MASCOT: A Quantization Framework for Efficient Matrix Factorization in Recommender Systems

ICDM 2021
IEEE International Conference on Data Mining
7 – 10 DECEMBER 2021
AUCKLAND NEW ZEALAND

Yunyong Ko[1], Jae–Seo Yu[1], Hong–Kyun Bae[1], Yongjun Park[1], Dongwon Lee[2], and Sang–Wook Kim[1]

Hanyang University, Republic of Korea[1]

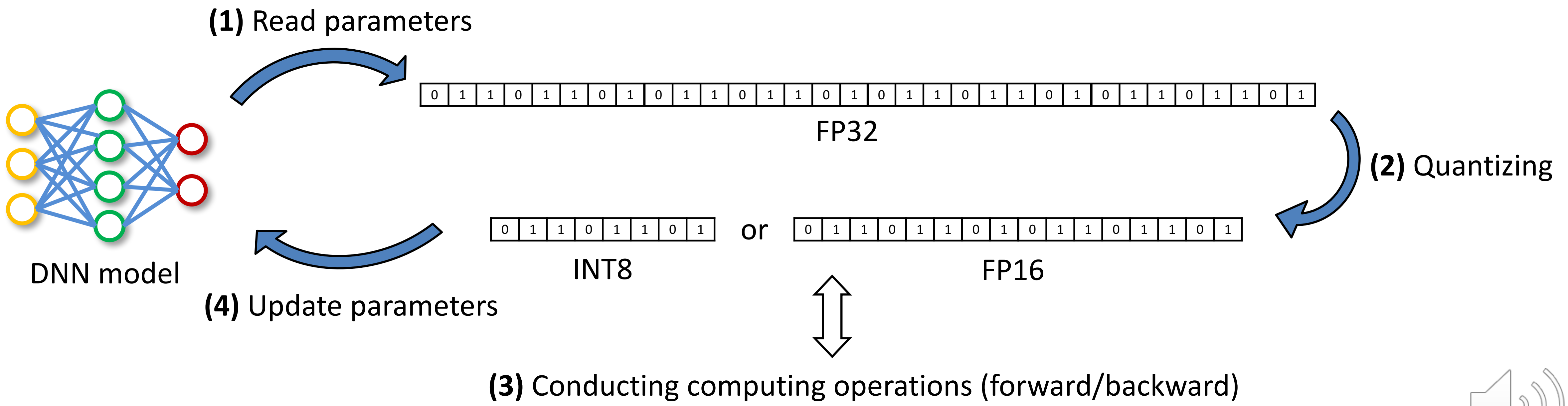
The Pennsylvania State University, PA, USA[2]

Table of Contents

Background

Quantization

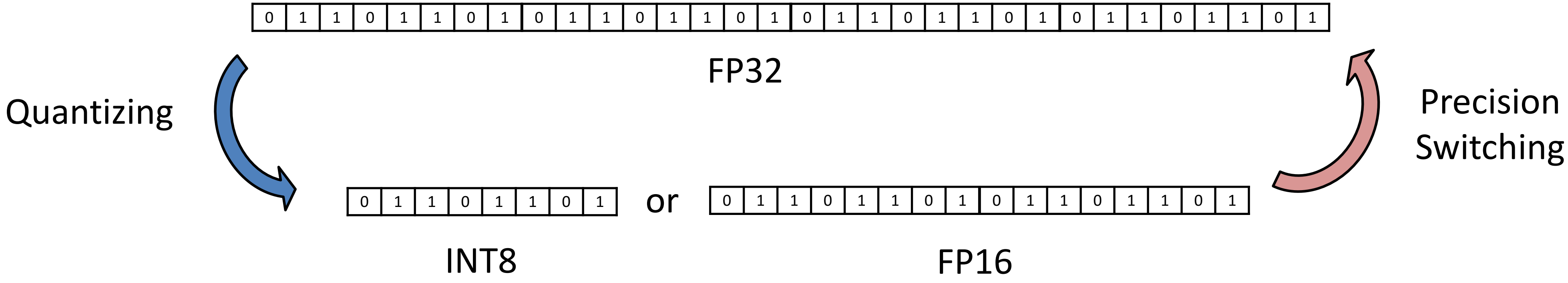
- Converting the parameter value into a lower precision
  - Ex) FP32 (i.e., single precision) → FP16 (i.e., half precision)
- For improving the training performance of DNN models
  - To reduce *the overhead of computing operations* and *the memory usage*



Background

☐ Presicion switching

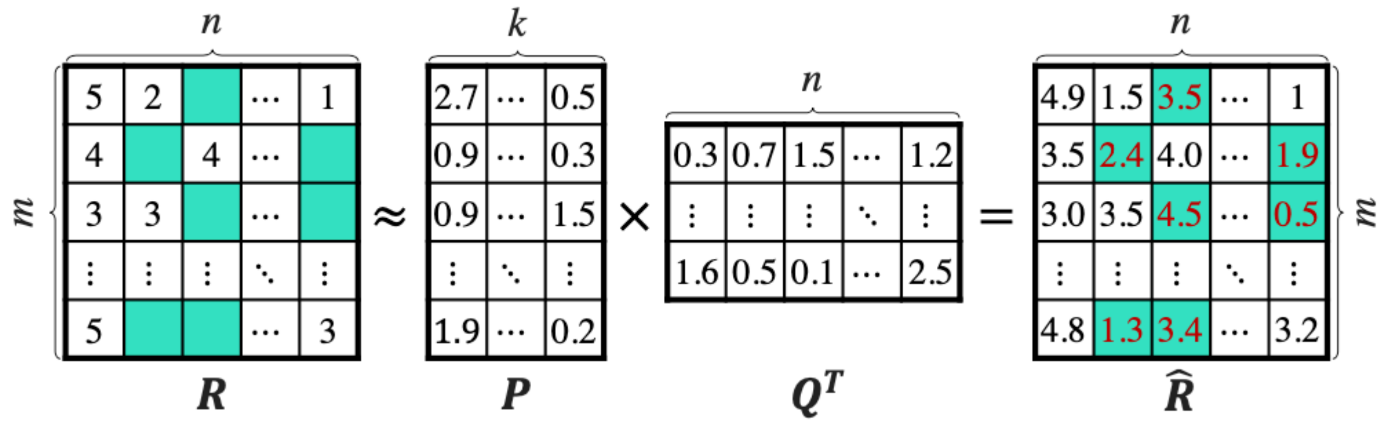
- The quantization error may result in degrading the model quality
  - ☐ Quantization error: the error caused by low precision
- Switching back the low precsion to high precision to prevent the loss of accuracy
  - ☐ i.e., FP16 (i.e., half precision) → FP32 (i.e., single precision)



Background

☐ Matrix Factorization (MF)

- One of the popular *collaborative filtering* algorithms in recommender systems (RS)
- Aiming to obtain the latent matrices ***P*** and ***Q*** (satisfying $R \approx P \times Q^T$)



☐ Challenge

- The growing scale of users/items and model architectures can significantly slow down the training of MF models

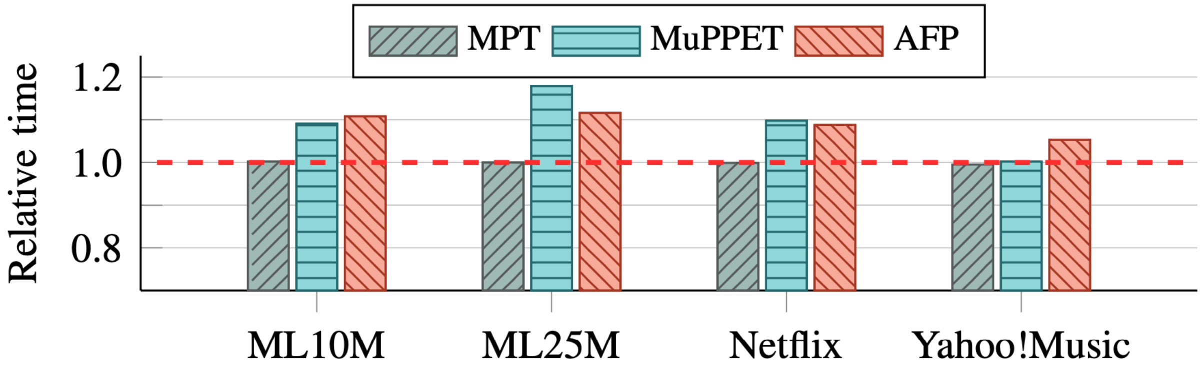
Motivation

Our question

- *“Can the proven quantization be adopted to improve the training of MF models in recommender systems?”*

Preliminary experiment with four RS datasets

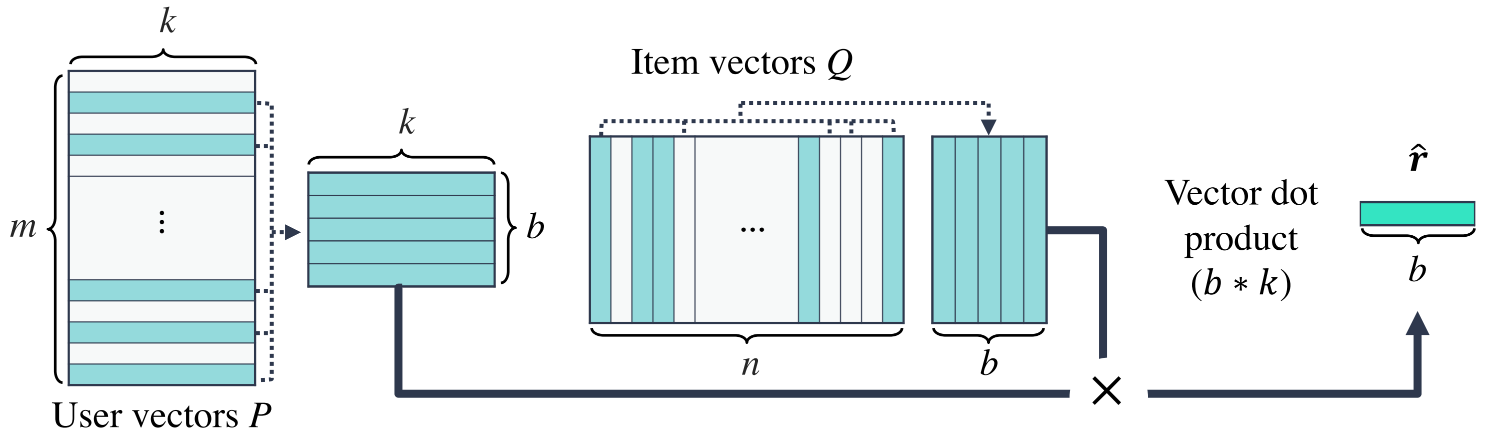
- SOTA quantization methods are *rarely effective* in MF model training



MPT: P. Micikevicius et al. Mixed precision training. In *ICLR*, 2018.
MuPPET: A. Rajagopal et al., MuPPET: A precision-switching strategy for quantised fixed-point training of CNNs. In *ICML*, 2020.
AFP: X. Zhang et al. Fixed-point back-propagation training. In *CVPR,* 2020.

Analysis on the Training of a MF Model

- **Training of a MF model is much more *memory-intensive* than that of a DNN model (*Observation 1*)**
  - Memory and computation costs of a MF model (vector dot product)
    - Memory cost $(b * 2k)$ **>** Computation cost $(b * k)$
  - *Little room* for the performance improvement by the quantization

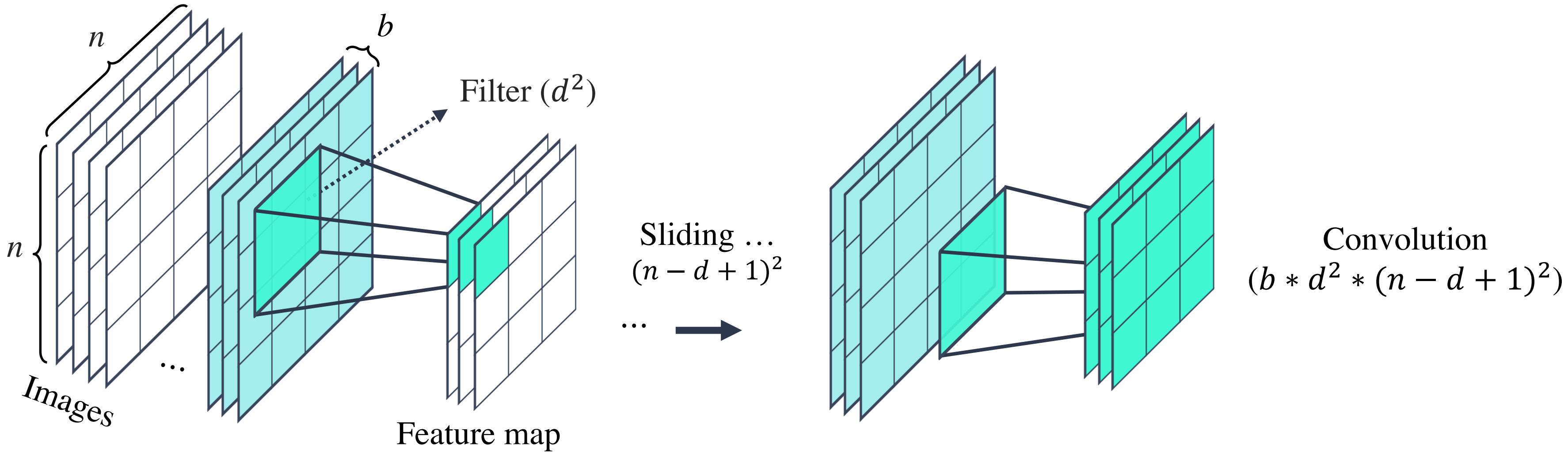


<Computational cost of a MF model>

Analysis on the Training of a DNN Model

The DNN model training is *computation-intensive*

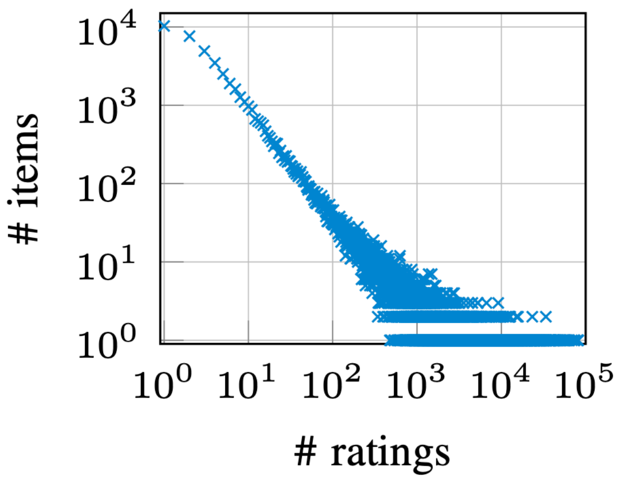
- Memory and computation costs of a convolution layer
  - Computation cost $(b * d^2 * (n - d + 1)^2)$ **>** Memory cost $(b * n^2 + d^2)$
- The room for the performance improvement by the quantization is sufficient



<Computational cost of a convolution layer>

Unique Feature of Datasets in RS

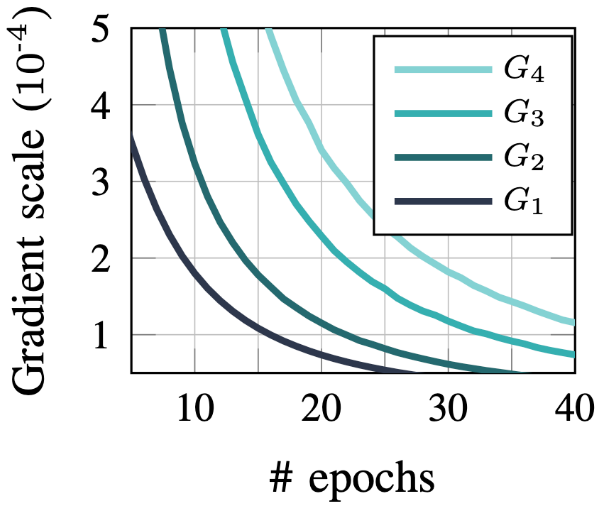
- **The datasets used in RS have a *power-law distribution***
  - A majority of users/items have a small number of ratings
  - A small number of users/items have a very large number of ratings
- **The MF model training with SGD**
  - The update frequency for each latent vector depends on the number of ratings
    - The latent vectors of a few users/items with many ratings are updated *frequently*
    - The latent vectors of many users/items with few ratings are updated *infrequently*



Quantization Error of Each Latent Vector

- **The scale of the gradient for each latent vector *varies*, depending on the number of ratinings**
  - The scale of gradient tends to decrease as the model is trained more

- **The quantization error of each user/item is *different* from each other, depending on the number of ratings (*Observation 2*)**



Limitations of Existing Precision Switching

- **Considering the quantization error for the entire model (i.e., not considering the difference among users/items)**
  - The loss of accuracy
    - For users with many ratings, the precision switching is likely to be applied *too late*
  - The reduced training performance
    - For users with few ratings, the precision switching is likely to be applied *unnecessarily quickly*

Overview of The Proposed Framework (MASCOT)

☐ Quantization strategy for memory access (*m*-quantization)

- Storing and managing the parameters of MF models in low precision

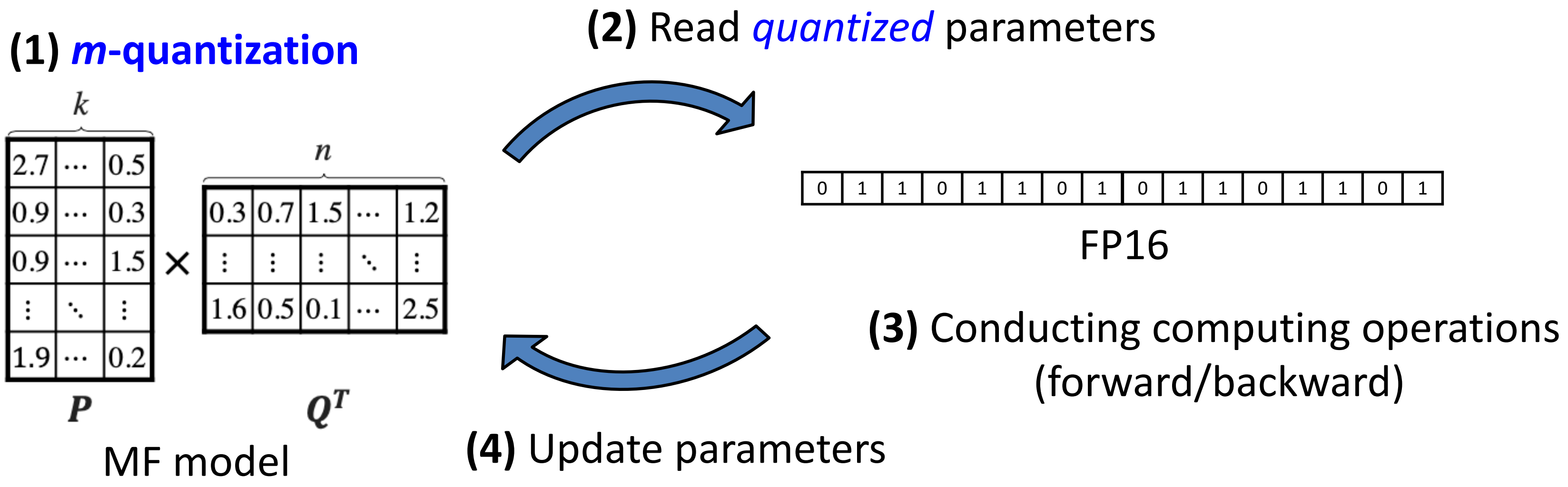
☐ Group-based precision switching strategy (*g*-switching)

- Grouping users/items having a similar number of ratings
- Applying precision switching in a group-wise manner

Strategy 1: *m*-quantization

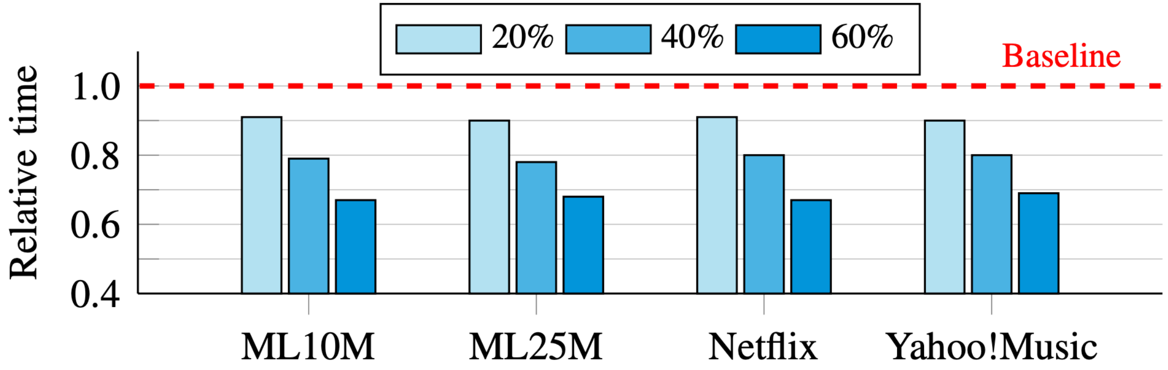
☐ Storing and managing the parameters of a MF model in low precision

