Matrix Factorization with Comparison Data

Mayeul Cassier
Supervised by Dr. Suryanarayana Sankagiri
Indy Lab, EPFL
June 6, 2025

Contents

1 Introduction

Matrix factorization (MF) is a foundational technique in modern machine learning and data science, particularly within the domain of recommender systems. It allows for modeling user-item interactions by decomposing a large, sparse utility matrix into low-dimensional latent representations. The central assumption behind matrix factorization is that user preferences and item characteristics lie on a lower-dimensional manifold — i.e., the interaction matrix is approximately low-rank. This hypothesis makes the problem tractable and allows for effective generalization even when a large portion of the data is missing[1].

In traditional MF, models are trained using explicit feedback such as ratings. However, in many real-world scenarios — such as clicks, purchases, views, or any form of implicit feedback — it is more natural and reliable to model user behavior in terms of *comparisons*.

In this project, we extend the classical MF framework to a comparison-based setting. Instead of relying on absolute utility scores, the model is trained from triplets (u, i, j) indicating that user u prefers item i over item j.

We aim to understand how matrix factorization behaves in this comparative framework, and to study the influence of key structural and algorithmic parameters on its optimization and predictive performance. These parameters include:

- The **scaling factor** s, which controls the sharpness of user preferences and reflects the noise level in the feedback.
- The **sparsity level** p, which determines how many triplet comparisons are observed for each user.
- The **comparison redundancy** k, which controls how many times a given pairwise preference is repeated (introducing statistical stability).
- The **sampling strategy**, which selects which triplets are observed — ranging from uniform sampling to more targeted approaches (e.g., popular items, close items, etc.).

To address these questions, we design controlled synthetic experiments based on known ground-truth embeddings. We simulate user preferences, generate pairwise comparison data, and study how stochastic gradient descent performs in recovering the underlying structure. Throughout the report, we evaluate both reconstruction quality and alignment with ground-truth preferences using metrics such as reconstruction error, Pearson and Spearman correlation, and classification accuracy on held-out comparisons.

This report is structured as follows. In Section 1, we introduce the motivation for matrix factorization from pairwise comparison data. Section 2 presents the theoretical foundation of our approach, including the data generation pipeline, loss functions, optimization methods, and evaluation metrics.

We then turn to experimental results. Section 3 verifies the consistency of our synthetic data generation setup. Section 4 analyzes the influence of the latent dimension d on reconstruction quality. In Section 5, we study how the sharpness parameter s affects learning performance, and how it interacts with regularization. Next, in Section 6, we investigate the role of comparison redundancy (k), and in Section 7, we analyze the joint impact of sparsity (p) and repetition (k). Section 8 explores trade-offs between data quantity and preference contrast.

In Section 9, we evaluate various triplet sampling strategies and show how their inductive biases affect learning in low-data regimes.

Finally, Section 10 summarizes our key findings and highlights promising directions for future work.

2 Theory

2.1 Matrix Factorization and Optimization Framework

Matrix factorization (MF) is a widely used technique in machine learning, particularly in recommender systems. Given a partially observed user-item interaction matrix X^*, the goal is to approximate it via low-dimensional latent factorization:

$$X^* \approx \hat{X} = UV^\top \tag{1}$$

where:

- $U \in \mathbb{R}^{n \times d}$ is the matrix of **user embeddings**,
- $V \in \mathbb{R}^{m \times d}$ is the matrix of **item embeddings**,
- d is the **latent dimension** capturing user and item features.

In our study, we focus on **comparative preference learning**, where instead of explicit ratings, users are presented with pairs of items and choose the preferred one. This setting follows the **Bradley-Terry-Luce (BTL) preference model**, in which the probability of a user u preferring item i over item j is given by:

$$P(i \succ j|u) = \sigma\left(s \cdot (X^*_{u,i} - X^*_{u,j})\right) \tag{2}$$

where:

- $\sigma(x) = \frac{1}{1+e^{-x}}$ is the **sigmoid function**,
- s is a **scaling factor** controlling confidence in user preferences.

This probabilistic formulation allows us to train the model using observed pairwise comparisons instead of explicit numerical ratings.

2.2 Data Generation Process (U^*, V^*)

Since real-world data is often expensive or unavailable for controlled analysis, we generate **synthetic data**. The true user and item embeddings are denoted as:

$$U^* \in \mathbb{R}^{n \times d}, \quad V^* \in \mathbb{R}^{m \times d} \tag{3}$$

These embeddings define the ground-truth preference structure, and the goal of the optimization process is to recover a factorization U, V that approximates U^*, V^*.

2.2.1 Generating User and Item Embeddings

The method chosen to generate embeddings was to construct a low-rank matrix $X^* = U^* S V^{*T}$ with orthogonal factors. Specifically, matrices $U^* \in \mathbb{R}^{n \times n}$ and $V^* \in \mathbb{R}^{m \times m}$ are sampled as random orthogonal matrices, and S is a diagonal matrix with the first d singular values set to $\frac{1}{\sqrt{d}}$ and the rest to zero.

This ensures that X^* has rank d and stable variance across latent dimensions. The final matrix is normalized by $\sqrt{n \cdot m}/2$, and is used as a ground-truth preference matrix X^*. Preference labels are then derived from X^* using the BTL model.

As a recap:

$$U^* \in \mathbb{R}^{n \times n}, \quad V^* \in \mathbb{R}^{m \times m}$$

and

$$X^* = U^* S (V^*)^\top \cdot \frac{\sqrt{nm}}{2}$$

where $S \in \mathbb{R}^{n \times m}$ is a diagonal matrix, typically defined as:

$$S = \text{diag}\left(\underbrace{\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}, \ldots, \frac{1}{\sqrt{d}}}_{d \text{ times}}, 0, \ldots, 0 \right) \in \mathbb{R}^{n \times m}$$

2.2.2 Selecting Item Pairs (i, j) for Comparison

A user u is chosen uniformly at random, and then a pair of items i, j is sampled uniformly such that $i \neq j$. Different methods for sampling item pairs will be described later.

2.2.3 Generating Pairwise Preference Labels

For each sampled triplet (u, i, j) — where user u and items $i \neq j$ are drawn uniformly at random — we generate a preference label $y_{u,i,j}$ in one of two ways:

1. **Deterministic label generation**: we assign a label of 1 if the probability that u prefers i over j exceeds 0.5:
$$y_{u,i,j} = \mathbb{1}_{[P(i \succ j | u) > 0.5]} \tag{4}$$
2. **Probabilistic label generation**: we sample the label from a Bernoulli distribution, where the success probability is the user's preference:
$$y_{u,i,j} \sim \text{Bernoulli}(P(i \succ j \mid u)) \tag{5}$$

The probability $P(i \succ j \mid u)$ is defined using the Bradley–Terry–Luce (BTL) model, and reflects how much user u prefers item i to item j based on the latent inner products. In our experiments, we use **deterministic labels** (option 1) to compute the Ground Truth Accuracy and generate reference signals, while **probabilistic labels** (option 2) are used to simulate noisy comparison data in realistic training scenarios. In practice, we generate such preference triplets on $p\%$ of the total possible (u, i, j) combinations — representing the available comparison data.

Given this, the total number of triplets we observe is approximately:

$$\frac{mnp}{2}$$

since each comparison involves a pair of items.

2.2.4 Averaging over k Repeated Observations

To obtain a more accurate estimate of the true preference probability, we assume that each triplet (u, i, j) can be observed multiple times independently. More precisely, we simulate k repeated judgments of the same comparison. These repetitions reduce the sampling noise induced by the probabilistic label generation.

Let $y^1_{u,i,j}, \ldots, y^k_{u,i,j}$ be k independent binary labels for the same triplet. We define the aggregated label as:

$$\hat{y}_{u,i,j} = \frac{1}{k} \sum_{\ell=1}^{k} y_{u,i,j}^{\ell} \tag{6}$$

This averaged quantity approximates the true value of $P(i \succ j \mid u)$, and serves as the ground-truth signal in our evaluation framework. The larger the value of k, the lower the variance in this estimate.

2.3 Loss Functions and Optimization

The goal of training is to learn embeddings U, V such that they minimize a loss function quantifying the discrepancy between model predictions and observed data.

2.3.1 Binary Cross-Entropy (BCE) Loss

Because the preference labels $y_{u,i,j}$ are probabilistic — either due to label noise or repeated stochastic sampling — we adopt a probabilistic loss function. Specifically, we use the **binary cross-entropy (BCE)** loss, which is well-suited to comparing predicted probabilities with observed Bernoulli labels.

Let $P(i \succ j \mid u)$ denote the model's predicted probability that user u prefers item i over item j. Under the BTL model, this probability is a function of the user and item embeddings:

$$P(i \succ j \mid u) = \sigma\left(\langle u, i\rangle - \langle u, j\rangle\right) = \sigma\left(U_u^\top V_i - U_u^\top V_j\right)$$

where $\sigma(x)$ is the sigmoid function.

The binary cross-entropy loss over the dataset D is then defined as:

$$\mathcal{L}_{\text{BCE}} = -\sum_{(u,i,j)\in D} \left[y_{u,i,j} \log P(i \succ j \mid u) + (1 - y_{u,i,j}) \log(1 - P(i \succ j \mid u))\right] \tag{7}$$

This loss is compatible both with binary labels and with **fractional labels** in $[0, 1]$, such as when $y_{u,i,j}$ is computed as the empirical average over k independent repetitions:

$$\hat{y}_{u,i,j} = \frac{1}{k} \sum_{\ell=1}^{k} y_{u,i,j}^{\ell}$$

In that case, the BCE loss acts as a natural maximum likelihood objective for modeling observed frequencies with predicted probabilities. It thus remains valid and differentiable even when the targets are fractional, and allows us to better approximate the underlying preference structure.

2.3.2 Regularization with Weight Decay

To prevent overfitting, we introduce **L2 regularization** (weight decay):

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + w_d(\|U\|_F^2 + \|V\|_F^2) \tag{8}$$

where w_d is a hyperparameter controlling the regularization strength.

2.3.3 Optimization with Adam

The parameters of the model — namely the user and item latent embeddings $U \in \mathbb{R}^{n \times d}$ and $V \in \mathbb{R}^{m \times d}$ — are jointly optimized using the **Adam optimizer** [2], a variant of stochastic gradient descent with adaptive moment estimation.

Let $\theta = (U, V)$ denote the collection of all parameters to be learned. The Adam update rule for each parameter at time step t is given by:

$$\theta_{t+1} = \theta_t - l_r \cdot \frac{m_t}{\sqrt{v_t} + \epsilon} \tag{9}$$

where:

- l_r is the learning rate,
- m_t is the exponential moving average of the gradients (first moment),
- v_t is the exponential moving average of the squared gradients (second moment),
- ϵ is a small constant added for numerical stability.

2.4 Evaluation Methods

To evaluate the model's performance, several metrics have been used, each capturing a different aspect of the prediction quality: score-level approximation, pairwise preference consistency, and structural alignment.

2.4.1 Reconstruction Error

$$\epsilon = \frac{\left\| X^* - \frac{UV^\top}{s} \right\|_F}{\|X^*\|_F} \tag{10}$$

Interpretation: this metric quantifies the relative error between the ground-truth matrix X^* and the reconstructed matrix $UV^\top$, after rescaling. The scaling factor $\frac{1}{s}$ is motivated by the way we inject noise in the Bradley–Terry–Luce (BTL) model: when generating comparisons, we apply a scaling factor s to the score differences:

$$P(i \succ j \mid u) = \sigma\left(s(X^*_{u,i} - X^*_{u,j})\right)$$

As a result, the model learns a transformed version of the true matrix: namely, a version that approximates sX^* rather than X^* itself. To fairly evaluate the reconstruction quality, we therefore rescale the learned approximation $UV^\top$ by $\frac{1}{s}$ before comparing it to X^*.

The Frobenius norm $\|\cdot\|_F$ measures the element-wise squared error across all user-item pairs, and the normalization by $\|X^*\|_F$ ensures that the metric is scale-invariant. Lower values of ϵ indicate a better global alignment between predicted scores and ground truth.

This metric doesn't depend on a test set.

2.4.2 Accuracy

$$\text{Accuracy} = \frac{1}{|\mathcal{T}|} \sum_{(u,i,j) \in \mathcal{T}} \mathbb{1}_{\left\{(\hat{X}_{u,i} - \hat{X}_{u,j}) \cdot (X^*_{u,i} - X^*_{u,j}) > 0\right\}}$$

Interpretation: Accuracy measures the proportion of test triplets for which the predicted preference direction (based on the reconstructed scores $\hat{X} = UV^\top$) agrees with the direction in the ground-truth matrix X^*. It evaluates the consistency of local ordering. Since this metric is evaluated on a separate test set $\mathcal{T}$, it depends directly on the structure and size of that set.

2.4.3 Ground Truth Accuracy

$$\text{GT Accuracy} = \frac{1}{|\mathcal{T}|} \sum_{(u,i,j)\in\mathcal{T}} \mathbb{1}_{\left\{\left[(X^*_{u,i} - X^*_{u,j}) > 0\right] = y_{u,i,j}\right\}}$$

Interpretation: Ground Truth Accuracy measures how consistent the observed labels $y_{u,i,j}$ are with the ranking induced by the true score matrix X^*. In effect, it quantifies the upper bound of what any model can hope to achieve in terms of accuracy. It is particularly useful to characterize the noisiness of the training data. Note that this quantity also depends on the test set $\mathcal{T}$, but doesn't depend on s.

2.4.4 Pearson Correlation

$$\text{Pearson}(x, y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}}$$

Interpretation: Pearson correlation measures the degree of linear relationship between two vectors. A value close to 1 indicates strong alignment along a straight line, regardless of the magnitude of the predictions. It is widely used to compare the structure of predicted score vectors with ground-truth scores. However, it is **scale-invariant** and thus does not penalize global miscalibration. Note that Pearson correlation can also depend on the scaling factor s, since overly small or large s can distort the geometry of $UV^\top$.

2.4.5 Spearman Correlation

$$\text{Spearman}(x, y) = \text{Pearson}(\text{rank}(x), \text{rank}(y))$$

If there are no tied ranks, it simplifies to:

$$\text{Spearman}(x, y) = 1 - \frac{6\sum_i d_i^2}{n(n^2 - 1)} \quad \text{where } d_i = \text{rank}(x_i) - \text{rank}(y_i)$$

Interpretation: Spearman correlation evaluates how well the relative order of predicted scores matches the ground-truth order, regardless of the actual values. Unlike Pearson, it is robust to non-linear rescalings and thus particularly informative when only rankings matter. It is especially useful in preference learning settings, where the absolute scores are often meaningless, but the order is critical.

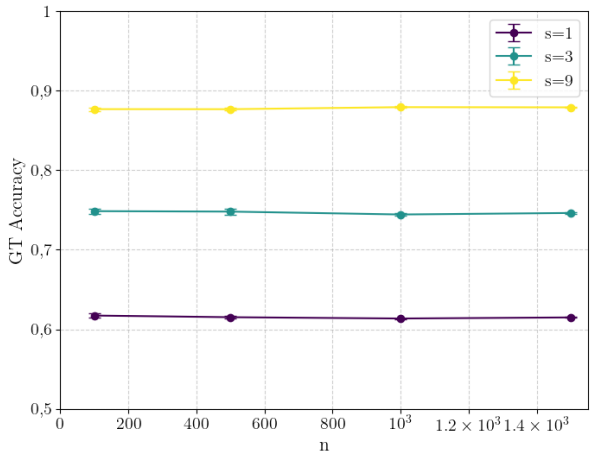
2.5 Variables

Parameter	Description	Range	Default Value
n	Number of users	$[100,\ 1500]$	1000
m	Number of items	$[100,\ 1500]$	1000
d	Latent embedding dimension	$[1,\ 10]$	2
p	Percentage of observed triplets	$[10^{-3},\ 1]$	0.2
lr	Learning rate for Adam optimizer	$[10^{-4},\ 10^{-2}]$	10^{-3}
w_d	Weight decay for regularization in Adam	$[10^{-6},\ 10^{-3}]$	10^{-5}
k	Number of responses per triplet (u, i, j)	$[1, 50]$	1
s	Scaling factor in the sigmoid preference model	$[10^{-2},\ 10^{3}]$	8

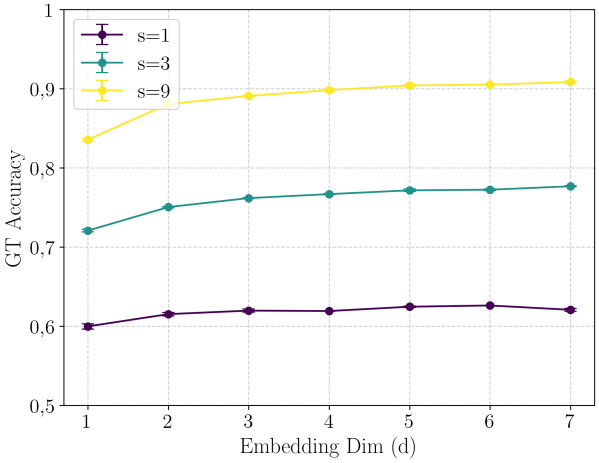
Table 1: Main hyperparameters and their usage in experiments.

3 Data Generation Consistency Analysis

To validate the reliability of our synthetic data generation process, we analyze how the ground-truth accuracy behaves under different structural parameters such as the number of users/items, latent dimension, and scaling factor.



(a) Ground Truth Accuracy as a function of n for different values of s



(b) Ground Truth Accuracy as a function of d for different values of s

Figure 1: Ground Truth Accuracy under different structural parameters. Fixed: $p = 0.5$, $k = 1$, $lr = 10^{-3}$, $wd = 10^{-5}$, $reps = 3$.

In order to validate the consistency of our data generation process, we verify that the ground-truth accuracy remains stable across different values of m, n, d, k, and p. As shown in Figure 1a, the accuracy is approximately constant when varying n, for fixed values of other parameters.

Furthermore, in Figure 1b, we examine the behavior of ground-truth accuracy as a function of the latent dimension d for various values of the scaling factor s. We observe a monotonic increase in accuracy with respect to d, and a saturation effect as s increases — which is expected, since higher s leads to sharper and more deterministic preference signals.

Validation of GT Accuracy Stability. To ensure that the ground-truth accuracy we rely on throughout the paper is itself stable and reliable, we conduct a complementary analysis varying p and k. The results are provided in subsection A.1, Figure 19, and justify the robustness of our metric under changing data regimes.

4 Analysis of the Effect of Embedding Dimension d

To better understand the role of latent dimension size in matrix factorization, we analyze how the accuracy varies with respect to both the embedding dimension d and the observed data sparsity p.

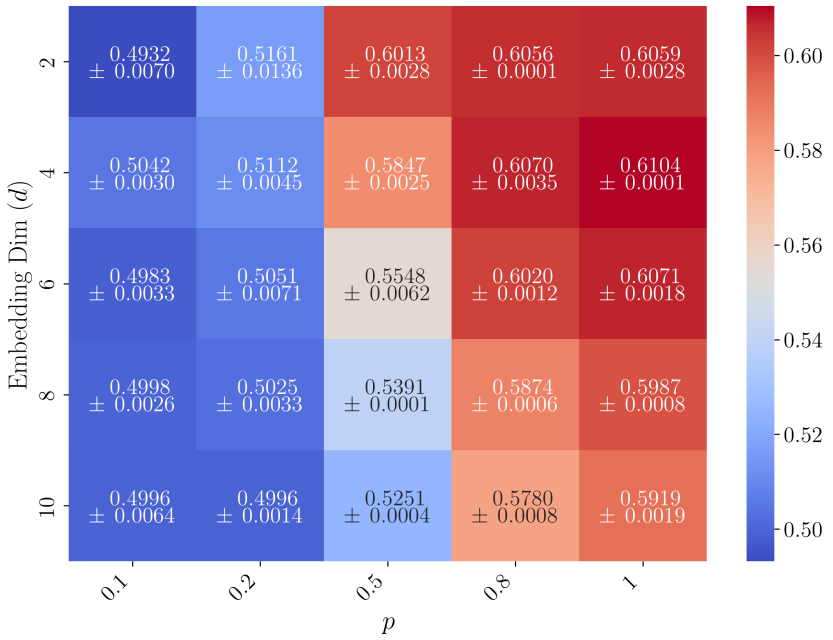


Figure 2: Heatmap of the validation accuracy as a function of embedding dimension d and data sparsity p. Higher values correspond to better local ordering accuracy. Fixed hyperparameters: $n = m = 1000$, $s = 1$, $lr = 10^{-3}$, $wd = 10^{-5}$, $k = 1$, $reps = 5$, $epochs = 30$.

Figure 2 shows a heatmap of the validation accuracy achieved across different combinations of p (fraction of observed preferences) and d (embedding dimension). Each cell in the map corresponds to the average accuracy over 5 training repetitions using the same configuration.

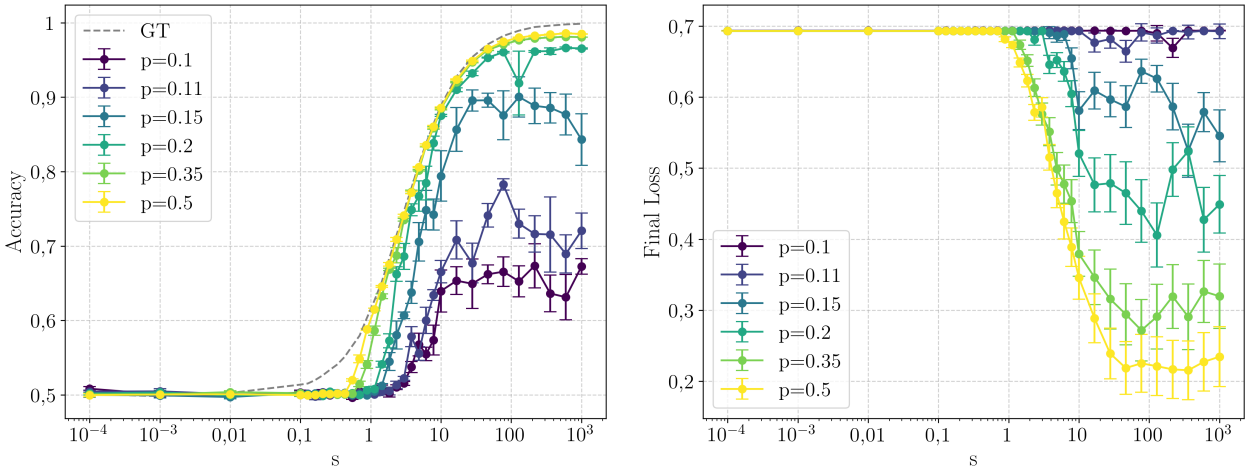
We observe several trends:

- For all values of p, decreasing d tends to improve accuracy up to a certain point. This reflects the fact that lower embeddings allow the model to capture the underlying structure of the ratings.
- The performance gain saturates quickly when p is small. In extremely sparse settings (e.g., $p = 0.1$), decreaing d beyond 4 or 6 does not help much — likely because the model lacks sufficient supervision to learn meaningful high-dimensional representations.
- Conversely, for dense settings (e.g., $p = 0.8$ or $p = 1.0$), the model fully exploits the expressivity of larger embeddings, achieving nearly optimal accuracy.

This analysis highlights an important interaction between model capacity (controlled by d) and data availability (through p): overparameterized models are only beneficial when enough information is available to train them.

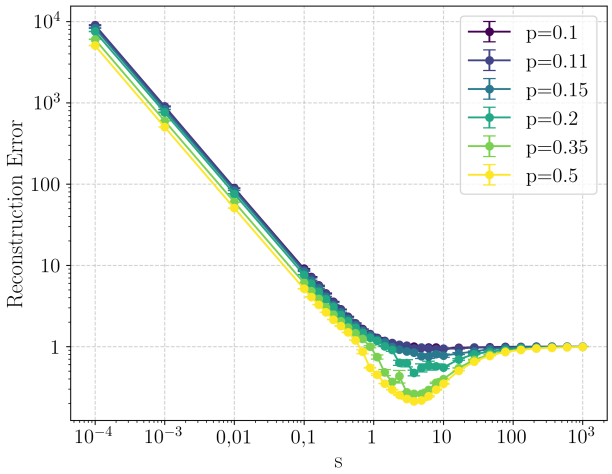
5 Analysis of the effect of the parameter s

$X^*[u, i]$ and $X^*[u, j]$ are the latent scores of user u for movies i and j. When generating preferences, the scaling factor s amplifies or dampens the probability of correctly classifying the preferred movie. If s is large, preferences become more decisive: even a small difference between $X^*[u, i]$ and $X^*[u, j]$ leads to a strong preference. If s is small, the model becomes more uncertain because of the data: the probability to classify a preference in the data remains close to 0.5 unless the difference is significant.



(a) Accuracy (GT and validation)

(b) Final validation loss



(c) Reconstruction error

Figure 3: Effect of the scale parameter s on accuracy, final loss, and reconstruction error. Fixed parameters: $n = m = 1000$, $d = 2$, $k = 1$, $l_r = 10^{-3}$, $w_d = 5 \cdot 10^{-6}$, $reps = 5$. Logarithmic scale on x-axis.

As the scaling factor s increases, the ground-truth accuracy also tends to increase. This is because the preference labels become more deterministic: when s is large, even a small difference in latent scores between two items results in a strong probability close to 0 or 1. This makes it easier for a model (or even just the ground truth embeddings) to predict the correct preference.

Consequently, the model accuracy typically improves as well, since it becomes easier to distinguish between the preferred and non-preferred items. The task becomes more "separable." The underlying preference labels become easier to predict because the signal-to-noise ratio increases: the probability distribution over labels becomes concentrated near 0 or 1. In practical terms, this means that the model faces a simpler classification task, as most pairwise comparisons become unambiguous. The latent score differences are amplified, allowing the decision boundary to more clearly distinguish between preferred and non-preferred items. This is clearly visible in Figure 3, subplot (a), where both ground truth and model accuracy increase with the scale parameter s.

However, beyond a certain point, increasing s further does not yield significant improvements in accuracy. This is because the underlying preference data is already well-separated—making the task too easy. In such cases, both the ground truth and the model already predict prefer-

ences correctly in most cases, and additional sharpening of the signal brings little to no gain. This results in a performance plateau in both ground-truth accuracy and model accuracy.

A similar phenomenon can be observed with the final loss, shown in Figure 3, subplot (b): as s increases, the loss initially decreases because the model can more easily match the deterministic labels. However, once the preferences are sufficiently well-separated, the loss also reaches a plateau.

Interestingly, the trend of the final loss behaves inversely to that of the accuracy — which is expected. As accuracy increases, the model makes fewer mistakes, and therefore the loss (which penalizes incorrect predictions or uncertain confidence) naturally decreases. When accuracy saturates, the loss does too, stabilizing at a low value.

5.1 Reconstruction Error Behavior Analysis

The reconstruction error, shown in Figure 3, subplot (c), exhibits a similar trend to the final loss but with an important nuance: we observe a minimum around $s \approx 1$. As the scaling factor increases from low values, the model is better able to identify preference signals, which improves the fit and reduces the reconstruction error.

Very low values of s show an important decrease in the reconstruction error. This decreasing trend in reconstruction error can be misleading. While it seems to indicate better alignment between the reconstructed matrix and the ground-truth, it can be artificially induced by tuning the `weight decay`. In fact, as shown in Figure 4, we can always force a low reconstruction error by increasing the `weight decay` for small s values. In this regime, the optimizer reduces the norm of $UV^\top$ to nearly zero, which drives the error expression toward $\|X^*\|_F$ and yields a normalized reconstruction error of nearly 1.

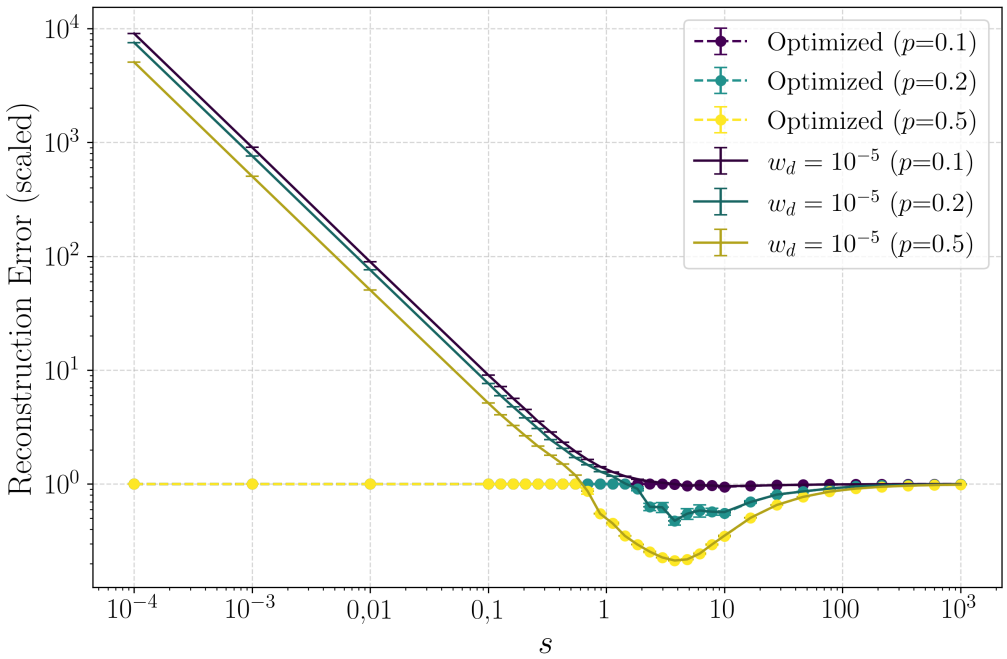


Figure 4: **Reconstruction Error vs Scaling Factor s with Weight Decay Sweeps.** This plot compares optimized reconstruction error to a fixed `weight decay` baseline (10^{-5}). Larger weight decay values at small s can artificially shrink the norm of $UV^\top$, leading to high apparent error. **Hyperparameters:** $n = m = 1000$, $d = 2$, $k = 1$, $l_r = 10^{-3}$, epochs $= 30$, $reps = 5$, $p \in \{0.1, 0.2, 0.5\}$

This explains the result in Figure 5, where the optimal `weight decay` becomes very large for small s values. We are not learning anything meaningful — just minimizing scale by shrinking all embeddings. However, as s increases, we gain more confidence in preference comparisons, enabling more accurate learning. A real minimum appears around $s = 5$, with both reconstruction quality and generalization improving.

This minimum is sensitive to the sampling rate p : as p increase, the minimum becomes sharper and shifts left. This is intuitive: with more data, confidence grows faster, and the model needs less scaling to reach optimal conditions. However, the error begins to increase again after the minimum.

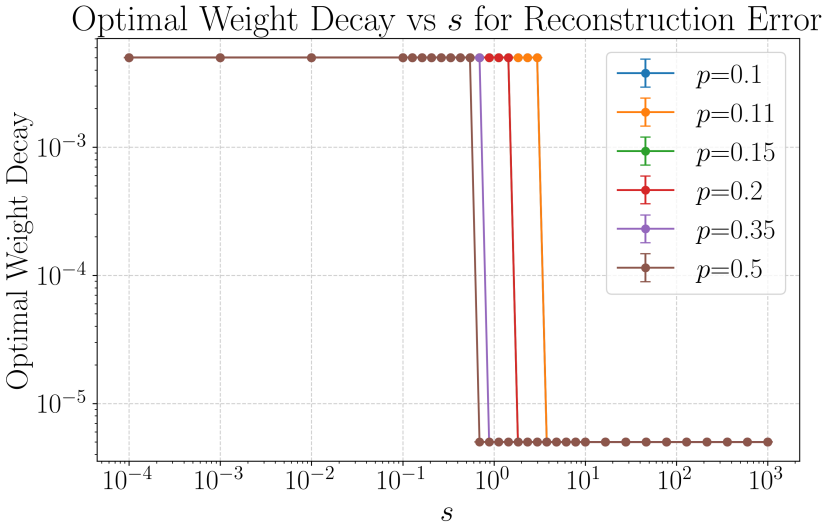


Figure 5: **Optimal Weight Decay vs Scaling Factor s.** For each s, the best-performing `weight decay` minimizing the reconstruction error is plotted. We observe that at small s, optimal w_d values are very large, shrinking $UV^\top$ to near-zero. **Hyperparameters:** $n = m = 1000$, $d = 2$, $k = 1$, $l_r = 10^{-3}$, epochs $= 30$, $reps = 5$, $p \in \{0.1, 0.2, 0.5\}$, optimized over different values of w_d.

Why does the reconstruction error increase for large values of s?

Recall that our data is generated under the Bradley-Terry-Luce (BTL) preference model, where pairwise labels are drawn from:

$$P(i \succ j \mid u) = \sigma(s(X^*_{u,i} - X^*_{u,j}))$$

This implies that the decision boundary is based on scaled differences of the latent utility matrix X^*. Consequently, a model trained to match this label distribution is effectively learning to approximate $s \cdot X^*$, rather than X^* itself.

For this reason, the reconstruction error is traditionally computed with an inverse scaling:

$$\text{Error} = \frac{\left\| X^* - \frac{UV^\top}{s} \right\|_F}{\|X^*\|_F}$$

which assumes that the learned matrix $UV^\top$ approximates $s \cdot X^*$.

Intuitively, one might expect this error to converge to 0 as s increases, since the labels become more deterministic and easier to fit. However, this is not what we observe in

practice. As s grows, the matrix $UV^\top$ tends to deviate from the simple scaling trend—either due to regularization, optimization effects, or mismatches in scale learning.

To account for this, we introduce a data-driven rescaling factor α, defined as the optimal scalar that aligns the learned matrix with the ground truth:

$$\alpha = \frac{\langle UV^\top, X^* \rangle}{\|UV^\top\|_F^2}$$

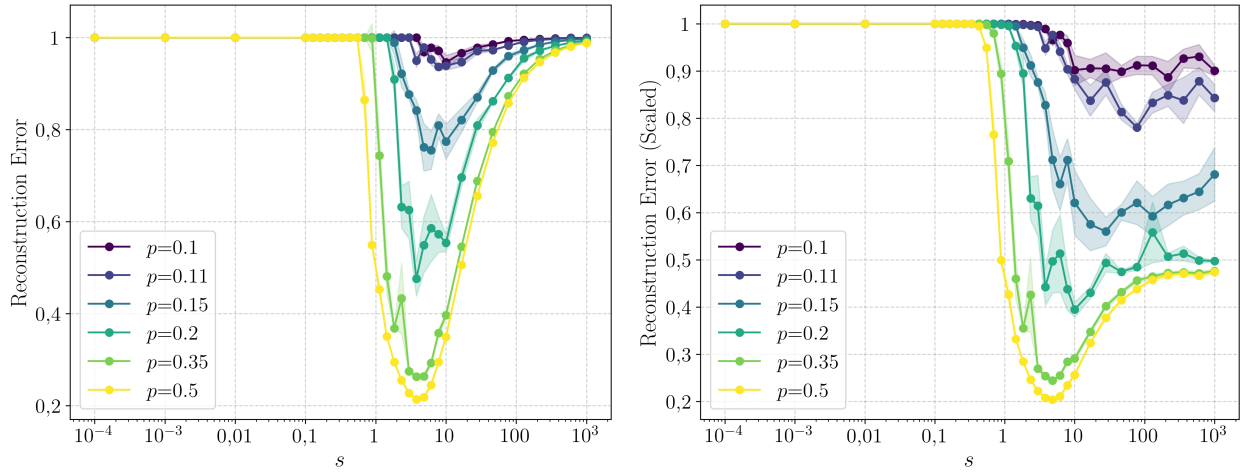
This quantity minimizes the Frobenius norm of the difference between $\alpha \cdot UV^\top$ and X^*.

Using it, we define the **scaled reconstruction error**:

$$\text{Scaled Error} = \frac{\|\alpha \cdot UV^\top - X^*\|_F}{\|X^*\|_F}$$

This metric provides a more accurate assessment of reconstruction quality, especially in high-s regimes where naive scaling no longer aligns the learned representation with the target matrix.

The resulting curve, shown in Figure 6, exhibits the desired behavior: it decreases with s and stabilizes around 0.5 for large s, independently of p. Yet, this plot still shows a local minimum before rising again, especially when $p = 0.5$.



(a) Raw Reconstruction Error (b) Scaled Reconstruction Error (with optimal α)

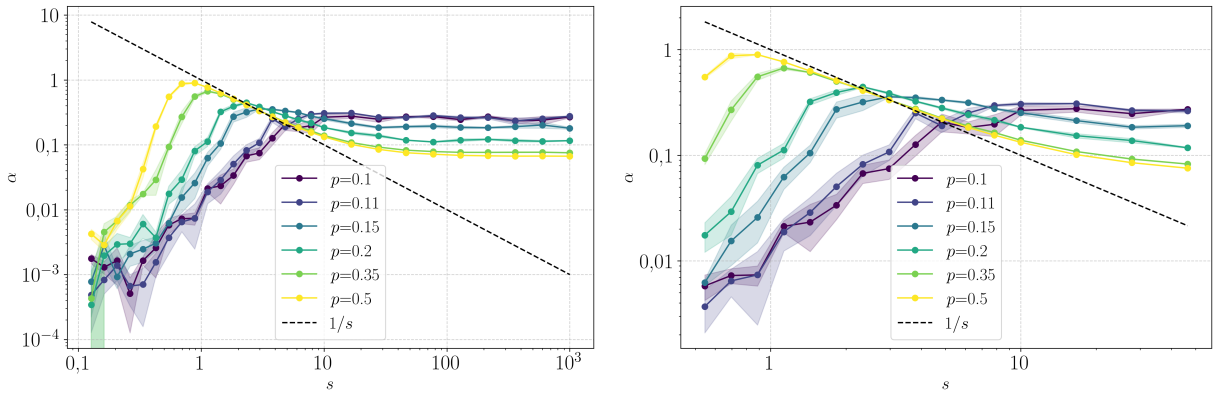
Figure 6: Comparison of raw and scaled reconstruction error as a function of scaling factor s, grouped by sparsity level p.

(a) Standard reconstruction error computed using $\frac{\|X^* - \frac{UV^\top}{s}\|_F}{\|X^*\|_F}$.

(b) Scaled reconstruction error using an optimal scalar alignment: $\alpha = \frac{\langle UV^\top, X^* \rangle}{\|UV^\top\|_F^2}$, with error computed as $\frac{\|\alpha UV^\top - X^*\|_F}{\|X^*\|_F}$.

This correction accounts for scale mismatches in $UV^\top$, revealing more consistent convergence trends across s.

Fixed hyperparameters: $n = m = 1000$, $d = 2$, $k = 1$, $lr = 10^{-3}$, $epochs = 30$, $reps = 5$, $p \in \{0.1, 0.2, 0.5\}$, optimizing over weight decay w_d.



(a) Full range (b) Zoom around optimum

Figure 7: Analysis of the optimal scaling factor α as a function of s. We compare α to the theoretical reference line $1/s$. (a): Full view from $s = 10^{-1}$ to 10^3. (b): Zoom on the region around the minimum of the scaled reconstruction error.
Fixed hyperparameters: $n = 1000$, $m = 1000$, $d = 2$, $lr = 10^{-3}$, $wd = 5 \cdot 10^{-6}$, $k = 1$, $reps = 5$.

Interpretation: To better understand the relationship between the learned scaling factor and the theoretical $1/s$ behavior assumed in the computation of reconstruction error, we visualize α in Figure 7 for each value of s. We observe that:

- Around $s = 5$, the minimum of the reconstruction error, the learned α closely follows $1/s$.
- For very small or large values of s, α deviates and can diverge.
- As the sparsity p increases, the alignment between α and $1/s$ improves.

This supports our earlier observations that the scaling mismatch in the reconstruction error is corrected when using α, particularly in the mid-range of s where the model best learns the structure of the ground-truth matrix.

While the globally scaled reconstruction error (Figure 6) corrects for uniform magnitude mismatch, it still exhibits a noticeable increase for large values of s, particularly when $p = 0.5$. This is counterintuitive: one would expect the error to remain low as model confidence improves. Surprisingly, the Pearson correlation coefficient, shown in Figure 8b, continues to increase toward 1, indicating increasingly strong alignment with the ground truth.

This apparent contradiction suggests a subtle issue: the Pearson coefficient is computed *per row* (i.e., per user), whereas the globally scaled reconstruction error applies a *single global factor* α to the entire matrix $UV^\top$. This distinction is crucial—heterogeneity in user preferences means that a single global scaling cannot align all rows of the predicted matrix simultaneously. To address this, we introduce a refined metric: the **row-wise scaled reconstruction error**, which more closely mirrors the behavior of the Pearson coefficient.

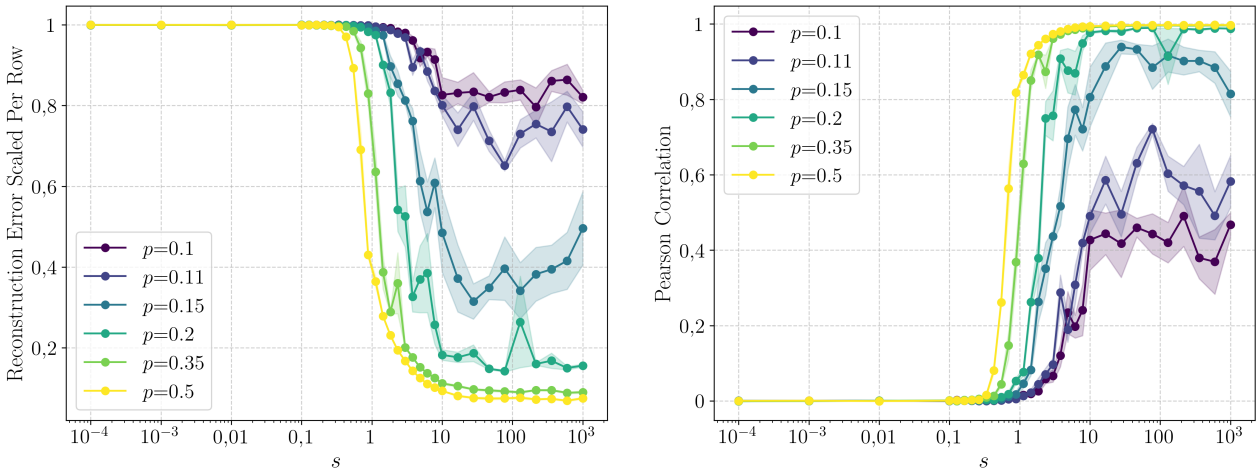
For each user u, we compute an individual optimal scaling factor:

$$\alpha_u = \frac{\langle (UV^\top)_u, X^*_u \rangle}{\|(UV^\top)_u\|_2^2} \tag{11}$$

The final error is then computed as:

$$\frac{\|\mathrm{diag}(\alpha_1, \ldots, \alpha_n) UV^\top - X^*\|_F}{\|X^*\|_F} \tag{12}$$

This row-wise version compensates for heterogeneous user magnitudes, removing artifacts caused by global mismatches in scale.



(a) Row-wise Scaled Reconstruction Error

(b) Pearson Correlation Coefficient

Figure 8: Comparison between row-wise scaled reconstruction error and Pearson correlation, as functions of the scaling factor s, grouped by sparsity level p.
(a) Each row of $UV^\top$ is individually rescaled by an optimal α_u to better align with X^*_u.
(b) The Pearson coefficient tracks how well the reconstructed user vectors preserve the ground truth preference structure.
Shared hyperparameters: $n = m = 1000$, $d = 2$, $k = 1$, $lr = 10^{-3}$, $reps = 5$, $p \in \{0.1, 0.2, 0.5\}$.

For further insights into the distribution of per-user scaling factors and their impact on reconstruction metrics, the reader is referred to Appendix A.2.

As s increases, the Pearson correlation coefficient quickly approaches 1, reflecting that the learned preferences align increasingly well with the ground truth ordering. This convergence is faster than that of the row-wise scaled reconstruction error (Figure 21d), due to differences in normalization. Nonetheless, the row-wise error confirms the same trend: the model captures user-specific ordering more accurately as s increases. The Spearman correlation (not shown) follows a similar pattern, suggesting improved ordering *and* score calibration across all rows.

In conclusion, the reconstruction error's raw trend can be misleading. A proper analysis using optimal scaling — both global and per-row — is necessary to reveal the true learning dynamics and avoid misinterpretations due to artifacts in scale or regularization. Importantly, this refined analysis shows that the model still learns the correct item differences, but they are poorly scaled — thus the model preserves preference structure even when the magnitude is off. More supporting details and diagnostics are provided in Appendix A.3.

5.2 Joint Impact of Scaling Factor s and Comparison Redundancy k

Before comparing their joint effects, we briefly recall the meaning of each parameter:

Scaling factor s controls the intensity of the preference values in the ground-truth matrix X^*. Higher values of s lead to stronger preference signals, i.e., greater differences between item scores.

Number of comparisons k defines how many independent triplet comparisons are observed per user during training. A higher k provides the model with more training data, improving robustness and reducing ambiguity in user preferences.

In the following, we explore how increasing either the expressiveness of the ground-truth matrix (via s) or the quantity of training information (via k) affects reconstruction quality and alignment with true preferences.

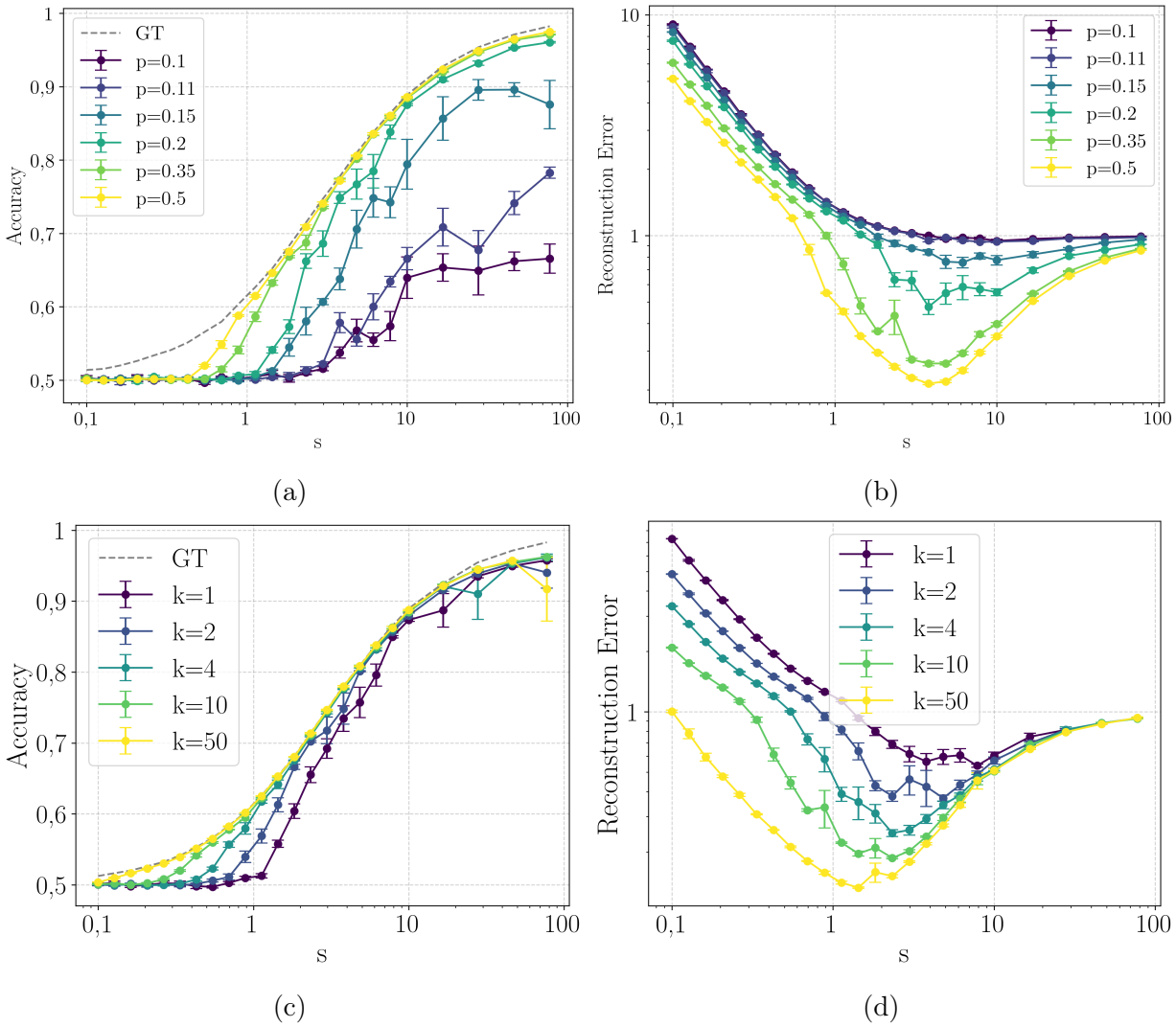


Figure 9: Analysis of the effect of the scaling factor s on accuracy and reconstruction error.
(a)-(b): Results grouped by sparsity p, with fixed $k = 1$.
(c)-(d): Results grouped by the number of comparisons k, with fixed $p = 0.2$.
Fixed parameters: $n = m = 1000$, $d = 2$, $l_r = 10^{-3}$, $w_d = 10^{-5}$, epochs $= 30$, $reps = 5$.

Figure 9 investigates how the scaling factor s affects model performance depending on the number of comparisons seen during training.

In subplots (a)-(b), we fix $k = 1$ and vary the sparsity level p, i.e., the proportion of triplets observed. As p increases, the model has access to more data. This results in both improved reconstruction accuracy and a more reliable prediction of the correct preference direction. In other words, seeing more comparisons per user—even if each comparison is only observed once—allows the model to generalize better.

Subplots (c)-(d) show the effect of increasing k, i.e., the number of repetitions per comparison triplet, for a fixed $p = 0.2$. Here too, we observe that increasing k leads to better alignment with the true matrix and improved reconstruction error. This is because multiple observations of the same underlying preference help reduce noise and enforce consistency in the learned matrix.

Overall, both increasing p (number of distinct comparisons) and k (repetition per comparison) lead to better approximation of the ground-truth matrix, but in complementary ways. Increasing p brings more unique information, while increasing k reinforces existing signals. In both cases, we converge toward better generalization as measured by accuracy and reconstruction metrics.

6 Impact of Comparison Redundancy k on Learning Dynamics

Fixed hyperparameters for this section: $n = 1000$, $m = 1000$, $d = 2$, $p = 0.2$, $lr = 10^{-3}$, epochs $= 30$, reps $= 5$.

In this section, we analyze the influence of the number of comparisons k—the number of repetitions of a triplet (u, i, j)—on model performance.

Role of Weight Decay and Scaling

Figure 25 clearly shows the importance of weight decay optimization when comparing results across different values of k. For each k, the reconstruction error reaches its minimum at a different scaling factor s, and this minimum shifts depending on the value of the weight decay.

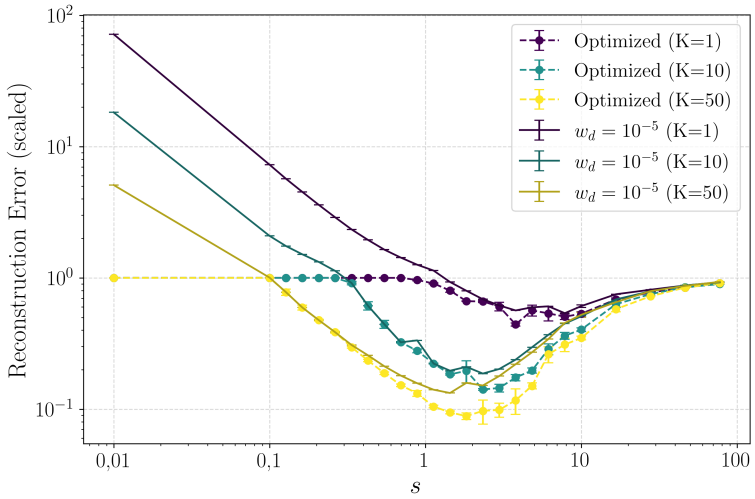


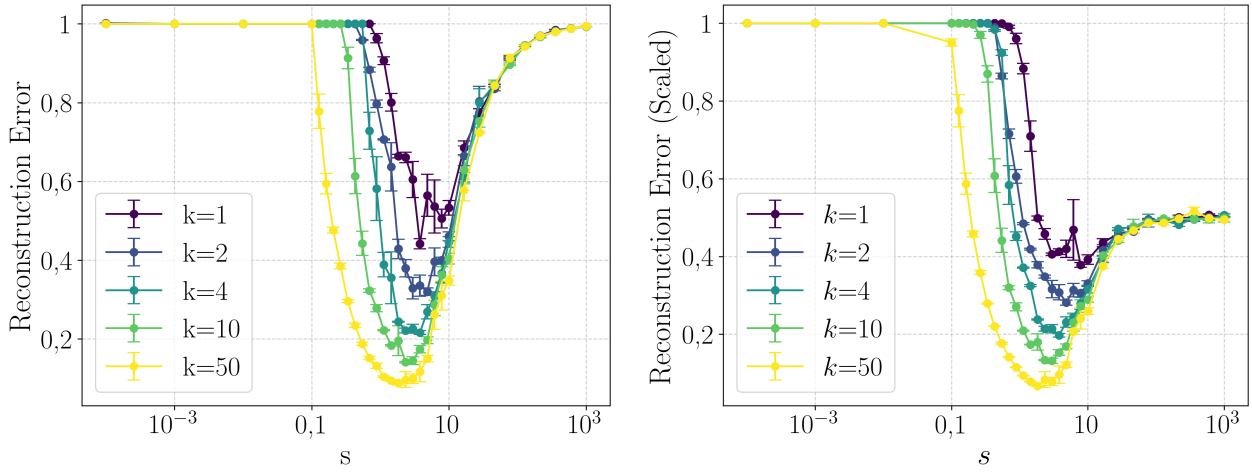
Figure 10: Comparison of reconstruction error across s for fixed vs. optimized weight decay, grouped by k.

To explore these interactions in more detail — particularly how weight decay interacts with s and k to influence accuracy and reconstruction — we include additional figures and analysis in Appendix A.4.

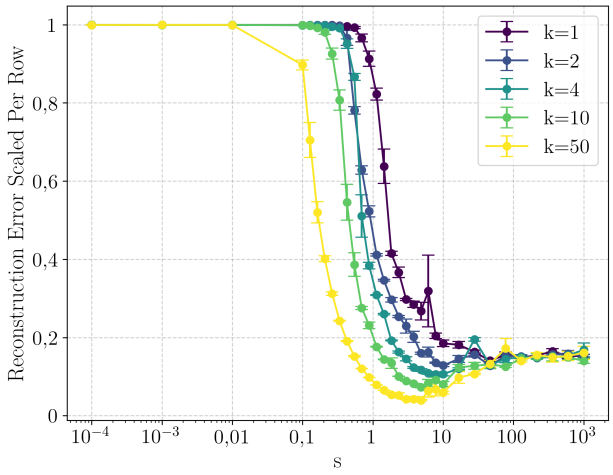
Final Optimized Results for k analysis

After selecting optimal weight decay values per configuration, we evaluate reconstruction error and alignment metrics across k in Figures 11a–c. Figure 11a shows the standard reconstruction error using the best weight decay. Figure 11b scales the error globally with the optimal alignment factor α, while Figure 11b performs row-wise scaling.

In all three cases, we observe that increasing k leads to better reconstruction, smoother convergence curves, and improved consistency between predicted and ground-truth matrices.



(a) Standard Reconstruction Error (b) Globally Scaled Reconstruction Error



(c) Row-wise Scaled Reconstruction Error

Figure 11: Reconstruction error metrics across s values after optimization of weight decay, for varying numbers of comparisons k.
(a) Raw reconstruction error $\|UV^\top - X^*\|_F$.
(b) Scaled version: $\|\alpha UV^\top - X^*\|_F / \|X^*\|_F$, with α computed via best alignment.
(c) Row-normalized error to evaluate structure preservation across users.

Interestingly, in Figure 11b, we observe that for very large values of the scaling parameter s, the row-scaled reconstruction error increases again and tends to match the error observed for $k = 1$, even when $k > 1$. This phenomenon can be explained by the behavior of the Bradley–Terry–Luce (BTL) model as $s \to \infty$: the sigmoid preference model becomes increasingly sharp and behaves like a step function. Consequently, the model predicts almost exclusively 0 or 1 for all comparisons (u, i, j), regardless of how many observations are available per user. This saturation limits the expressiveness of the model, and we effectively revert to the $k = 1$ regime, where only binary preference outcomes are captured. Therefore, increasing k no longer improves the row-wise reconstruction quality in this extreme regime.

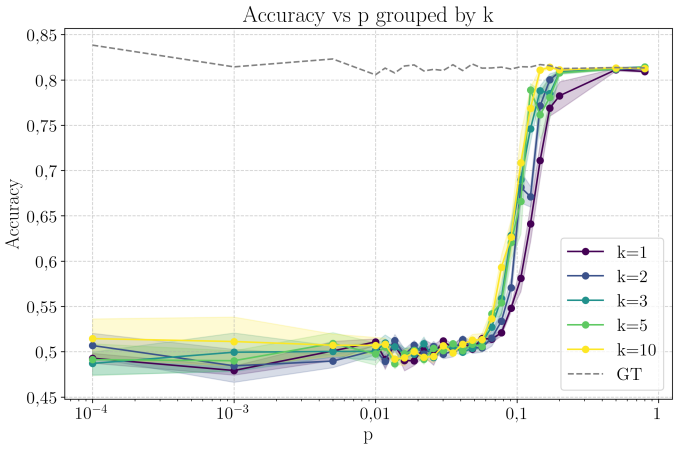
For a deeper understanding of how the scaling factor s interacts with the learned magnitude of the reconstructed matrix, we refer the reader to Appendix A.5. There, we analyze the evolution of the optimal alignment coefficient α and its deviation from the expected $1/s$ behavior.

For further insights on the Pearson correlation for different values of k and its distinction with the reconstruction error scaled per row — see the extended slope analysis in Appendix A.3.

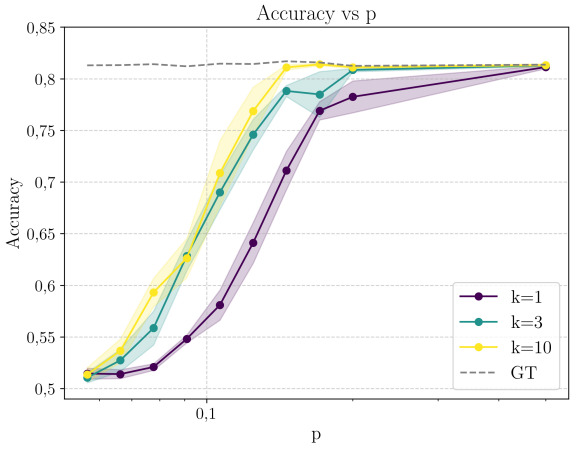
7 Balancing Sparsity and Repetition: Joint Impact of p and k

In this section, we investigate how the sparsity level p (i.e., the proportion of available triplet comparisons) and the number of repeated comparisons k interact. Since both factors enhance model performance independently, one may ask whether they can compensate for each other — i.e., whether increasing k could mitigate the effect of low p, or vice versa.

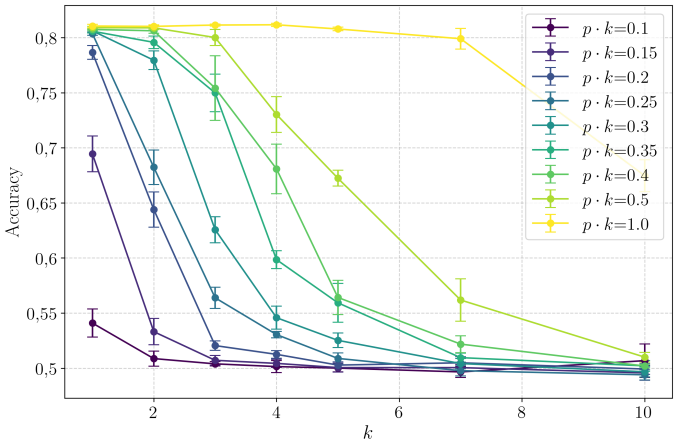
From Figures 12a and b, we observe that higher values of k significantly improve performance even when p is small. For instance, with $k = 10$, accuracy increases sharply with modest increments of p, whereas for $k = 1$, a much denser dataset is needed to reach similar accuracy levels. This suggests that repeated sampling over the same preferences enhances signal quality and stabilizes learning.



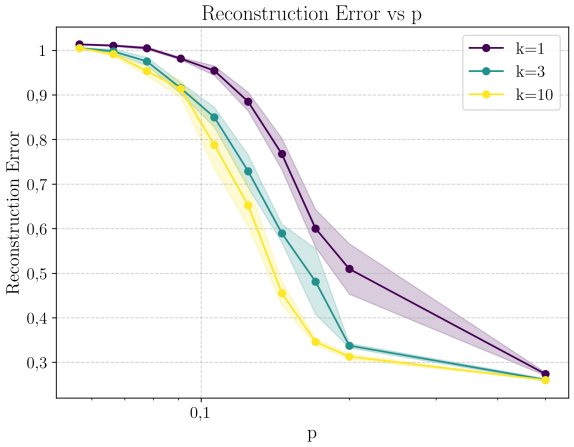
(a) Accuracy vs p, grouped by k



(b) Zoom of plot (a)



(c) Accuracy vs k, grouped by constant $p \cdot k$



(d) Reconstruction Error vs p, grouped by k

Figure 12: Analysis of validation accuracy and reconstruction error as functions of sparsity p and number of comparisons k.
(a) Validation accuracy vs p, grouped by values of k.
(b) Zoomed view for $k \in \{1, 2, 3, 10\}$.
(c) Accuracy vs k, grouped by constant $p \cdot k$ to control information density.
(d) Reconstruction error vs p, grouped by k.
Fixed parameters: $n = m = 1000$, $d = 2$, $s = 5.0$, $l_r = 10^{-3}$, $w_d = 10^{-5}$, $reps = 5$.

Figure 12c explores the idea of grouping experiments by constant $p \cdot k$ — a proxy for the total number of preference signals per user. The intuition is to keep information density fixed

while varying the relative contribution of p and k. However, we find that this normalization does not yield consistent accuracy across k. In particular, when p is very small, even large k fails to recover meaningful structure, highlighting that pure repetition cannot fully replace data diversity.

Figure 12d confirms this trend for the reconstruction error: increasing k helps mitigate sparsity, but does not entirely substitute for richer observation matrices.

This suggests that even under a constant label budget, the trade-off between repetition and data coverage is complex and depends on more than just the total amount of supervision. Increasing k helps stabilize preference estimation, but the benefit depends heavily on the signal-to-noise ratio induced by s, and cannot be fully compensated for by simply fixing $p \cdot k$.

For additional insights, we further analyze whether keeping the product $p \cdot k$ constant leads to a clear optimal value of k under different scaling regimes. The detailed results, including validation accuracy and reconstruction error grouped by constant label density, are presented in Appendix A.6.

8 Exploring the Trade-off Between p and s

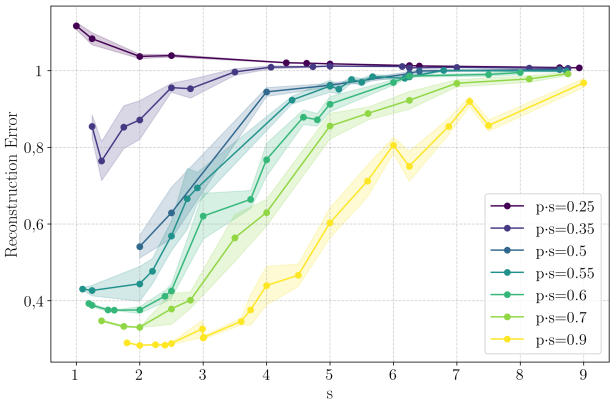
In this section, we investigate whether an optimal balance exists between the sparsity level p (i.e., the fraction of available comparisons) and the preference sharpness parameter s (which controls how decisive the comparisons are). Intuitively, both parameters contribute to the effective information provided to the model: higher p increases the number of training samples, while higher s amplifies the contrast between item scores. Our goal is to understand whether a principled trade-off can be made — and if so, whether a meaningful optimality can be observed in practice through accuracy and reconstruction-based metrics.

Subfigure 13c shows that as the sparsity p increases, the model is able to learn more effectively across a wider range of s values. For small p, accuracy remains low unless s is very high. This is expected, since with limited data, the model relies on strong (i.e., sharp) preference signals to extract useful information. As p increases, however, the accuracy improves even at moderate values of s, indicating that the model benefits from a richer dataset and no longer requires extreme scaling to learn effectively.

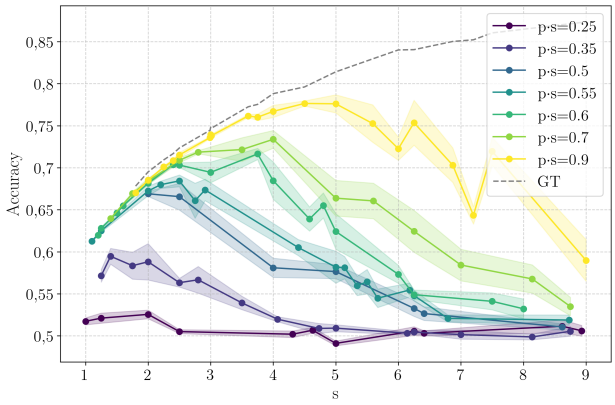
Subfigure 13b provides another perspective by grouping results according to constant values of $p \cdot s$, which represent the effective signal-to-noise ratio in the preference labels. In this setting, the quantity $p \cdot s$ can be seen as a proxy for the signal-to-noise ratio: increasing p reduces the impact of randomness through data aggregation, while increasing s sharpens the preference signal — making it easier for the model to learn meaningful structure. We observe that for sufficiently large values of $p \cdot s$, the model reaches high accuracy — but only up to a point: for each level of $p \cdot s$, there seems to be a local optimum in s, beyond which the accuracy decreases. This suggests that the best s depends not only on the dataset size (p), but also on how sharp or noisy the comparisons are.

From a recommender system perspective — for example, predicting movie preferences — this means that if we collect only a few comparisons per user (low p), we need those comparisons to be very clear and decisive (high s), i.e., users strongly favor one movie over another. But if we have more data per user (high p), even subtle preferences (lower s) can be sufficient for the model to learn, since it can aggregate weaker signals over more observations.

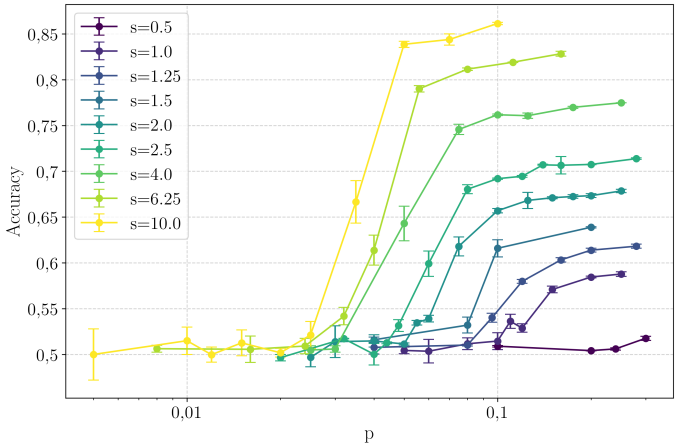
Finally, subfigure 13a presents the reconstruction error under constant $p \cdot s$. While some trends are visible, no clear minimum appears. This suggests that the unscaled reconstruction error is not always reliable for identifying the best learning regime.



(a) Reconstruction Error vs s, grouped by $p \cdot s$



(b) Accuracy vs s, grouped by constant $p \cdot s$



(c) Accuracy vs p, grouped by scaling factor s

Figure 13: Joint analysis of scaling factor s and sparsity p.
(a) Reconstruction error vs s for fixed $p \cdot s$ values — optimality is harder to identify here than in scaled row-wise reconstruction.
(b) Accuracy vs s grouped by constant values of $p \cdot s$ (proxy for signal-to-noise ratio).
(c) Validation accuracy as a function of p, grouped by s.
Fixed parameters: $n = m = 1000$, $d = 2$, $k = 1$, $l_r = 10^{-3}$, $w_d = 10^{-5}$, $reps = 7$.

So far, we have controlled the scaling factor s explicitly to influence the clarity of preference signals. In the following section, we explore an alternative approach: how different *sampling strategies* can implicitly act as varying s values. That is, some triplet selection methods naturally produce easier comparisons (akin to a higher s), while others introduce ambiguity or difficulty (lower effective s). This perspective allows us to reinterpret the effect of sampling through the lens of preference separability and optimization dynamics.

9 Results on Triplet Sampling Strategies

To study matrix factorization from comparison data (binary preferences between two items), several strategies for generating triplets (u, i, j) are implemented. Each strategy tests specific inductive biases or learning behaviors in optimization. Below, we summarize and detail their implementation and behavior.

1. Random

Description: User u, and two distinct items i, j are sampled uniformly at random.
Goal: Serve as a baseline.
Implementation: Uses `torch.randint` over the user and item space. Avoids duplicates and ensures $i \neq j$.

2. Max-Min-based

Description: For user u, item i is sampled among the top-k highest scores and item j among the bottom-k.
Goal: Enforce highly separable comparisons.
Implementation: For each user, compute X_u, and sample from `torch.topk(X_u, k)` and $-X_u$. Default $k = 100$ or capped by m.
Expected Effect: Fast learning since sampled comparisons are likely to reflect strong preferences.

3. Close-Call Strategy

Description: For user u, two items i, j are chosen such that $|X_{ui} - X_{uj}| \leq \delta$, where δ is an adaptive margin.
Goal: Emphasize ambiguous comparisons (learning boundaries).
Implementation: We compute the average score spread for the first $k = \min(10, n)$ users:

$$\text{spread}_u = \max_j X_{uj} - \frac{1}{m}\sum_{j=1}^{m} X_{uj}, \quad \delta = \left(\frac{1}{k}\sum_{u=1}^{k} \text{spread}_u\right) \cdot p$$

Pairs (i, j) are resampled until the condition $|X_{ui} - X_{uj}| \leq \delta$ is met.
Expected Effect: Refines the model on difficult decisions but may increase label noise.

4. Popularity-based

Description: Items i, j are drawn from a predefined popularity distribution.
Goal: Simulate real-world biases (e.g., Zipfian).
Implementation:

- Zipf: $p_k \propto 1/k^\alpha$

Sampling is done using `np.random.choice` based on these probabilities.
Expected Effect: Helps assess robustness to skewed distributions. Bias toward frequent items.

5. Top-10% Sampling

Description: Items i, j are both selected from the user's top-k preferred items.
Goal: Study subtle distinctions within a user's favorite items.
Implementation:

- k is either set explicitly or estimated automatically:

$$k = \min(m, \max(5, \lfloor 0.1 \cdot m \rfloor))$$

- For each user, compute `torch.topk(X_u, k)`.

Expected Effect: Local preference resolution is emphasized. These comparisons are hard and encourage finer embedding alignment.

6. SVD Projection

Description: Project users and items onto principal components via truncated SVD.
Goal: Focus sampling on dominant latent factors.
Implementation:

- Perform truncated SVD: `scipy.sparse.linalg.svds(X, k=max(n,m)p/2)`.
- Compute ℓ_2 norms of projected vectors (of $S \cdot X_{SVD}$ and $(S \cdot V_{SVD})^\top$).
- Keep top 30% users/items by norm and sample triplets within this subset.

Expected Effect: Accelerates convergence by guiding learning toward principal structure. Less sensitivity to local noise.

Description of Groups of Strategies

We divided the sampling strategies into two distinct groups to facilitate clearer visual comparisons based on their underlying mechanisms:

- Group 1 includes: `random`, `Max-Min`, `Close-Call`, `top_10%`, and `svd`. These methods tend to focus on local or structure-driven properties (e.g., random sampling, distance-based comparisons, or embedding norms).
- Group 2 includes: `random` and `popularity`. These strategies emphasize distributional or statistical signals (e.g., frequency, variability, or group-based structure).

This separation allows us to highlight behavioral trends within more homogeneous sets of strategies while maintaining a common baseline (`random`) across both groups.

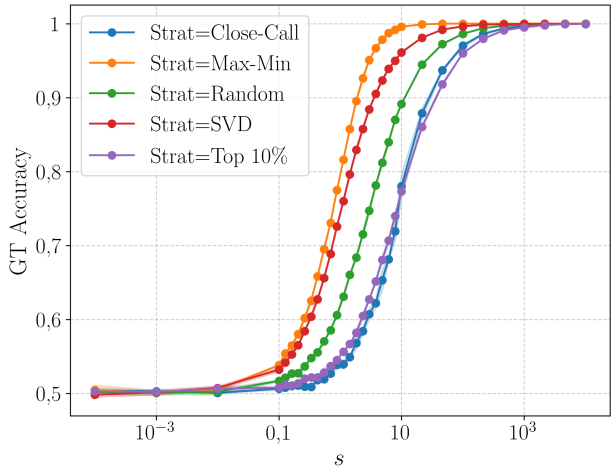
9.1 Impact of the Scaling Factor s Across Sampling Strategies

Strategy Evaluation – Group 1

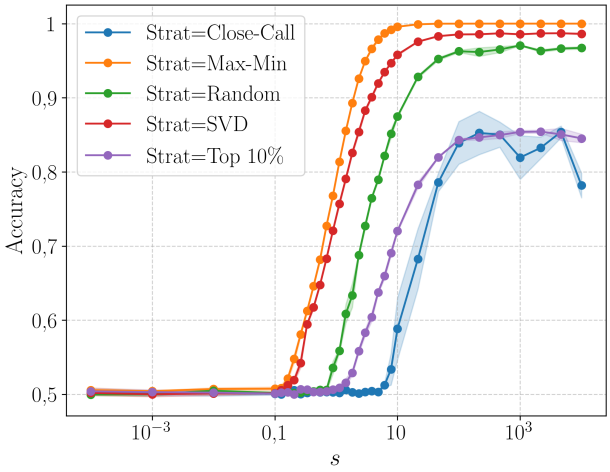
We analyze the performance of five triplet sampling strategies: `random`, `Max-Min`, `svd`, `Close-Call`, and `top_10%`. These plots evaluate the impact of the scaling factor s across different metrics.

Fixed hyperparameters: $n = m = 1000$, $d = 2$, $p = 0.2$, $l_r = 10^{-3}$, epochs $= 30$, $reps = 3$, $k = 1$, $w_d = 10^{-5}$.

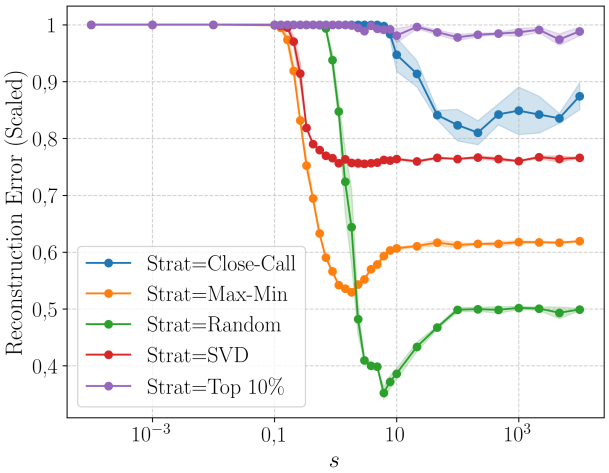
We observe in Figure 14 that the strategies `Max-Min` and `svd` achieve notably high ground-truth accuracies. This is expected, as both methods sample triplets with large differences in scores, which are easier to label correctly and thus naturally inflate the accuracy.



(a) GT Accuracy



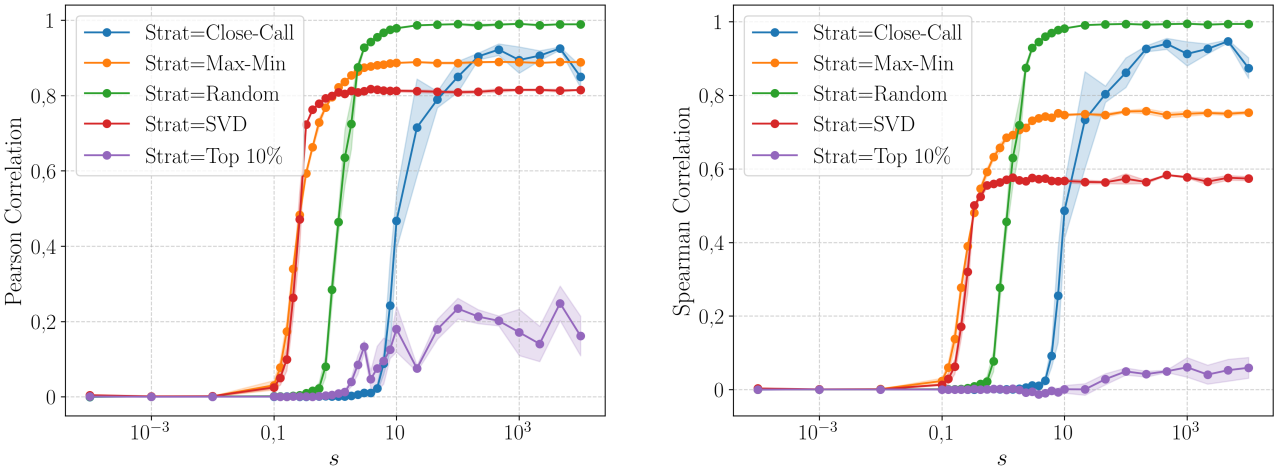
(b) Accuracy



(c) Scaled Reconstruction Error

Figure 14: Performance metrics across scaling factor s for Group 1 strategies.

In contrast, strategies such as `top_10%` and `Close-Call`, which select item pairs with close scores (i.e., ambiguous or near-boundary comparisons), result in significantly lower GT accuracy and validation accuracy. Their scaled reconstruction errors also reflect this trend: they either decrease very late (in the case of `Close-Call`) or remain high across all s values (in the case of `top_10%`), indicating difficulty in learning a faithful reconstruction. Interestingly, while one might expect that high GT accuracy would translate into better matrix reconstruction, this is not the case. Although `Max-Min` maintains relatively low scaled reconstruction error, `svd` fails to match even the `random` baseline, suggesting that it captures preference orderings without preserving magnitude or structure. This highlights a key insight: high classification accuracy on triplets does not guarantee a low reconstruction error — a method can perfectly identify which item is preferred but still fail to recover the underlying scoring function accurately.



(a) Pearson Correlation

(b) Spearman Correlation

Figure 15: Alignment metrics for Group 1 strategies.

Figure 15 provides further insights into the structural alignment between the predicted matrix $\hat{X} = UV^\top$ and the ground truth X^*. Surprisingly, the `svd` strategy achieves very high values for both Pearson and Spearman correlations, despite its poor reconstruction error observed previously. This suggests that while the model fails to recover the absolute values of the scores, it preserves their relative structure — both in linear correlation (Pearson) and rank order (Spearman). This behavior is expected from a method that uses singular vector projections, which inherently preserve global directions but may ignore magnitude scaling.

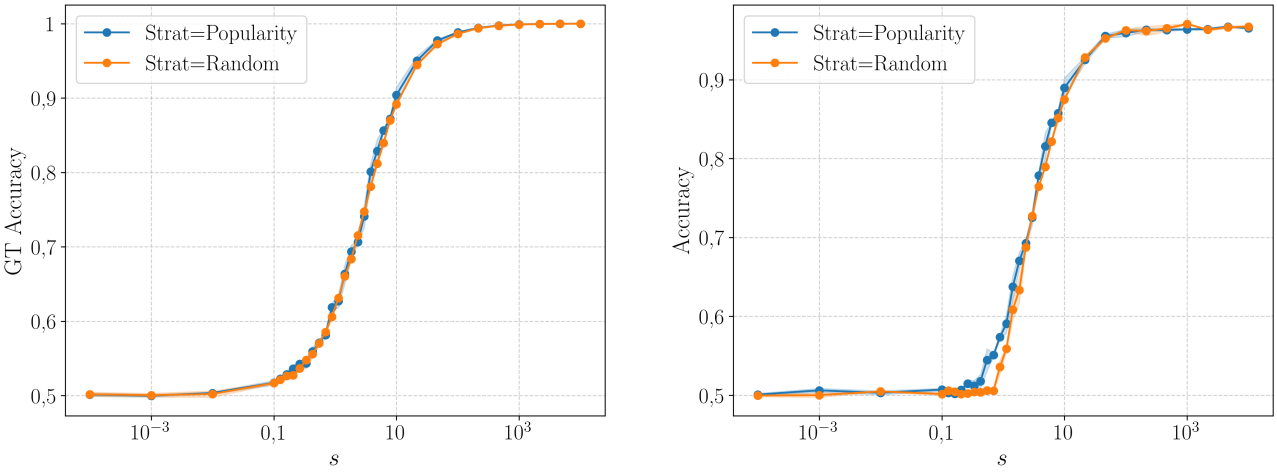
On the other hand, `top_10%` continues to underperform across both metrics, confirming its weakness in capturing any meaningful structure from the comparison data.

Another notable observation is that both `Max-Min` and `svd` strategies exhibit higher Pearson correlations than Spearman. This implies that these methods are better at preserving the relative score differences (i.e., linear scaling) than the exact ranking of items. This is particularly reasonable for `Max-Min`, which favors comparisons with strong score contrasts, making it easier for the model to infer relative magnitude but not necessarily the full ranking among similar items.

Strategy Evaluation – Group 2

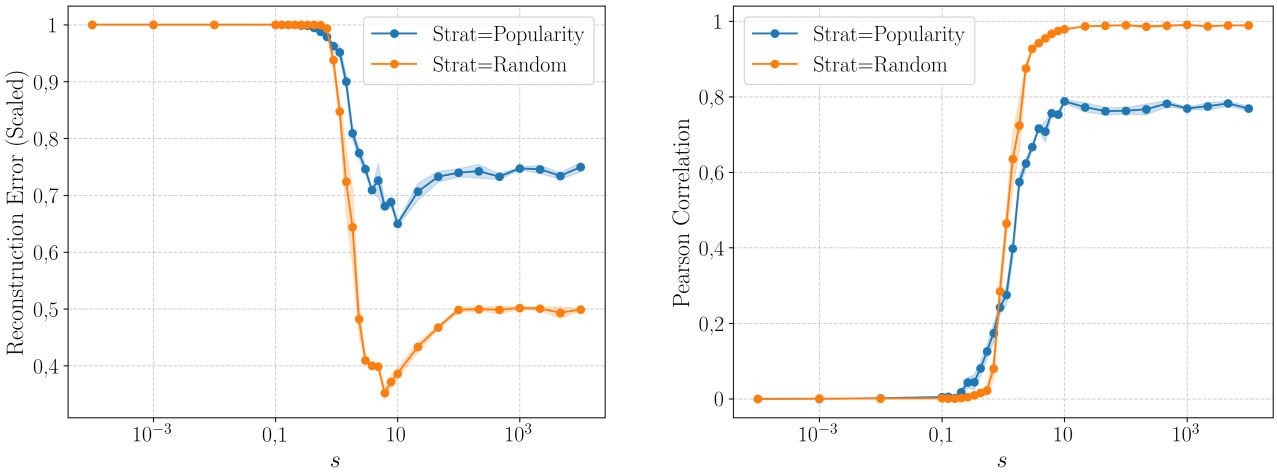
We now evaluate three alternative strategies: `random` and `popularity`. As before, we analyze how the scaling factor s affects model behavior.

Fixed hyperparameters: $n = m = 1000$, $d = 2$, $p = 0.2$, $l_r = 10^{-3}$, epochs $= 30$, $reps = 3$, $k = 1$, $w_d = 10^{-5}$.



(a) GT Accuracy

(b) Validation Accuracy



(c) Scaled Reconstruction Error

(d) Pearson Correlation

Figure 16: Performance and alignment metrics across scaling factor s for Group 2 strategies.

Figure 16 presents the performance and alignment metrics for the second group of strategies, namely `base` and `popularity`.

The `popularity` strategy achieves similarly high GT accuracy and reconstruction error levels, yet it consistently lags in actual validation accuracy. This gap indicates that while the model may reconstruct the preference matrix reasonably well under popularity-based sampling, it struggles to generalize those preferences into accurate predictions.

This discrepancy reveals an important limitation: selecting comparisons based on the most popular items may reduce the diversity of observed preferences, making it harder for the model to learn nuanced distinctions. In other words, even if we are confident in the preference labels (i.e., high s), the informational content remains limited when comparisons are drawn from a narrow pool of highly popular items. This highlights a practical challenge in real-world applications such as movie recommendation: relying solely on comparisons involving popular items may lead to poor personalization performance.

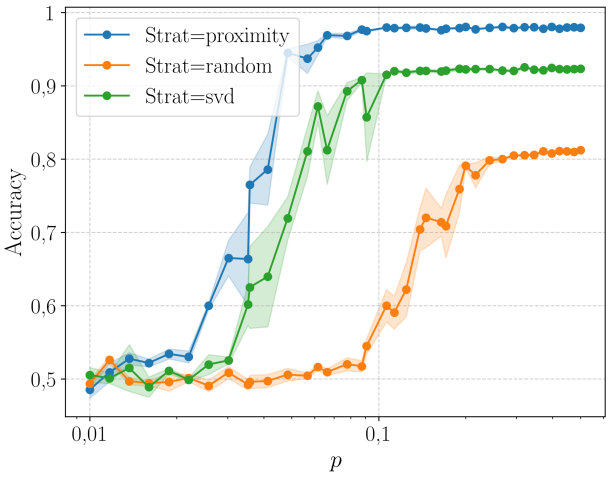
Finally, the Pearson correlation metric in subfigure (d) mirrors the behavior of the reconstruction error (subfigure c). The alignment with the ground-truth matrix increases with s, particularly for the `base` strategy — reinforcing the idea that higher s improves the model's ability to capture fine-grained user preferences when the sampling strategy is sufficiently diverse.

9.2 Impact of Sparsity Level p Across Sampling Strategies

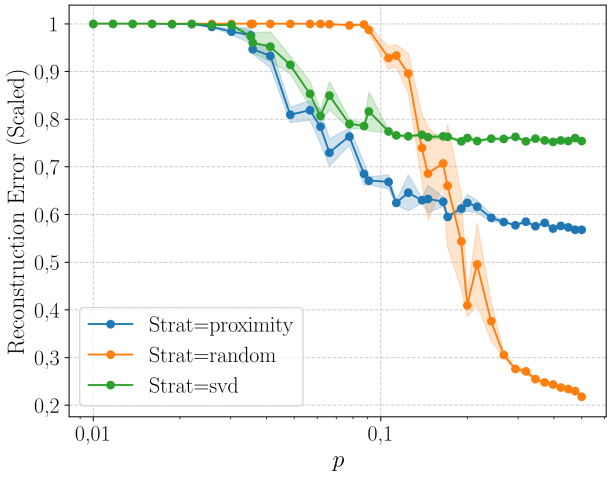
We now evaluate the performance of different sampling strategies as a function of the sparsity parameter p, with fixed scaling $s = 5$. As previously, we divide the strategies into two groups for clarity.

Fixed hyperparameters: $n = m = 1000$, $d = 2$, $s = 5$, $l_r = 10^{-3}$, $w_d = 10^{-5}$, epochs $= 30$, $k = 1$, $reps = 3$.

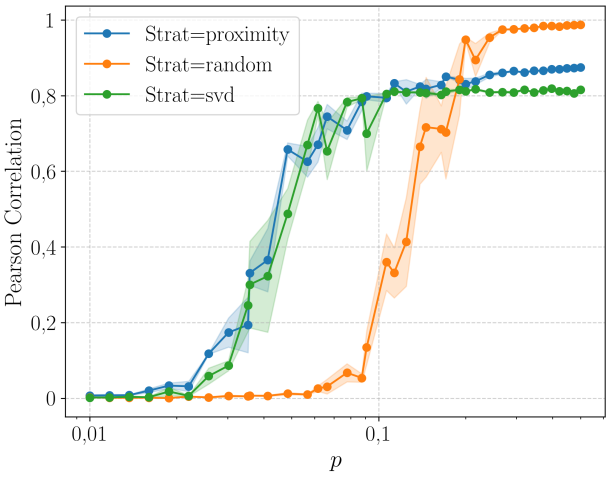
Note: The strategies `top_10%` and `Close-Call` were excluded from the plots due to consistently poor or uninformative behavior across the parameter range.



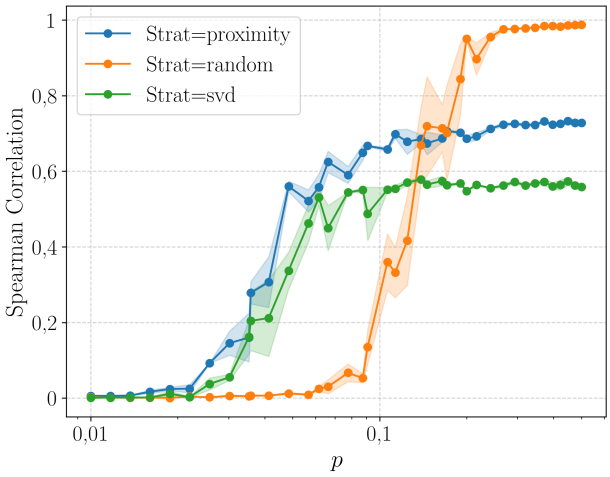
(a) Validation Accuracy – Group 1



(b) Scaled Reconstruction Error – Group 1



(c) Pearson Correlation – Group 1



(d) Spearman Correlation – Group 1

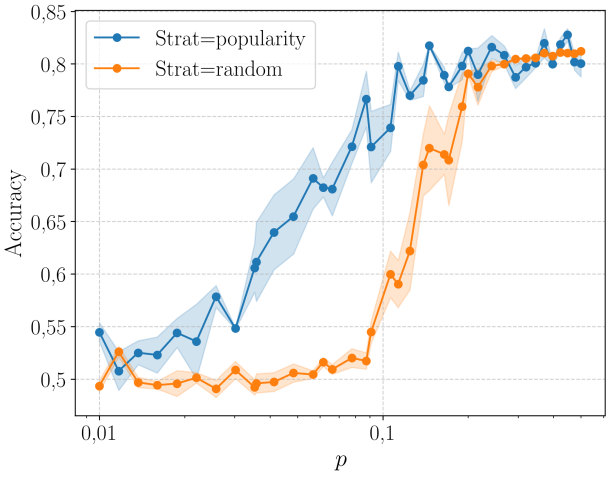
Figure 17: Performance and alignment metrics across sparsity p for Group 1 strategies: `random`, `Max-Min`, and `svd`.

Figure 17 shows how validation accuracy, scaled reconstruction error, and alignment metrics vary with the sparsity p for Group 1 strategies: `random`, `Max-Min`, and `svd`.

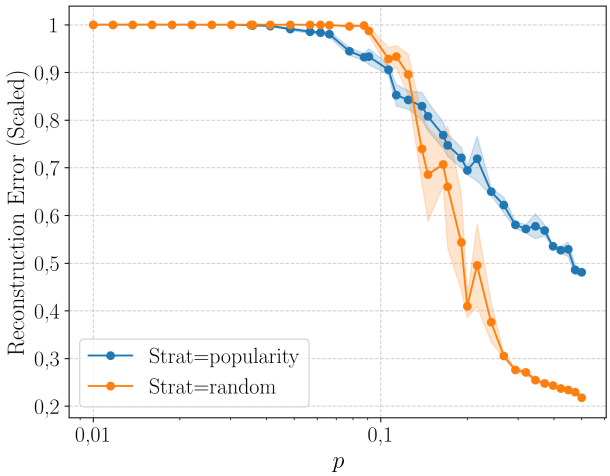
As expected — and consistent with our analysis over the scaling factor s — increasing p generally improves accuracy and correlation scores while decreasing the reconstruction error. However, an important difference emerges here: `Max-Min` and `svd` perform significantly better than `random` when p is small. This suggests that when only a few comparisons are available per user (low p), focusing on items with highly dissimilar scores (as done by `Max-Min` and `svd`) yields much more informative training data.

This advantage is visible not only in the accuracy and reconstruction error curves but also in the alignment metrics. Both the Pearson and Spearman coefficients increase more rapidly for `Max-Min` and `svd` compared to `random`, especially in the low-p regime. Interestingly, as in the previous analysis with s, the Spearman correlation consistently exceeds the Pearson correlation for these two strategies. This further confirms that these methods are better at preserving the **relative order** of the reconstructed values, even if the exact **magnitudes** are not perfectly aligned.

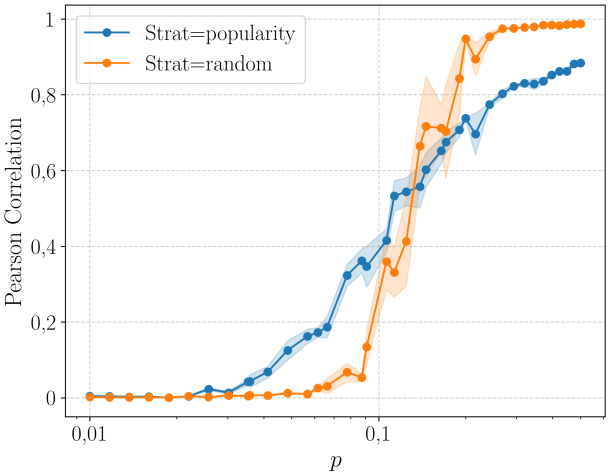
Overall, these results emphasize that strategic sampling — particularly targeting pairs with strong contrast — can be especially beneficial in sparse-data scenarios.



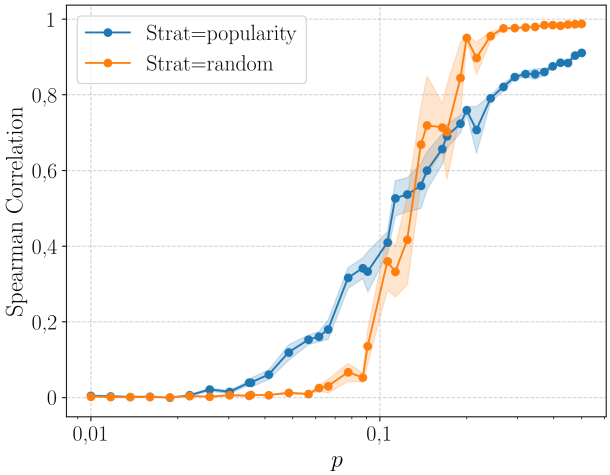
(a) Validation Accuracy – Group 2



(b) Scaled Reconstruction Error – Group 2



(c) Pearson Correlation – Group 2



(d) Spearman Correlation – Group 2

Figure 18: Performance and alignment metrics across sparsity p for Group 2 strategies: `random` and `popularity`.

Figure 18 presents the performance metrics and alignment scores as a function of sparsity p for the Group 2 strategies: `random` and `popularity`.

The `popularity` strategy stands out in the low-p regime: it consistently yields higher validation accuracy when only a small number of comparisons are available. This suggests that, in extremely sparse settings, comparing each item to a small pool of well-known or frequently compared reference items can help stabilize learning and improve predictive performance.

However, the picture is less clear when examining the scaled reconstruction error. The `popularity` curve does not follow the same clear improvement pattern as its accuracy, making

this advantage more debatable from a reconstruction standpoint.

To resolve this ambiguity, we turn to the Pearson and Spearman correlation coefficients. Both metrics show that `popularity` outperforms `random` more convincingly, even at low p. This implies that the learned embeddings are better aligned with the underlying structure of the data — both in terms of intensity (Pearson) and relative ordering (Spearman).

This result is somewhat surprising: one might expect that using a fixed set of reference items — as in `popularity` — would lead to biased learning, overly focused on a small region of the item space. Yet the model seems to leverage these anchors effectively, even improving global alignment. This highlights a counterintuitive phenomenon: under certain sparsity conditions, reduced diversity in item comparisons may actually help structure the embedding space more coherently.

10 Conclusion

This work presents a detailed and systematic analysis of matrix factorization from comparison data, extending classical MF frameworks to a setting where supervision comes from binary preferences rather than explicit ratings.

Through controlled synthetic experiments, we evaluated how different structural and algorithmic parameters impact both the reconstruction of the ground-truth matrix and the ability to recover user-specific preference structures. In doing so, we examined:

- The role of the scaling factor s in shaping the signal-to-noise ratio of the data;
- The influence of data sparsity p and comparison redundancy k on model performance;
- The effects of various triplet sampling strategies on learning behavior;
- The sensitivity of standard metrics (e.g., reconstruction error) to normalization, and the benefits of alignment-aware evaluations using optimal global and row-wise scalings.

Our findings reveal several key insights:

- **Lower latent dimension (d) improves learnability in low-data regimes.** Reducing the number of latent factors concentrates information and reduces overfitting, leading to better generalization — especially when the amount of observed data is limited.
- **Confidence and coverage must be balanced.** When the preference sharpness (via s) increases, learning improves up to a point — but only if enough data is available. Beyond that, high s demands higher observation density (p) to avoid instability. The optimal accuracy emerges from a suitable trade-off between signal confidence and data coverage.
- **Redundancy (k) and sparsity (p) are not interchangeable**. While both contribute positively, they interact non-trivially. In particular, increasing k under fixed label budgets does not always yield gains.
- **Standard reconstruction error can be misleading**. Without correcting for scaling effects, it underestimates performance in many regimes. Global and row-wise scaled errors reveal that the model captures preference *structure* even when magnitudes are misaligned.
- **Sampling strategy plays a crucial role in low-data settings**. Strategies like MaxMin or SVD-based sampling outperform uniform baselines when data is sparse, suggesting that prioritizing contrastive comparisons can accelerate learning.

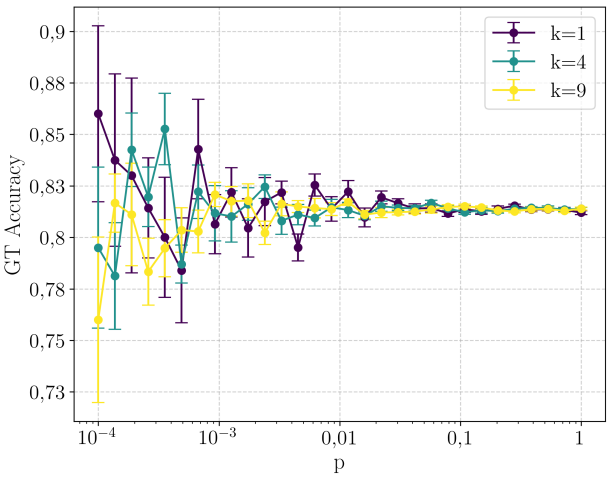
In conclusion, learning from comparative preference data offers a powerful yet subtle framework. To fully understand and optimize such models, one must go beyond classical loss functions and standard evaluation pipelines. Proper normalization, sensitivity analyses, and understanding of structural dynamics (e.g., how s, p, k, and strategy interact) are necessary for robust and interpretable learning.

Future directions include extending this analysis to real-world datasets, testing on partial or biased comparison graphs, and exploring adaptive sampling strategies that dynamically adjust based on learning uncertainty.

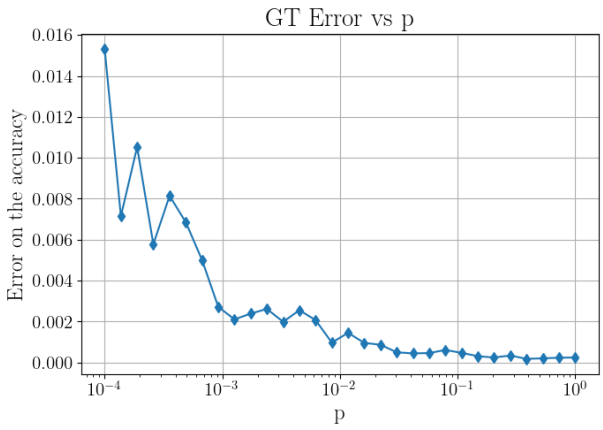
A Appendix

A.1 Stability of Ground Truth Accuracy

To verify the reliability of our evaluation metrics, we analyze how the ground-truth accuracy behaves as a function of the sparsity level p and the number of comparisons k. These experiments confirm that our accuracy metrics are stable and interpretable across a broad range of data regimes. This validates the use of GT Accuracy as a foundation for evaluating model behavior throughout the report (see references in Section 3).



(a) Ground-truth accuracy across varying sparsity p and number of comparisons k



(b) Standard error of the mean on GT accuracy as a function of p

Figure 19: Stability of ground-truth accuracy under different data sparsity and comparison redundancy settings.
(a) GT Accuracy vs p and k.
(b) Error bars (standard error) vs p.

Figure 19(a) shows that the **ground-truth accuracy remains remarkably stable across values of p and k**, particularly as k increases. This provides strong evidence that our evaluation baseline is robust for downstream analysis.

In Figure 19(b), we observe that **increasing p significantly reduces the variance of the GT Accuracy estimates**, as reflected by shrinking error bars. The standard error decays exponentially, confirming the statistical consistency of our estimates. Even though the errors remain small for most p, this behavior should be considered when interpreting narrow differences in model performance across settings.

These results justify the use of GT Accuracy throughout the paper as a reliable reference metric.

A.2 Global vs. Row-Wise Reconstruction Error

To better understand the discrepancies between the global reconstruction error and its row-wise counterpart, we analyze the distribution of the optimal scaling coefficients α_u used per user. These values represent how each row of $UV^\top$ must be rescaled to best align with its corresponding row in the ground-truth matrix X^*.

This section illustrates how the variability in these α_u values impacts the effectiveness of global vs. local (per-row) rescaling strategies. As we will see, this variability depends strongly on the scaling factor s used in the data generation process.

In particular, we observe that:

- For small s, the α_u values are tightly concentrated around a common mean and are generally aligned with the global scaling factor α. This indicates that global rescaling is sufficient to capture the structure of X^*.
- For larger s, the spread of α_u widens. This increased heterogeneity implies that different users require distinct scaling adjustments, making a single global α inadequate. In some cases, the distribution of slopes is even noticeably shifted relative to both the global α and the average slope.

These observations highlight the need to compute reconstruction error in a row-wise fashion when working in high-confidence regimes (large s), where individual user dynamics deviate from the global structure.

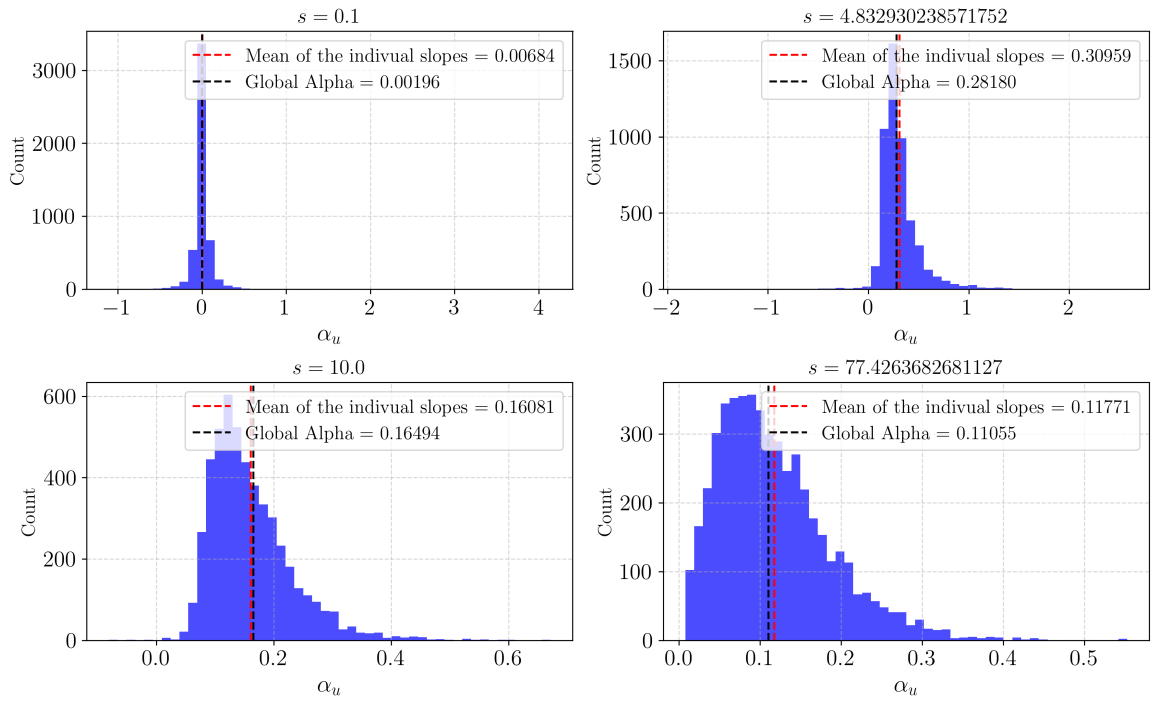
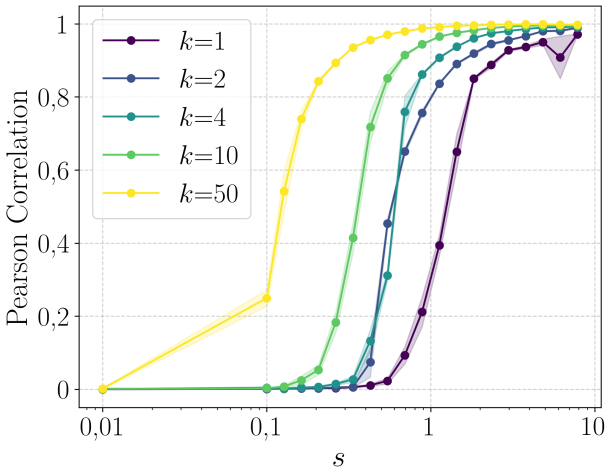


Figure 20: **Histogram of per-user optimal scaling coefficients α_u across four s values.** The black dashed line represents the global optimal scaling factor α computed over the full matrix, while the red dashed line marks the mean of individual per-user slopes. For small s, individual slopes are well concentrated around the global α. As s increases, we observe more variance in α_u, which explains why the global scaling becomes less effective. This highlights the need for row-wise adjustment in high-confidence regimes.

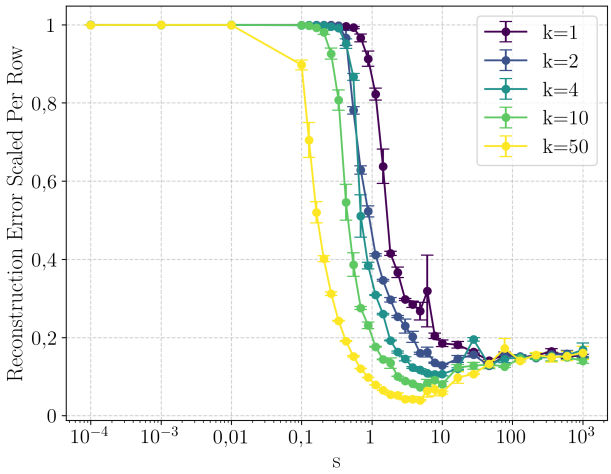
A.3 Analysis on the difference between Reconstruction Error scaled per row and Pearson Correlation

To better understand the limitations of traditional reconstruction metrics and to illustrate its difference with pearson correlation coefficient, we provide several additional plots below.

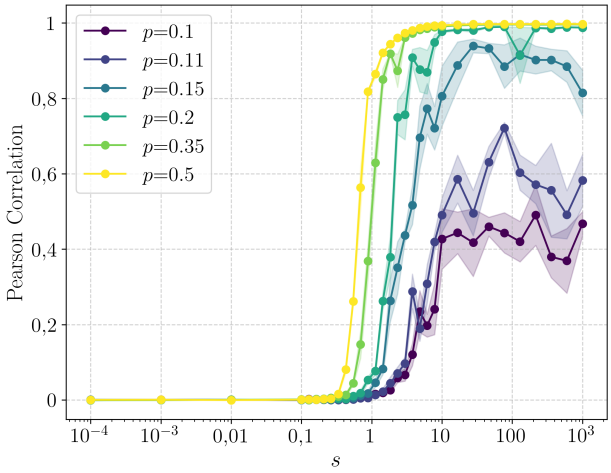
The Pearson correlation coefficient, shown in Figure 21a (as a second explanation for the section 6), and in Figure 21c (same as above on Figure 8b) approach 1 more rapidly than the row-normalized reconstruction error converges to 0. This difference arises from their distinct normalization schemes: while reconstruction error is sensitive to magnitude mismatches, the Pearson coefficient is scale-invariant and only measures alignment.



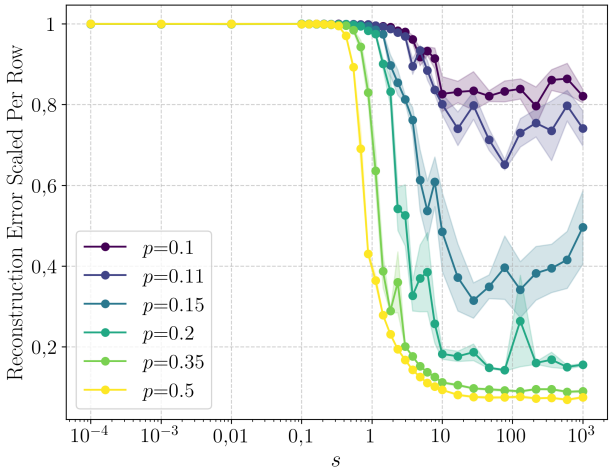
(a) **Pearson Correlation — Grouped by** k



(b) **Row-wise Scaled Error — Grouped by** k



(c) **Pearson Correlation — Grouped by** p



(d) **Row-wise Scaled Error — Grouped by** p

Figure 21: **Comparison Between Pearson Correlation and Row-wise Scaled Reconstruction Error as Functions of** s.
The plots show how well the model preserves user-specific preference structures as s increases. **Top row (a,b):** Grouped by k. Pearson correlation increases faster and saturates earlier than reconstruction error.
Bottom row (c,d): Grouped by p. The same pattern holds across sparsity levels — while correlation approaches 1, reconstruction error decreases more slowly. This highlights that Pearson is invariant to scale, while reconstruction error is not.

To further illustrate this phenomenon, we simulate the impact of a single outlier on both metrics. In this setting, we compute the Pearson correlation and reconstruction error between a reference vector x and a noised version y where only the last element is perturbed with increasing magnitude. The results are shown in Figure 22.

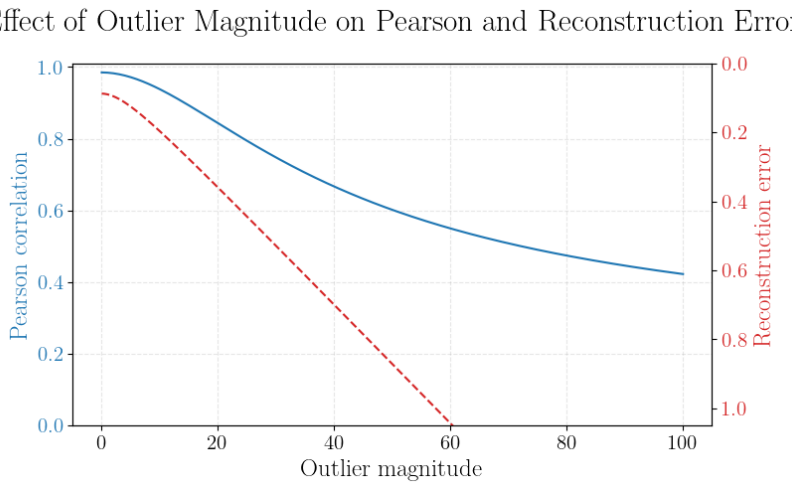


Figure 22: Effect of increasing outlier magnitude on Pearson correlation and reconstruction error. Pearson remains robust while reconstruction error quickly deteriorates.

The figure shows that even in the absence of outliers, the reconstruction error begins at a higher value than the Pearson correlation, indicating greater sensitivity to small deviations. As the magnitude of the outlier increases, the reconstruction error sharply deteriorates, whereas the Pearson correlation drops more slowly. This confirms that Pearson is less sensitive to localized noise or extreme values, and can remain high even when absolute predictions are incorrect — as long as the relative ordering remains intact.

This robustness explains why, in earlier plots, the Pearson coefficient converges faster than the reconstruction error. It highlights that models can still capture meaningful preference structures (in terms of rankings) even if the predicted magnitudes are misaligned.

The Spearman coefficient follows a similar trend as the Pearson coefficient. Together, these metrics confirm that increasing k leads to better identification of both the ranking and intensity of preferences, with consistent improvements in alignment and reconstruction.

Figure 23 provides qualitative insight into the alignment between predicted rows $UV_u^\top$ and their ground-truth counterparts X_u^*, for increasing values of the scaling factor s. Each plot represents a single user's predicted and true item scores, sorted by the ground-truth vector X_u^* for clarity.

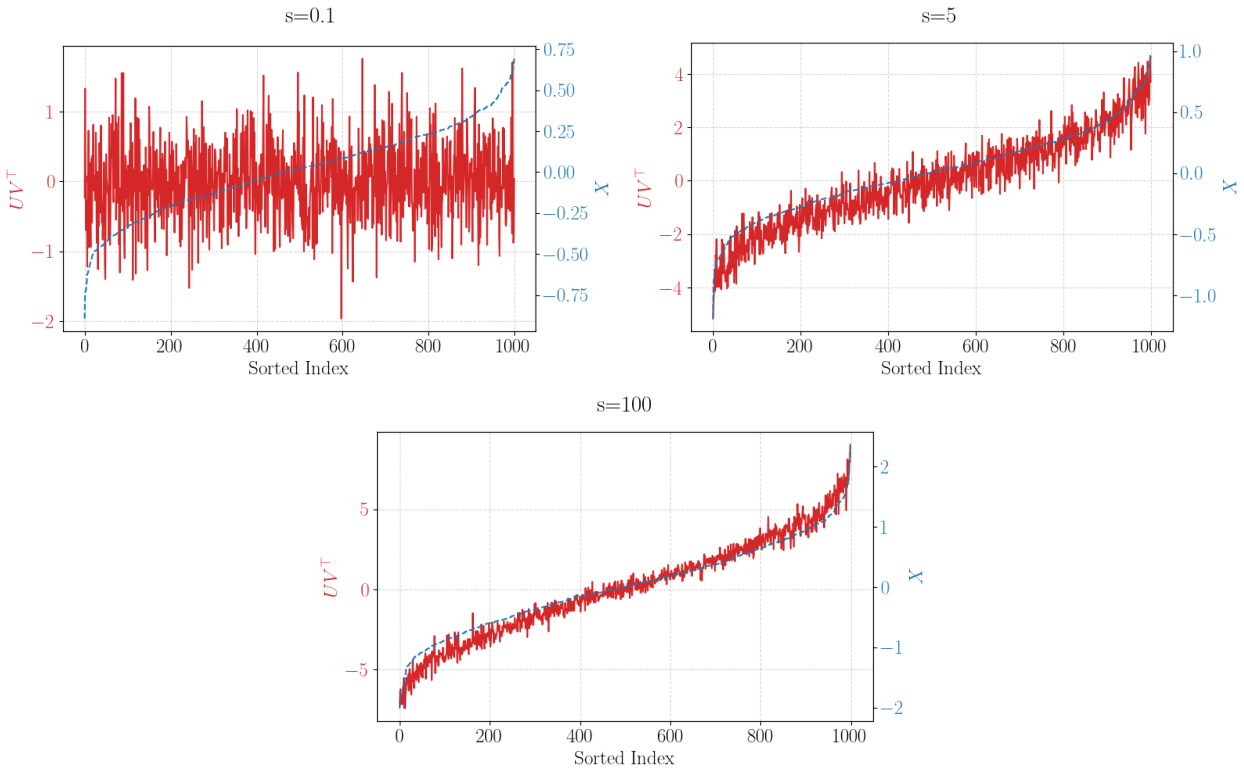


Figure 23: **Example row-level predictions: $UV^\top$ vs X for different s values.** Each curve shows a predicted vector compared to the true X vector, sorted by ground-truth values.

For very small values of s (e.g., $s = 0.1$), the learned signal appears almost indistinguishable from noise. The predicted curve does not exhibit any meaningful structure, indicating that the model fails to recover user preferences under such weak signal conditions.

As s increases (e.g., $s = 5$), the predicted values start to follow the shape of the ground-truth more closely. The signal becomes clearer, and the model begins to learn consistent item-level preferences. The predicted curve becomes smoother and more aligned with the ground-truth ordering.

At high scaling levels (e.g., $s = 100$), the predicted structure is sharply aligned with the target, suggesting that the preference model becomes highly confident. However, we still observe a systematic **scaling offset**: although the shape is correct, the predicted magnitudes of $UV^\top$ are not always well aligned with X^*. This persistent bias likely explains why the Pearson correlation coefficient — which is scale-invariant — remains high, while the reconstruction error remains suboptimal, especially when computed globally.

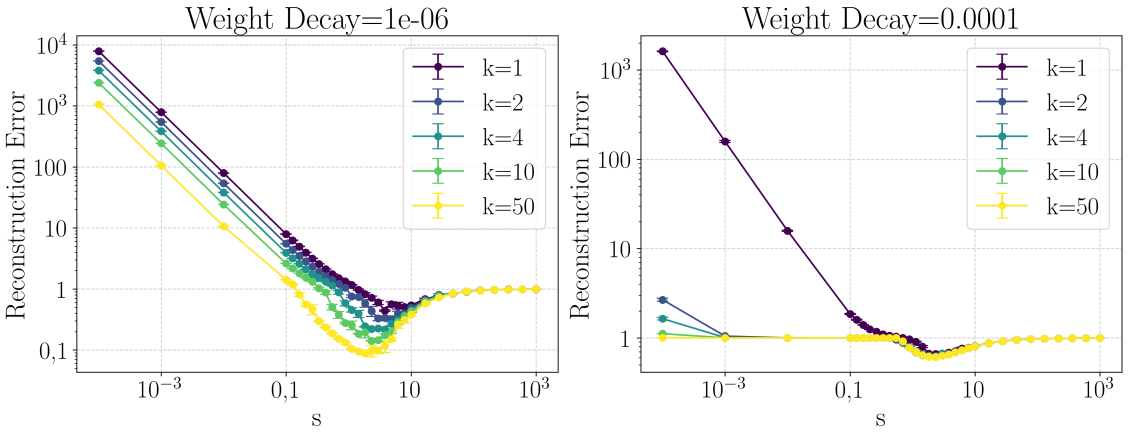
This visualization highlights that while preference directions are correctly learned as s increases, the model still struggles to match the true score scale, reinforcing the need for row-wise normalization or scaling-aware evaluation metrics.

A.4 Extended Analysis: Weight Decay vs. Scaling

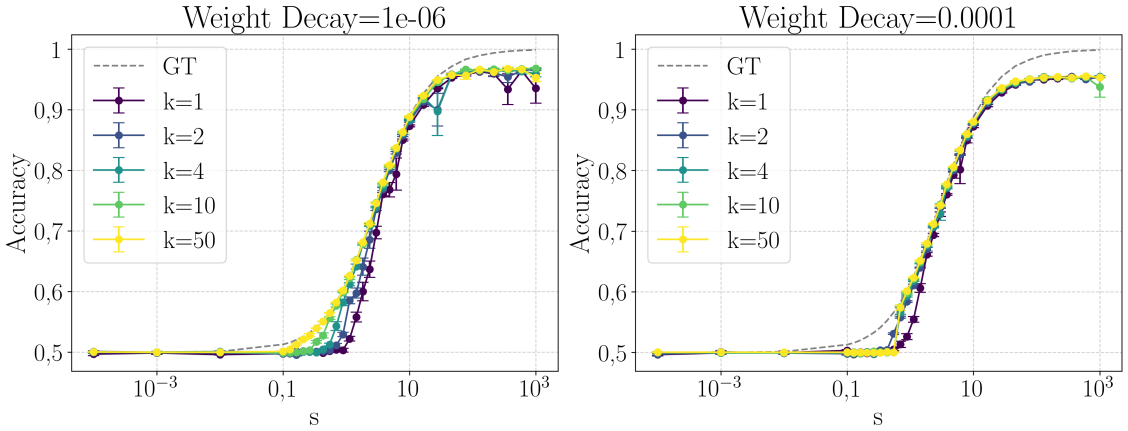
This appendix expands on the observations introduced in Section *Role of Weight Decay and Scaling* (see Section 6 for context), where we highlighted that the optimal amount of regularization depends both on the number of preference repetitions k and the confidence scaling s.

Here, we analyze how both reconstruction error and accuracy evolve across different values

of k when varying the weight decay. This deeper look helps explain why the optimal value of w_d decreases as k increases, and how it affects both under- and over-regularization in different signal regimes.



(a) Reconstruction Error grouped by k for different w_d



(b) Accuracy grouped by k for different w_d

Figure 24: Effect of weight decay across different k values. Increasing k requires less regularization, especially for large s values.

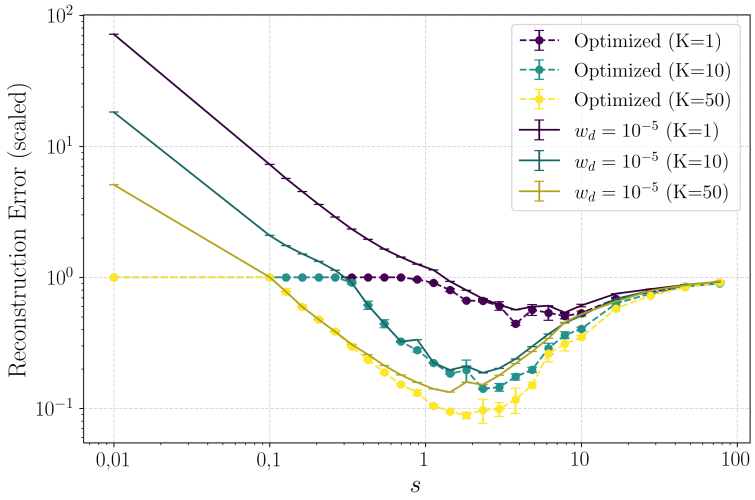


Figure 25: Comparison of reconstruction error across s for fixed vs. optimized weight decay, grouped by k.

A.5 Scaling Factor and α Coefficient

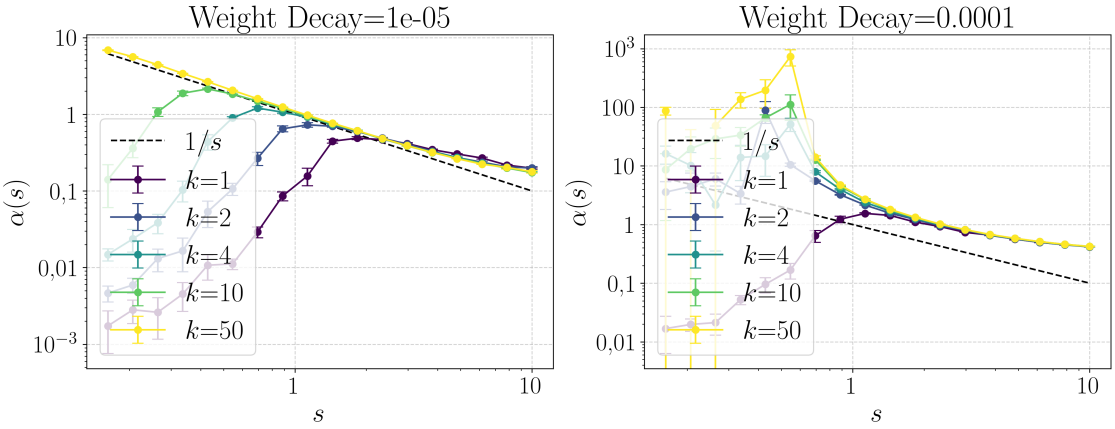
This section complements the discussion from Section 6, where we analyzed the effect of s and weight decay on reconstruction error. