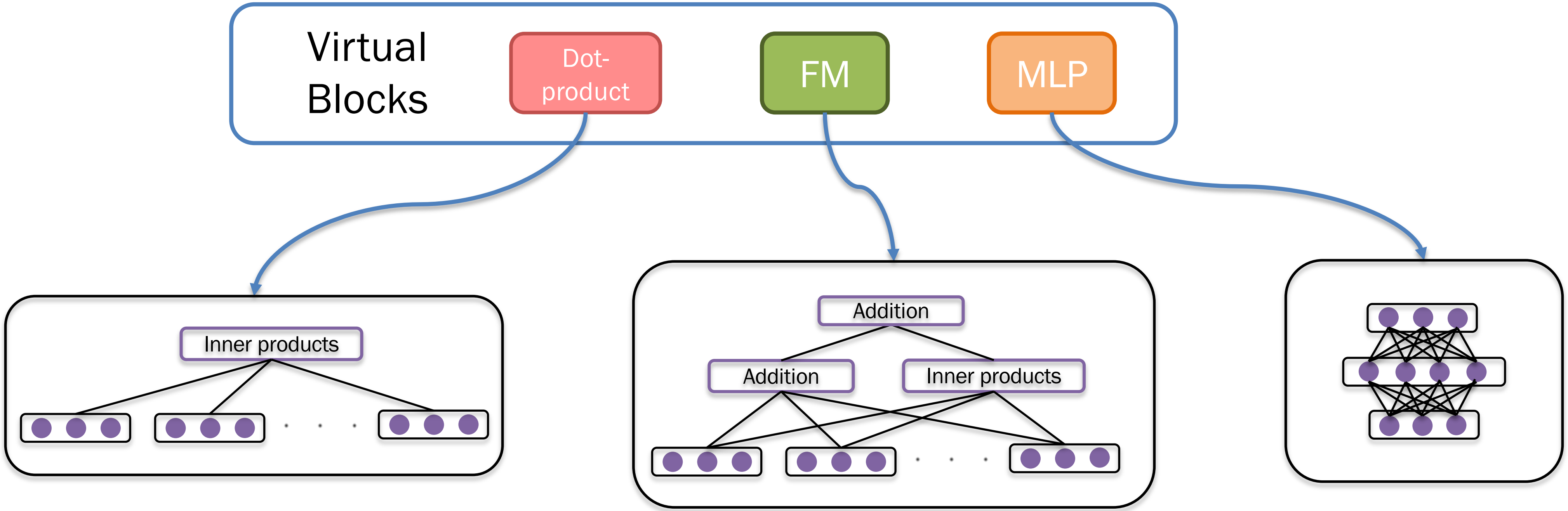
Outline

- **AutoML in Embedding Layer**
  - **NIS** - Neural Input Search for Large Scale Recommendation Models (KDD’2020)
  - **ESAPN** - Automated Embedding Size Search in Deep Recommender Systems (SIGIR’2020)
  - **AutoDim** - Field-aware Embedding Dimension Search in Recommender Systems (WWW’2021)
  - **AutoDis** - Automatic Discretization for Embedding Numerical Features in CTR Prediction (AAAI’2021)

- **AutoML in Interaction Layer**
  - **AutoFIS** - Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction (KDD’2020)
  - **AutoGroup** - Automatic Feature Grouping for Modelling Explicit High-Order Feature Interactions in CTR Prediction (SIGIR’2020)
  - **AutoCTR** - Towards Automated Neural Interaction Discovery for Click-Through Rate Prediction (KDD’2020)

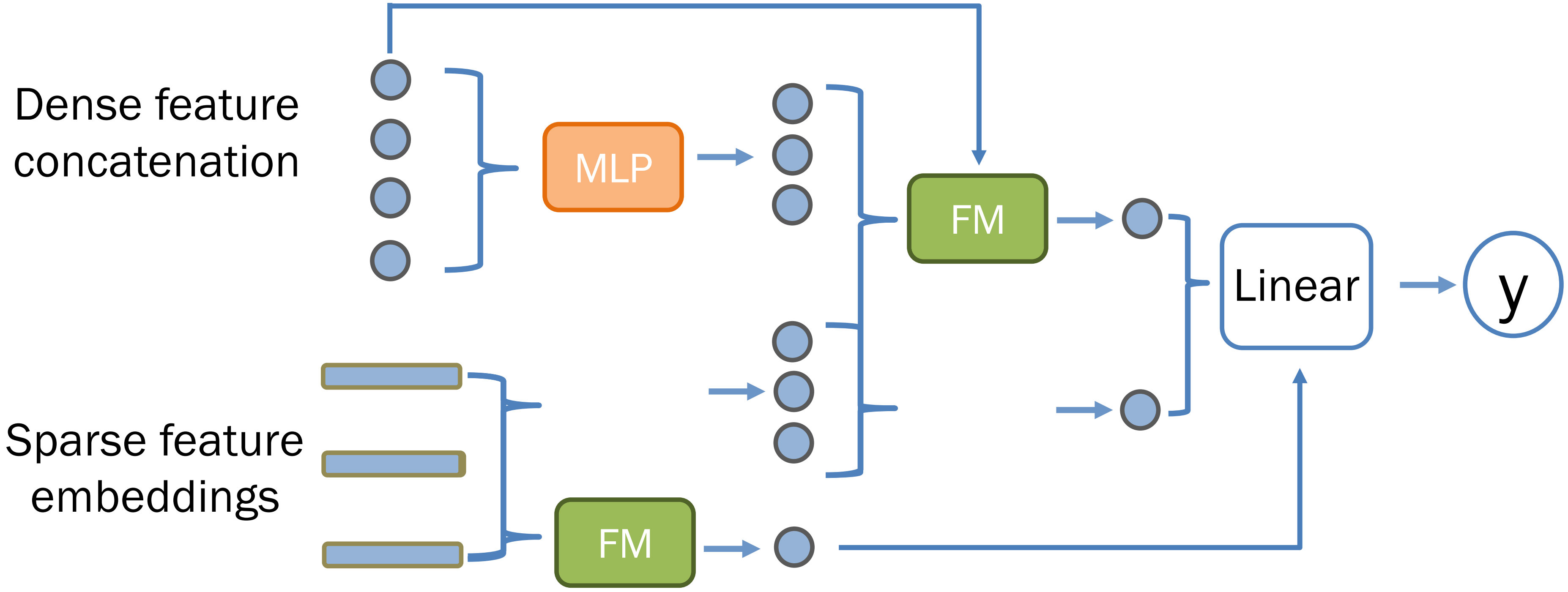
AutoCTR - Hierarchical Search Space

- Virtual block abstraction
  - Properties: functionality complementary, complexity aware, …
  - Examples: MLP block, dot-product block, factorization-machine block, …

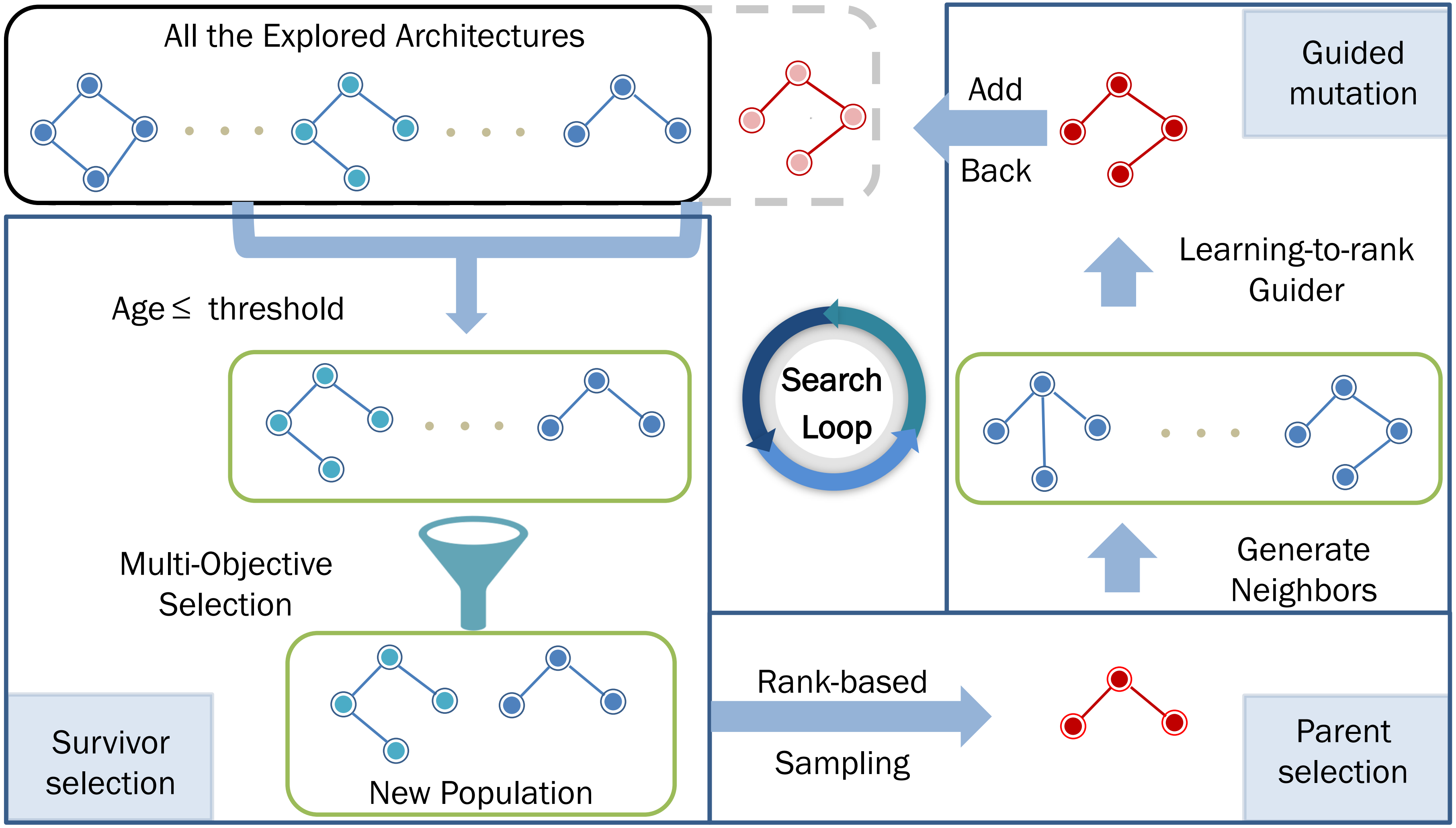


AutoCTR

- Search space construction
  - DAG of virtual blocks and grouped feature embeddings
  - Both block hyperparameters and connection among blocks are to be searched



AutoCTR - Multi-Objective Evolutionary Search Algorithm

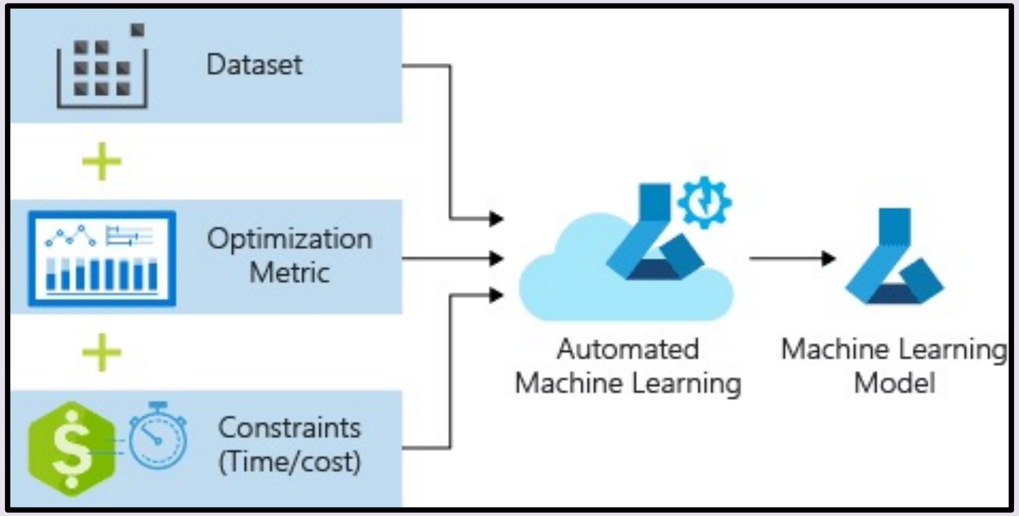


Conclusion

- Deep architectures are designed by the machine automatically

- Advantages
  - Less expert knowledge
  - Saving time and efforts
  - Different data → different architectures

Automated Machine Learning



Future Directions

- Applying AutoML to more tasks
  - Feature engineering, model selection, optimization algorithm, model evaluation, etc