- For *improving the memory access operations* in the training of MF models



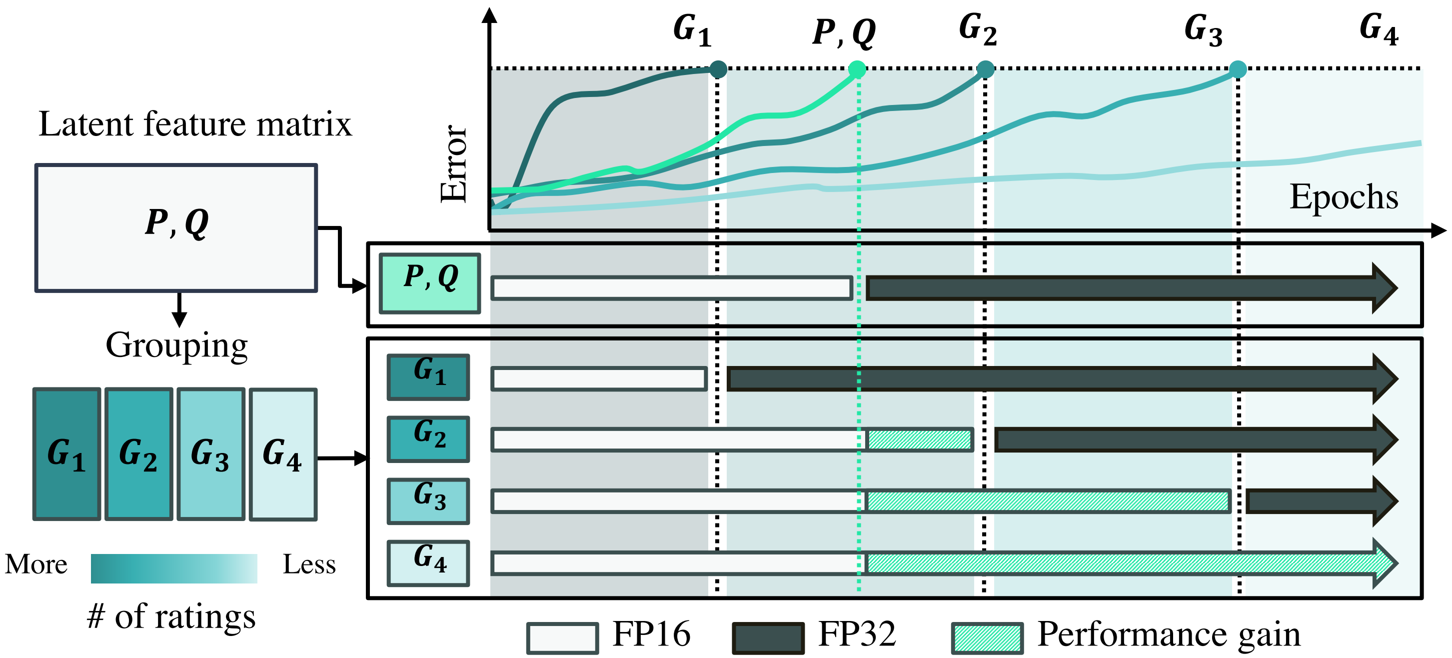
Strategy 1: *m*-quantization

- **The potential of the *m*-quantization strategy for improving the training performance of MF models**



Strategy 2: *g*-switching

- ***Grouping*** **users/items having a** ***similar*** **number of ratings and** ***applying*** **precision switching in a** ***group-wise*** **manner**
  - For *considering the difference among users/items* in RS datasets

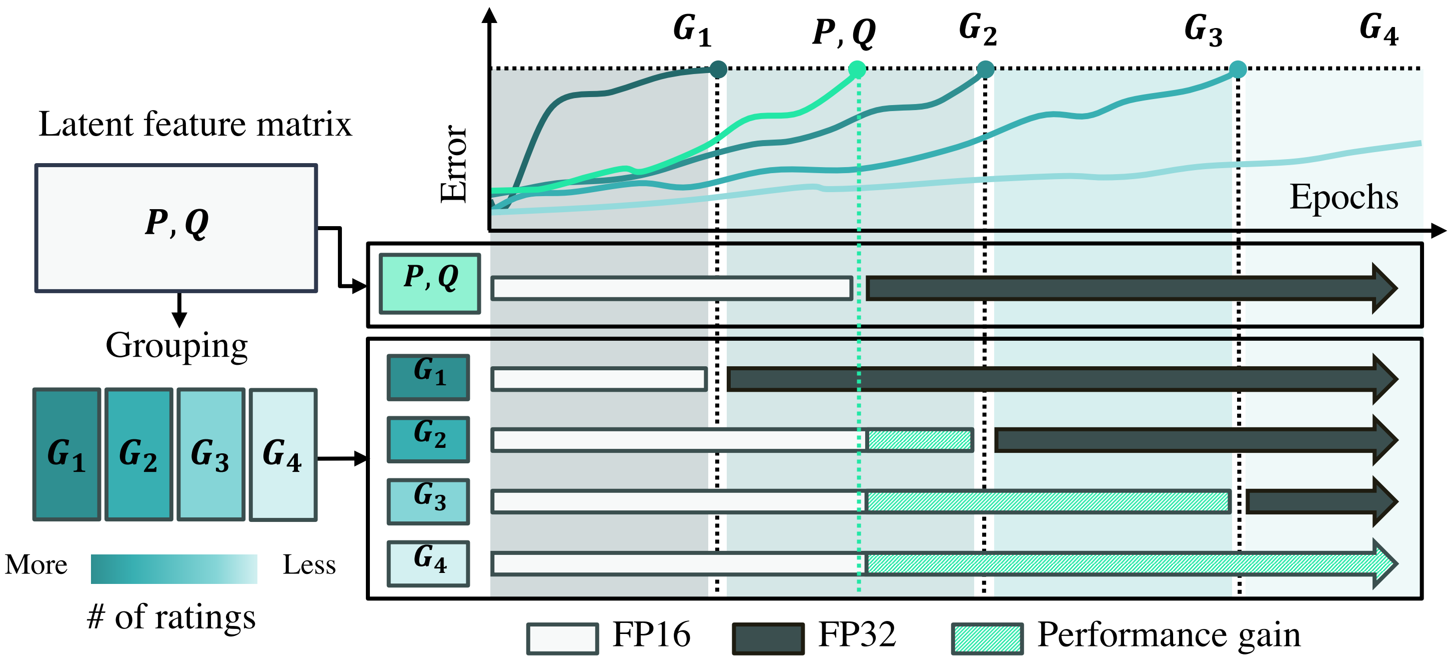


<The performance improvement by the g-switching of MASCOT>

Strategy 2: *g*-switching

☐ Performance improvemnet by the *g*-switching

- Applying precision switching *only to the groups highly likely to cause the model error*



<The performance improvement by the g-switching of MASCOT>

Algorithm

☐ Training process of MASCOT

1) Grouping users/items vectors
2) For each user/item,
   - ☐ Read the embedding vector;
   - ☐ Compute the loss and gradients;
   - ☐ Update the embedding vector;
   - ☐ Compute the quantization error of each group;
3) Determining to apply precision switching to each user/item group

Algorithm 1 Training of MASCOT

Require: $R \in \mathbb{R}^{m \times n}$, $P \in \mathbb{R}^{m \times k}, Q \in \mathbb{R}^{n \times k}$, # of groups g, error estimate period π, sample ratio γ, error threshold θ, learning rate η

$R, P, Q \leftarrow$ grouping(R, P, Q, g)
Initialize P, Q with half precision
for $t = 1, ..., T$ **do**
  for each rating $r_{u,i}$ **do**
    $\nabla p_u \leftarrow e_{u,i} \cdot q_i - \lambda_P \cdot p_u$
    $\nabla q_i \leftarrow e_{u,i} \cdot p_u - \lambda_Q \cdot q_i$
    $p_u \leftarrow p_u + \eta \cdot \nabla p_u$
    $q_i \leftarrow q_i + \eta \cdot \nabla q_i$
    if $S \sim B(\gamma)$ **then**
      $S_j^U.push(\nabla p_u), S_j^I.push(\nabla q_i)$
    end if
    update_latent_matrix(P, Q, p_u, q_i)
  end for
  if $t \pmod{\pi} == 0$ **then**
    for $j = 1, ...g$ **do**
      $\epsilon^U \leftarrow q\text{-}error(S_j^U), \epsilon^I \leftarrow q\text{-}error(S_j^I)$
      precision_switching$(P, Q, \epsilon^U, \epsilon^I, \theta)$
    end for
  end if
  $\forall j \in \{1, ..., g\}, S_j^U.flush(\cdot), S_j^I.flush(\cdot)$
end for
Return P, Q

Experimental Setup

Models & Datasets

- MF models with varying the dimensionality of laten space (64, 128)
- Statistics of datasets

Datasets	# of users	# of items	# of ratings	Sparsity
ML10M	69,878	10,677	10,000,035	98.66%
ML25M	162,541	59,047	24,997,208	99.74%
Netflix	480,189	17,770	100,480,507	98.82%
Yahoo!Musics	1,000,990	624,961	256,804,235	99.96%

Competing algorithms

- MPT (ICML'18)
- MuPPET (ICML'20)
- AFP (CVPR'20)
- Two baselines (FP32, FP16)

Questions to Answer

☐ Q1: Training performance

- Does MASCOT improve the training performance of MF models more than existing quantization methods?

☐ Q2: Model quality

- Does MASCOT provide the errors of MF models lower than existing quantization methods?

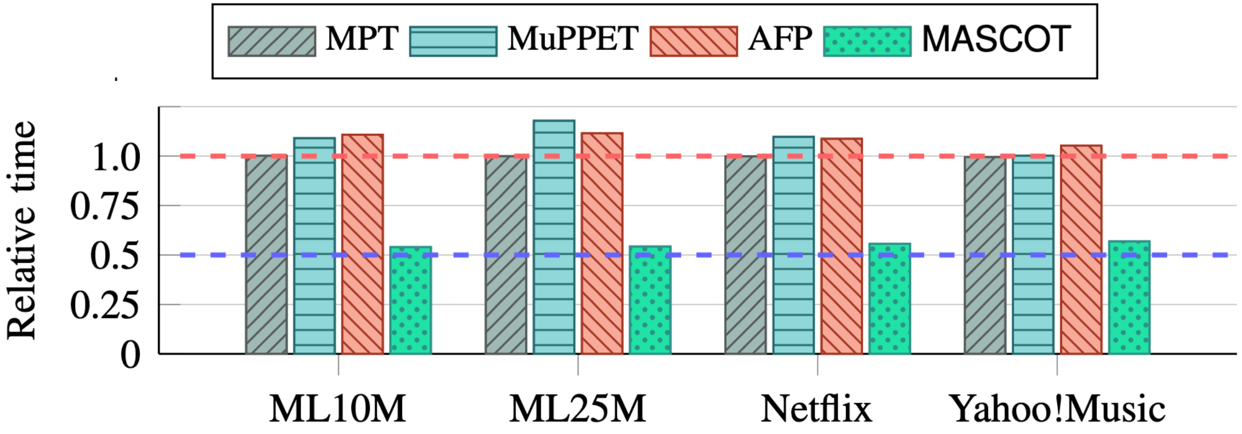
☐ Q3: Effectiveness of each strategy

- How effective are the strategies of MASCOT in improving the MF model training?

Q1. Training Performance

- MASCOT improves the training performance of MF models most
  - About *45% performance improvement* on average

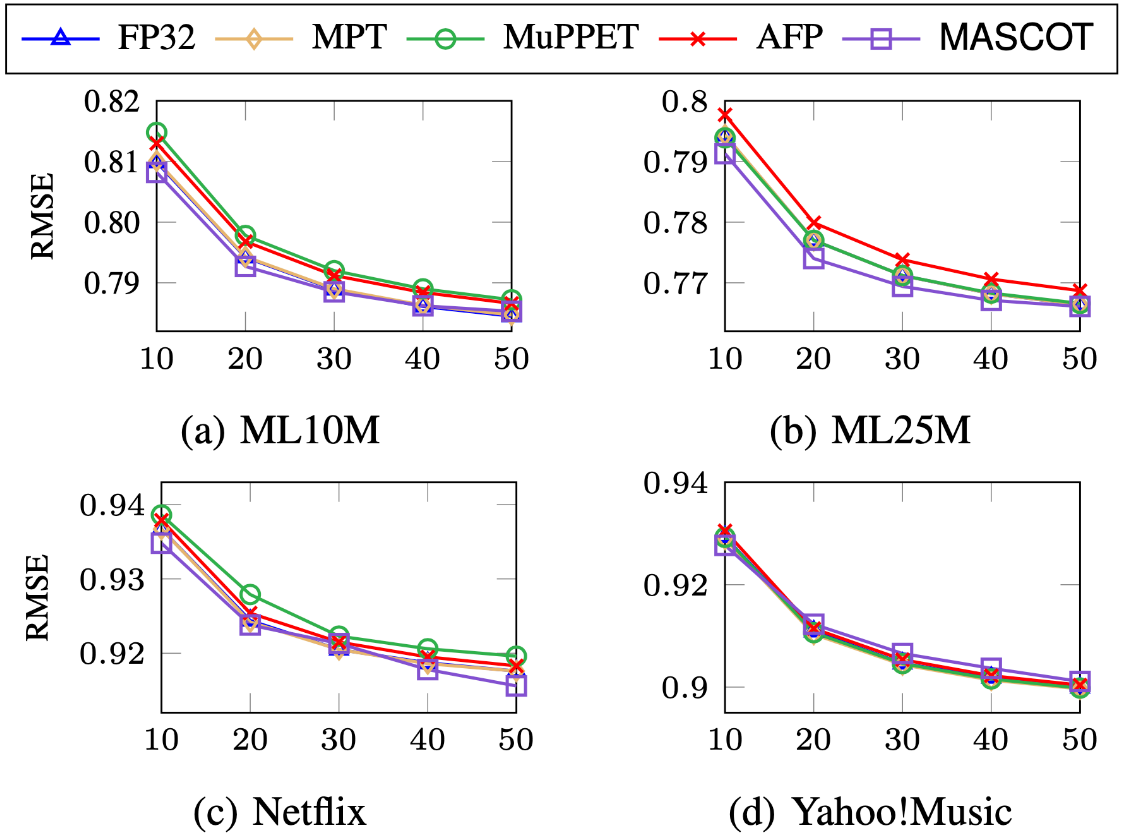
- Existing SOTA quantization methods ***degrade*** the training performance of MF models



Q2. Model Quality

☐ MASCOT achieves low model errors comparable to that of FP32

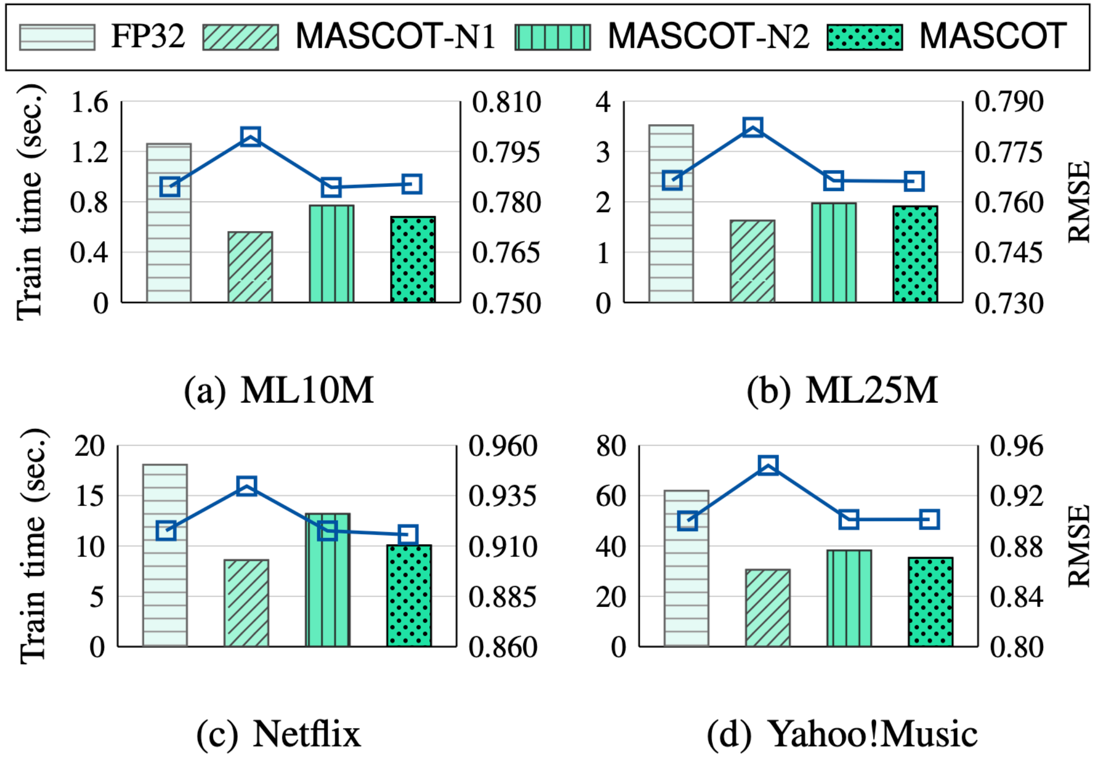
- The g-switching applies the precision switching only to the groups that are highly likely to incur significant model errors



Q3. Ablation Study: Effectiveness of Strategies

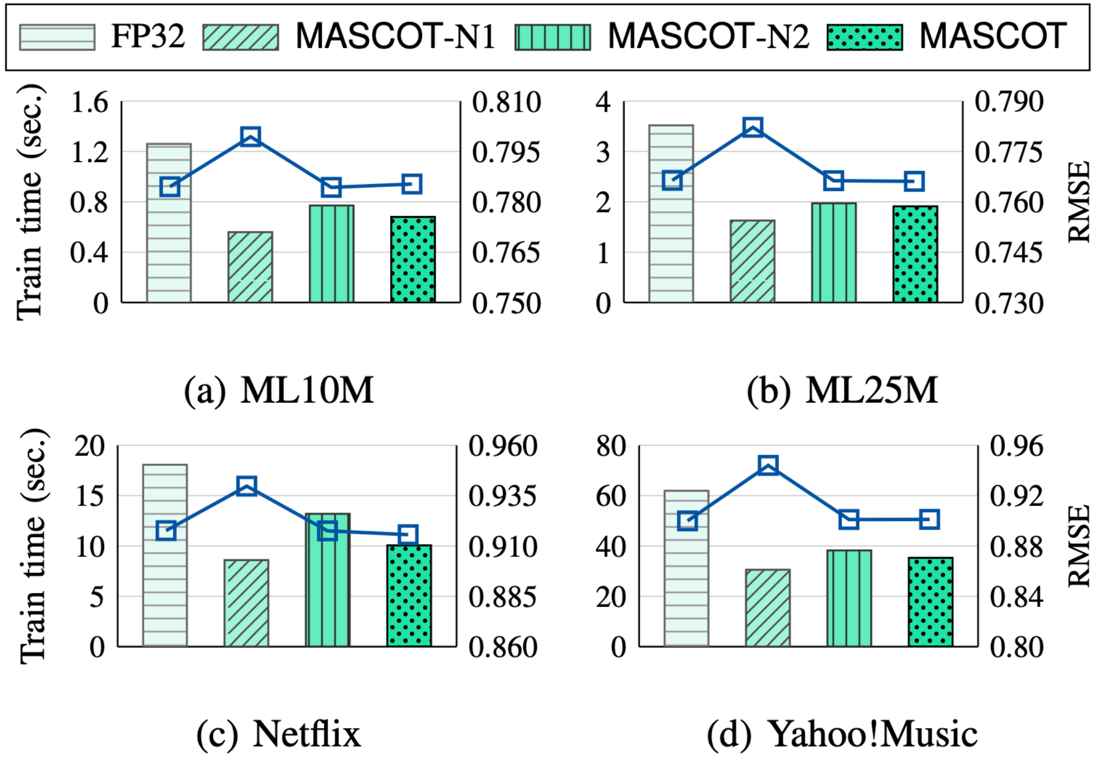
☐ Three versions of MASCOT

- MASCOT-N1 is with only *m*-quantization
- MASCOT-N2 is with *m*-quantization and the existing precision switching method
- MASCOT is with both *m*-quantization and *g*-switching



Q3. Ablation Study: Effectiveness of Strategies

- **Both strategies of MASCOT are quite effective in the MF model training.**
  - MASCOT-N1 shows the best improvement in reducing training time (*m-quantization*)
  - MASCOT outperforms MASCOT-N2 in terms of both training performance and model quality (*g-switching*)



Conclusions

- **Discovering that existing SOTA quantization techniques are rarely effective in the training of MF models**
- **Identifying two unique features of the training of MF models**
  - (i) The training of MF models is more memory-intensive than that of DNN models
  - (ii) The quantization error of each user/item differs, depending on the number of
- **Proposing a quantization framework for efficient training of MF models**
  - Employing two strategies to address the unique features of the training MF models
- **Comprehensive evaluation verifying the effectiveness of MASCOT in the training MF models**
  - Improving the training performance by about 45% on average (almost ideal)

Thank You !

Email: koyunyong@hanyang.ac.kr