AutoDim: Field-aware Embedding Dimension Search in Recommender Systems

Xiangyu Zhao[1], Haochen Liu[1], Hui Liu[1], Jiliang Tang[1]

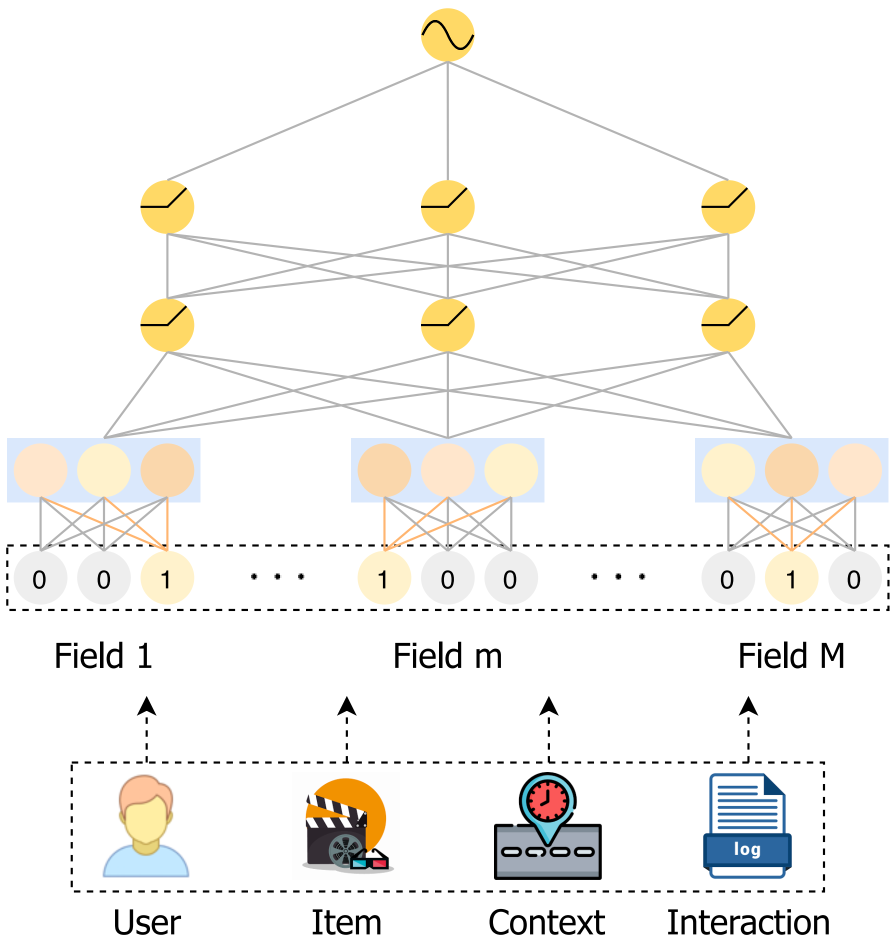
Weiwei Guo[2], Jun Shi[2], Sida Wang[2], Huiji Gao[2], Bo Long[3]

1: Data Science and Engineering Lab, Michigan State University

2: LinkedIn 3: JD.com

Embedding Components

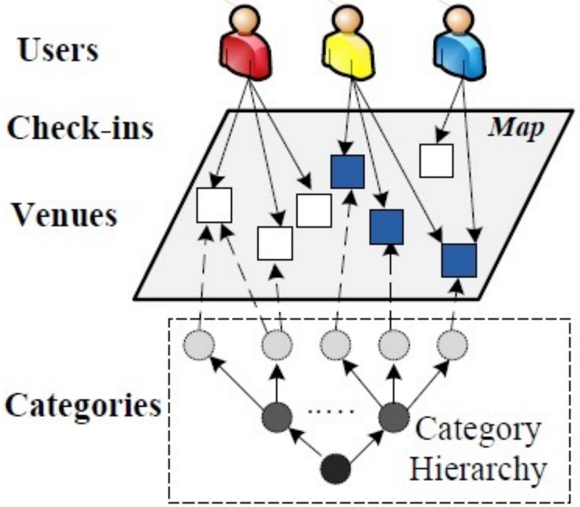
- Real-world recommender systems involve numerous feature fields
  - Users
    - e.g., gender and age
  - Items
    - e.g., category and price
  - Contextual information
    - e.g., time and location
  - Their interactions
    - e.g., ***users***' purchased ***items*** at ***location*** *A*



- Features → Embeddings
  - Unified dimension for all feature fields

Unified Embedding Dimension

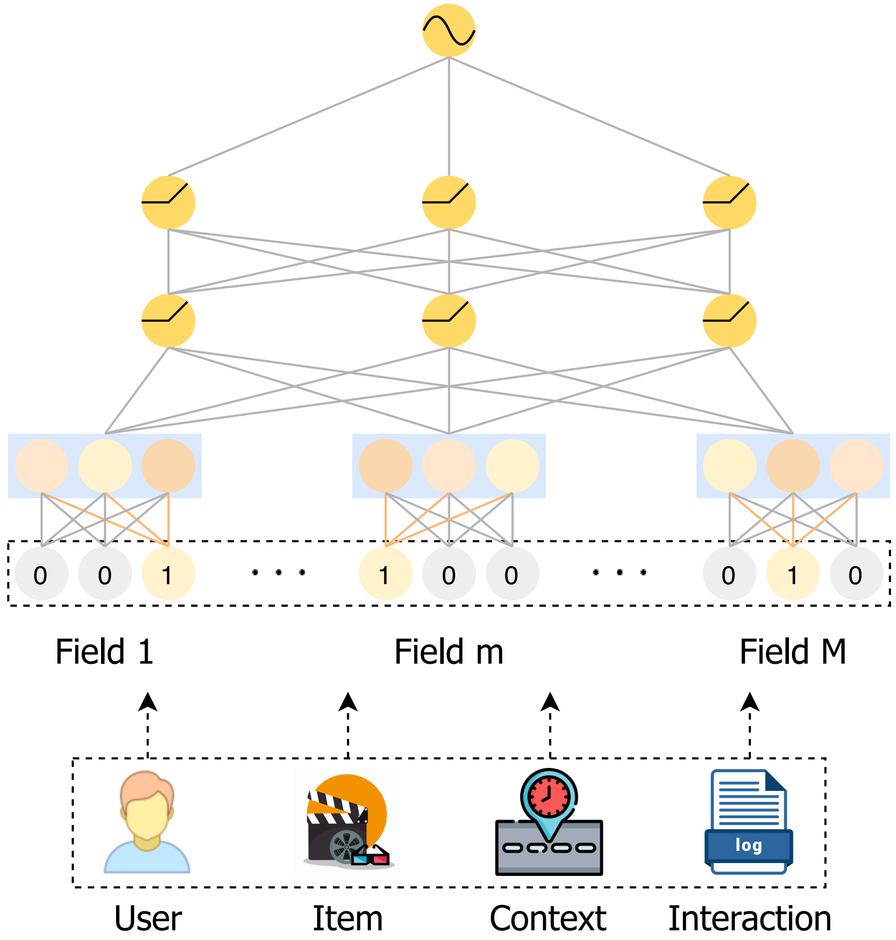
- Memory inefficiency problem
  - Embedding dimension → Capacity to encode information
  - Different feature fields have different cardinality



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

Challenges

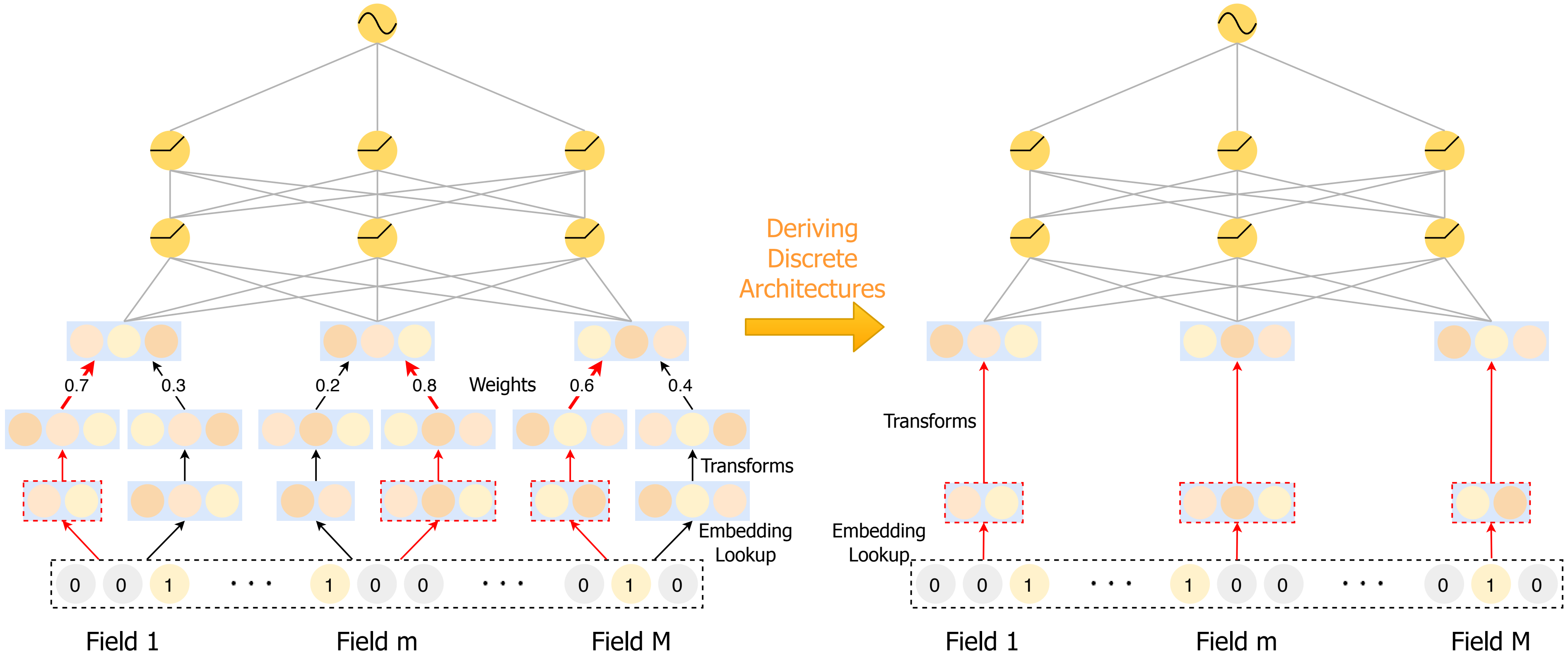
- Complex relationship
  - Embedding dimensions
  - Feature distributions
  - Neural network architectures
- Large search space
  - M feature field (M > 100)
  - K candidate dimensions
  - K^M selecion space



- AutoDim: Automated embedding dimension selection

Overview

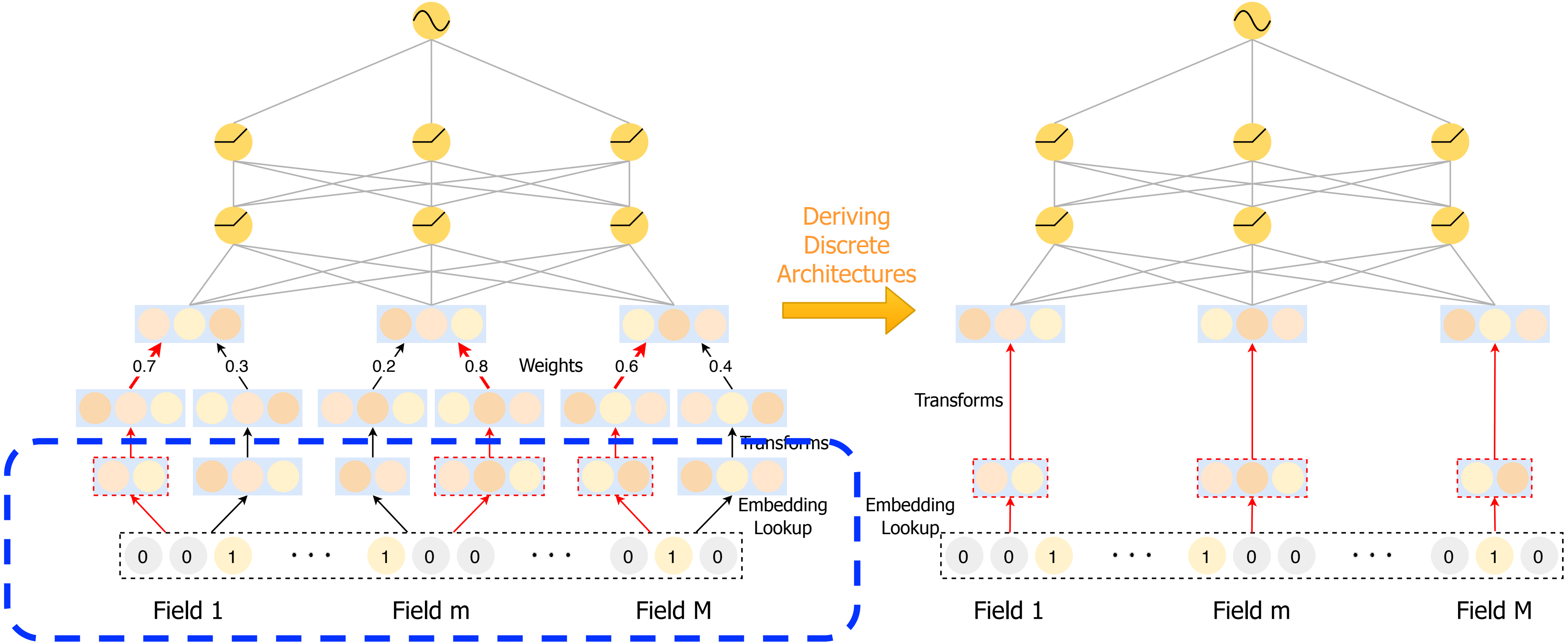
- Two-stage framework



(a) Dimension Search

(b) Parameter Retraining

Dimension Search Stage

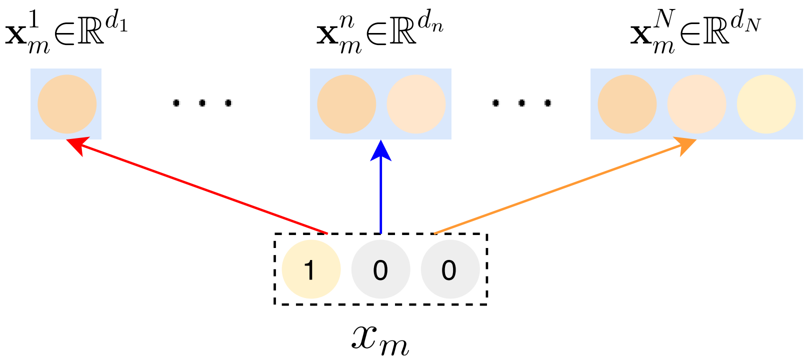


（a）Dimension Search

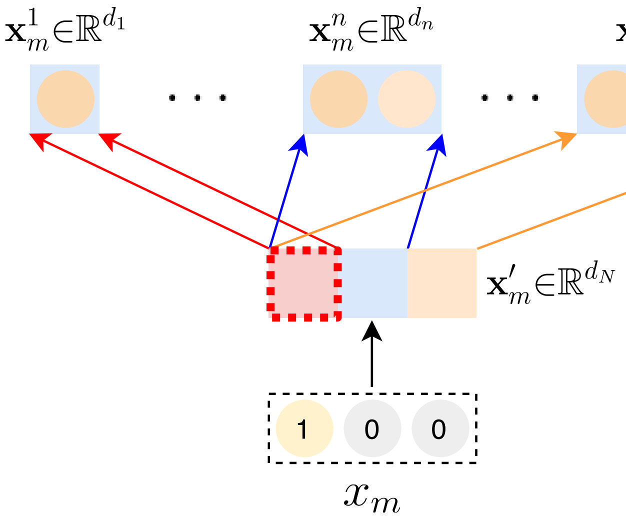
（b）Parameter Retraining

Candidate Embedding Assignment

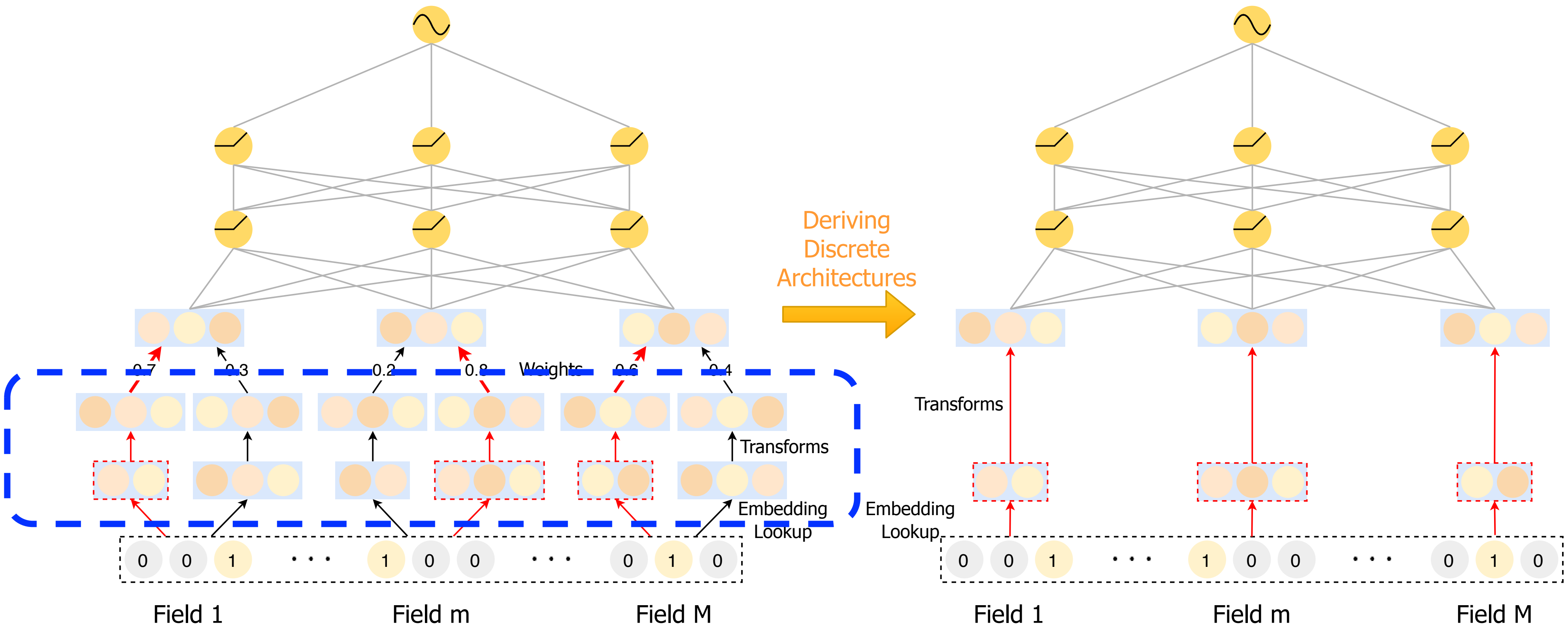
- Separate Embeddings



- Weight-sharing Embeddings



Dimension Search Stage

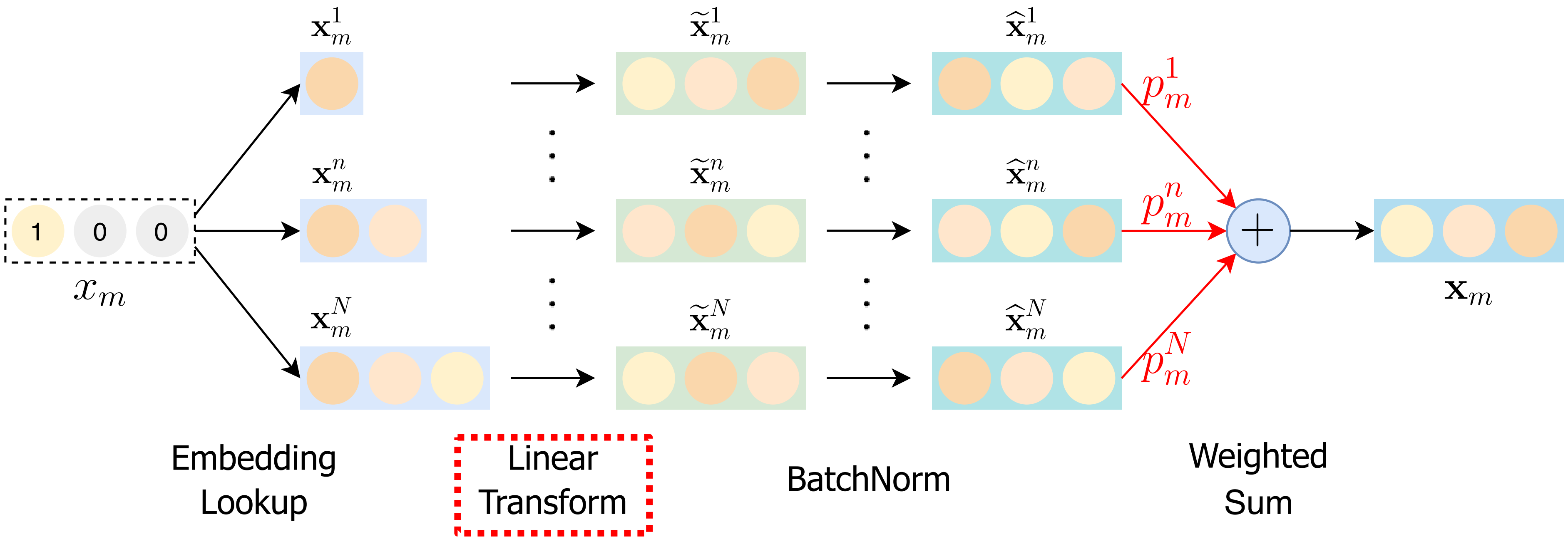


（a）Dimension Search

（b）Parameter Retraining

Unifying Various Dimensions

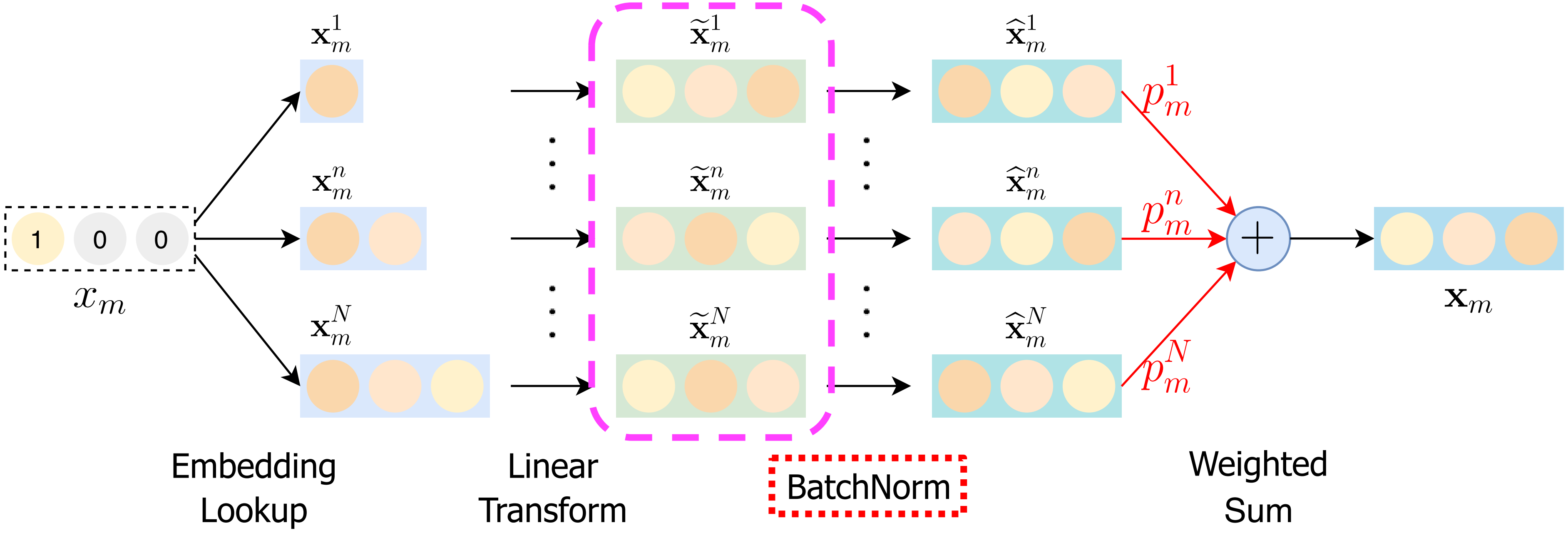
- Linear Transformation



$$\widetilde{\mathbf{x}}_m^n \leftarrow \mathbf{W}_n^\top \mathbf{x}_m^n + \mathbf{b}_n \quad \forall n \in [1, N]$$

Unifying Various Dimensions

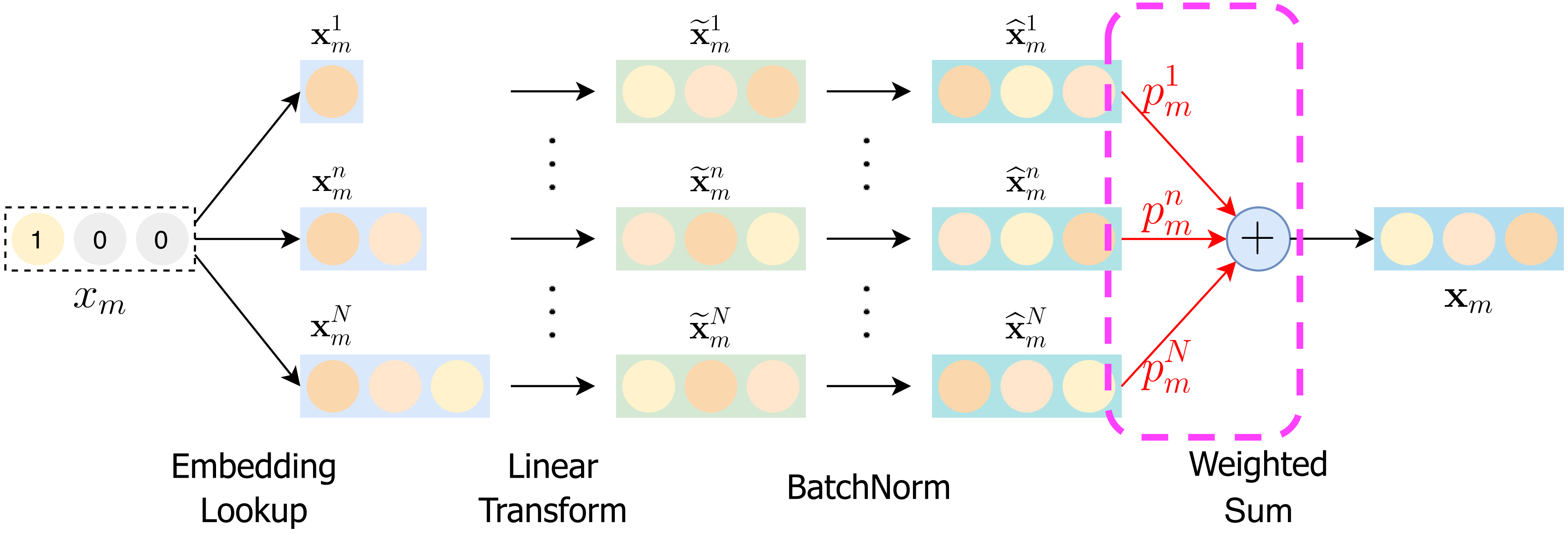
- Linear Transformation



$$\widetilde{\mathbf{x}}_m^n \leftarrow \mathbf{W}_n^\top \mathbf{x}_m^n + \mathbf{b}_n \quad \forall n \in [1, N]$$

Unifying Various Dimensions

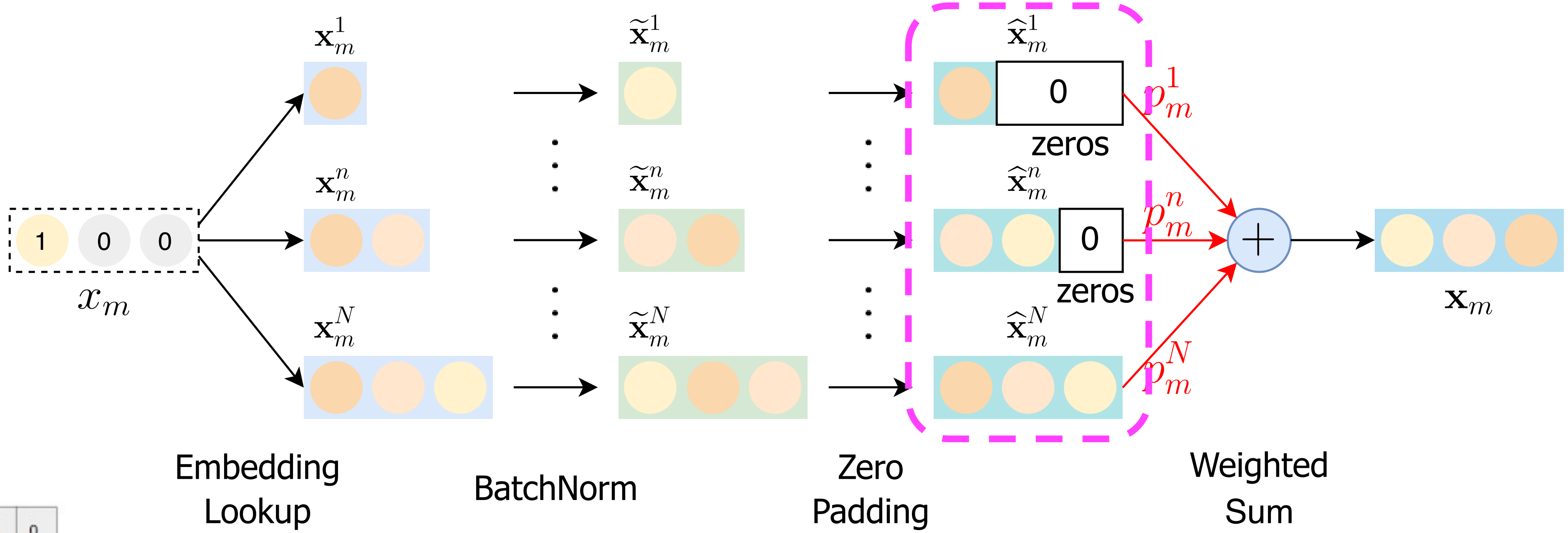
- Linear Transformation

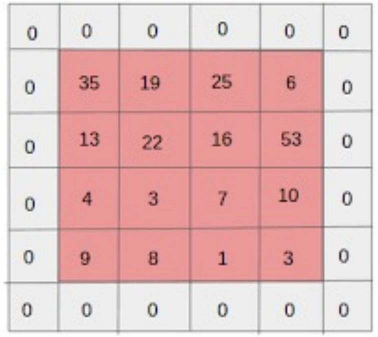


$$\widetilde{\mathbf{x}}_m^n \leftarrow \mathbf{W}_n^\top \mathbf{x}_m^n + \mathbf{b}_n \quad \forall n \in [1, N]$$

Unifying Various Dimensions

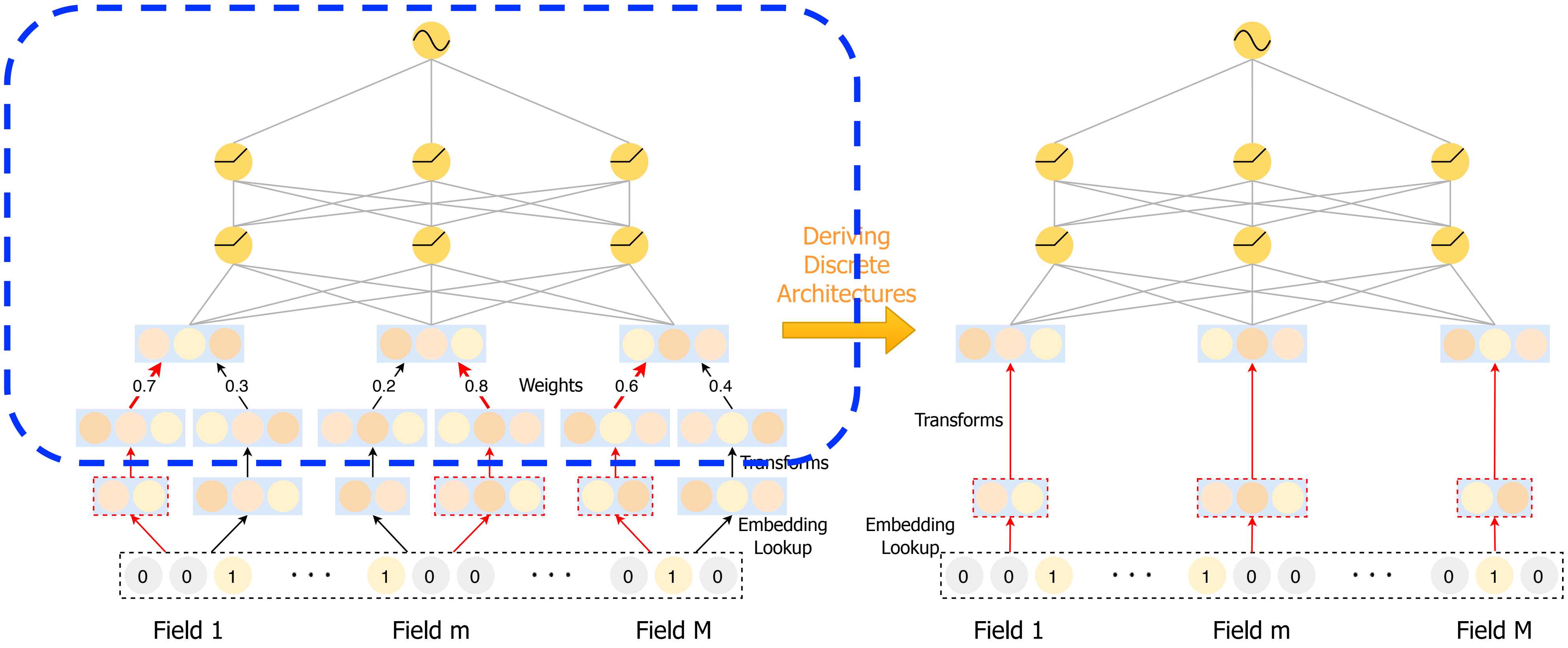
- Zero Padding





0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

Dimension Search Stage

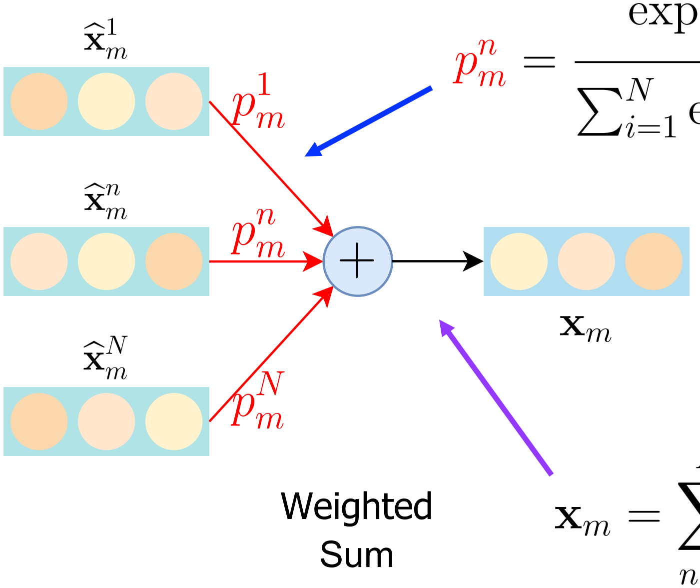


(a) Dimension Search

(b) Parameter Retraining

Dimension Selection

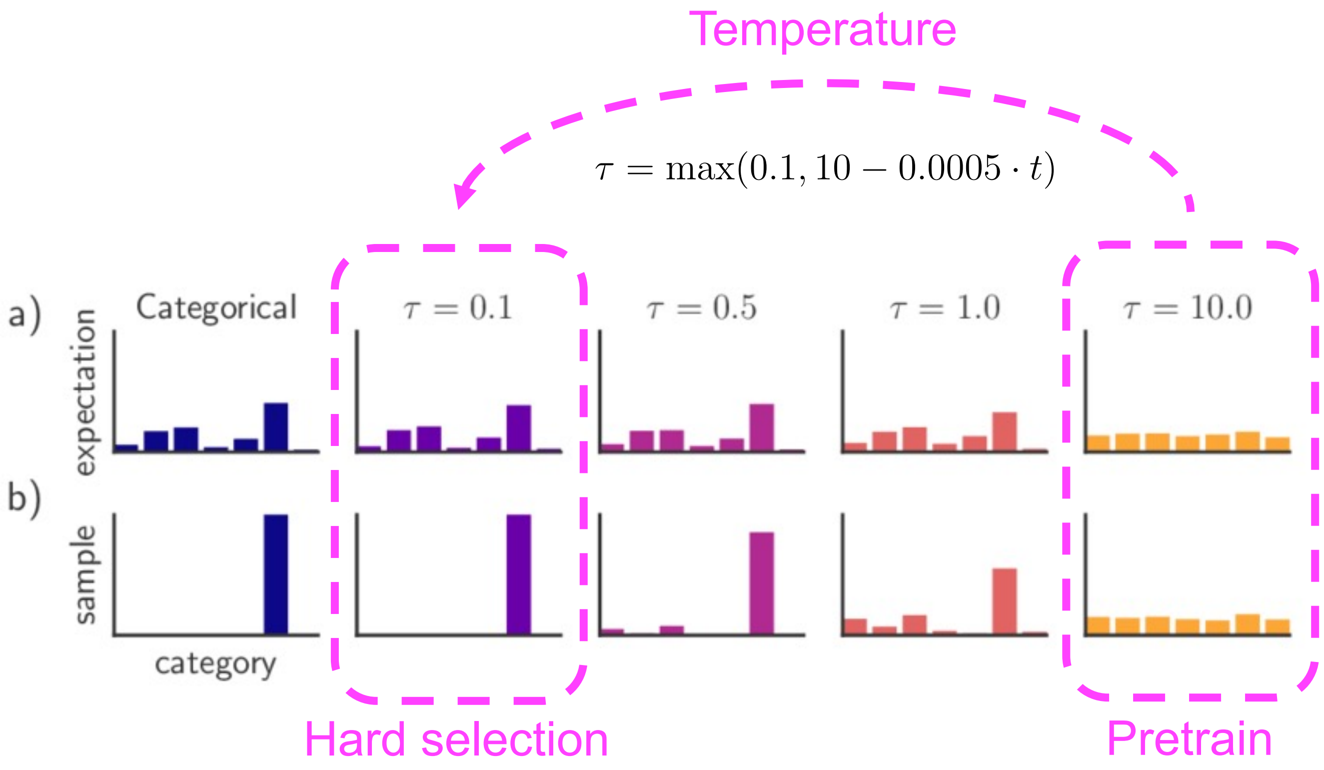
- Hard selection from categorical distribution
  - Search framework is non-differentiable
- Gumbel-softmax approximates the hard selection



$$p_m^n = \frac{\exp\left(\frac{\log(\alpha_m^n)+g_n}{\tau}\right)}{\sum_{i=1}^{N}\exp\left(\frac{\log(\alpha_m^i)+g_i}{\tau}\right)}$$

Temperature

$$\mathbf{x}_m = \sum_{n=1}^{N} p_m^n \cdot \widehat{\mathbf{x}}_m^n$$



Inference and Loss

- Inference layer
  - Hidden layer

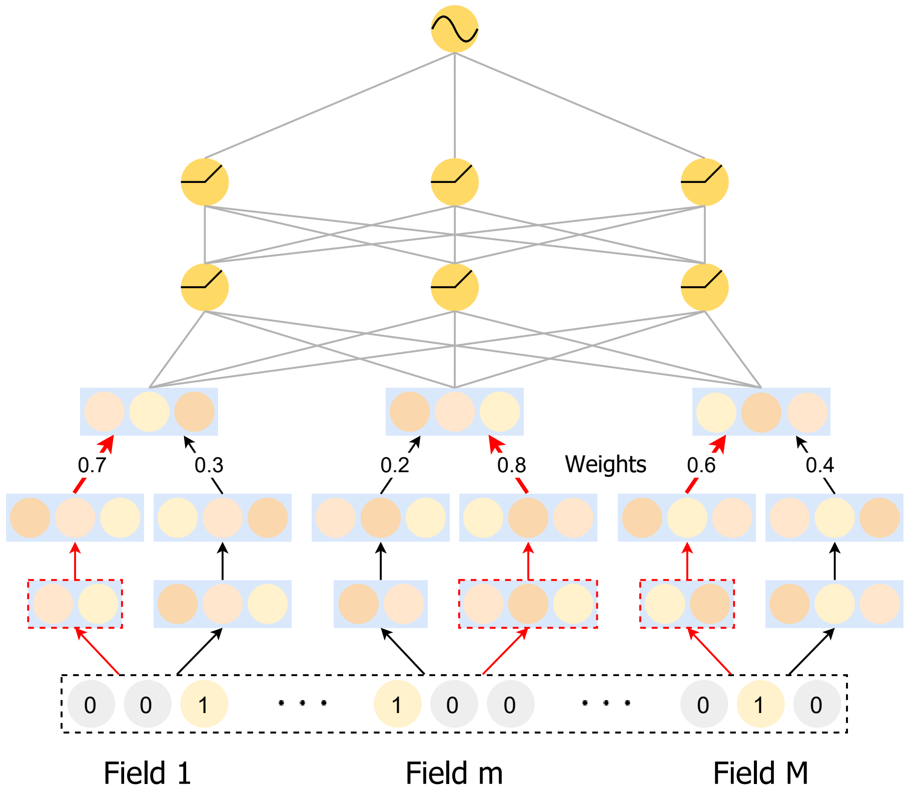
$$\mathbf{h}_l = \sigma\left(\mathbf{W}_l^\top \mathbf{h}_{l-1} + \mathbf{b}_l\right)$$

  - Output layer

$$\hat{y} = \sigma\left(\mathbf{W}_o^\top \mathbf{h}_L + \mathbf{b}_o\right)$$

- Click-Through Rate prediction
  - y = 1: click   0: non-click

$$\mathcal{L}(\hat{y}, y) = -y \log \hat{y} - (1-y)\log(1-\hat{y})$$



Bilevel Optimization

- Two set of parameters
  - Normal deep RecSys parameters **W**
  - Architectural weights **α** (weighted sum probabilities)
- Alternately update **W** on the training set and **α** on the validation set

$$\min_{\boldsymbol{\alpha}} \mathcal{L}_{val}\big(\mathbf{W}^*(\boldsymbol{\alpha}), \boldsymbol{\alpha}\big)$$
$$s.t.\ \mathbf{W}^*(\boldsymbol{\alpha}) = \arg\min_{\mathbf{W}} \mathcal{L}_{train}(\mathbf{W}, \boldsymbol{\alpha}^*)$$

where $\mathcal{L} = -y \log \hat{y} - (1-y)\log(1-\hat{y})$

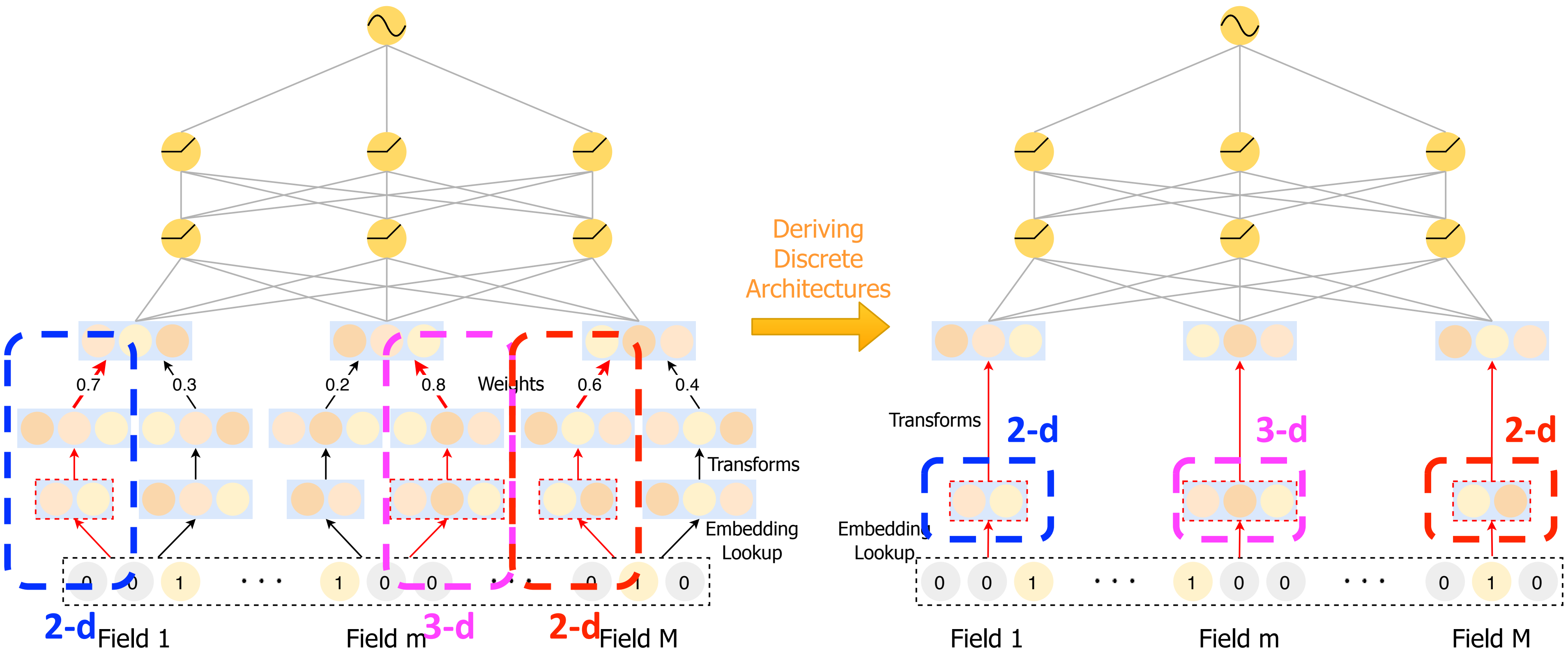
Algorithm 1 DARTS based Optimization for AutoDim.

Input: the features $(x_1, \cdots, x_M)$ of user-item interactions and the corresponding ground-truth labels y

Output: the well-learned DLRS parameters $\mathbf{W}^*$; the well-learned weights on various embedding spaces $\boldsymbol{\alpha}^*$

```
while not converged do
    Sample a mini-batch of user-item interactions from validation data
    Update α by descending ∇_α L_val(W*(α), α) with the approximation in Eq.(12)
    Collect a mini-batch of training data
    Generate predictions ŷ via DLRS with current W and architectural weights α
    Update W by descending ∇_W L_train(W, α)
end while
```

Parameter Retraining Stage



（a）Dimension Search

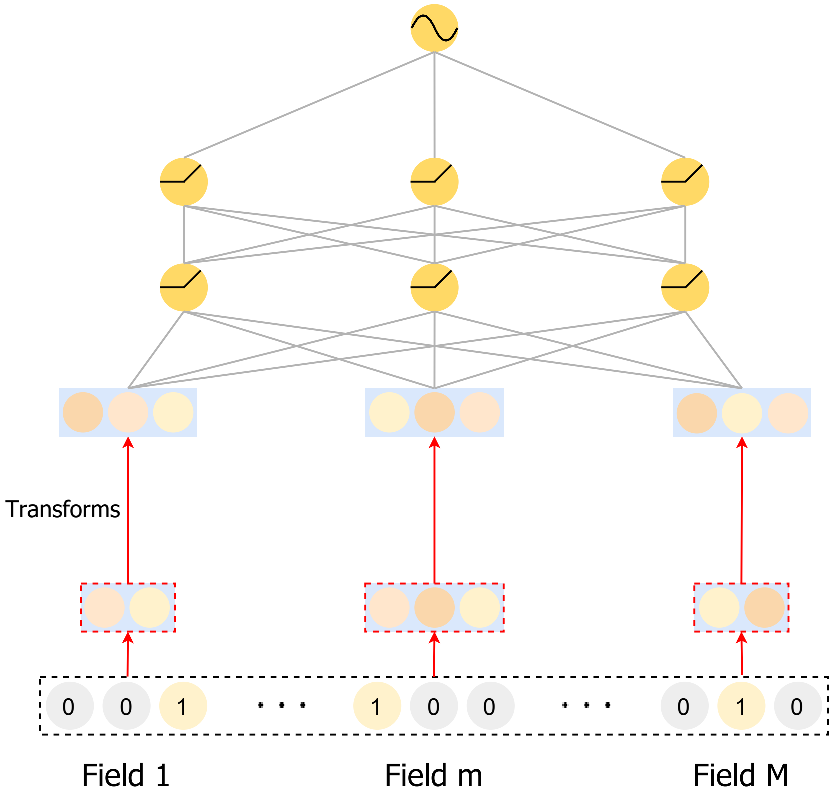
（b）Parameter Retraining

Parameter Retraining Stage

- Retraining stage is necessary
  - To eliminate the influence of suboptimal embedding dimensions
- Unify the selected embeddings into the same dimension
  - Interaction among feature fields

$$y_{\mathrm{FM}}(x) = sigmoid\left(\sum_{i=1}^{N} \boldsymbol{\omega}_i x_i + \sum_{i=1}^{N}\sum_{j=i+1}^{N} \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle x_i x_j\right)$$

- BatchNorm is no longer in use
  - There is no competition between candidate embeddings



Experimental Settings

- AutoDim is general for any deep recommender systems with embedding layer
- Recommendation models
  - AutoDim → FM, W&D and DeepFM
- Pubilc benchmark datasets
  - Criteo and Avazu
- Candidate dimensions
  - {2,8,16,24,32}

Table 1: Statistics of the datasets.

Data	Criteo	Avazu
# Interactions	45,840,617	40,428,968
# Feature Fields	39	22
# Sparse Features	1,086,810	2,018,012

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)

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)
- Unified dimension → Wasting memory, and downgrading model performance

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)
- Unified dimension → Wasting memory, and downgrading model performance
- Different feature values with various dimensions → Large search space

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)
- Unified dimension → Wasting memory, and downgrading model performance
- Different feature values with various dimensions → Large search space
- RaS → Large search space, AD-s → Over-fitting problem

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)
- Unified dimension → Wasting memory, and downgrading model performance
- Different feature values with various dimensions → Large search space
- RaS → Large search space, AD-s → Over-fitting problem
- AutoDim → Best AUC and Logloss, and saving 70~80% embedding parameters

Conclusion

- AutoDim can automatically select the proper dimensions for all feature fields
  - It can be applied to any deep recommender systems with embedding layer
  - It can save embedding parameters
  - It can improve recommendation performance

zhaoxi35@msu.edu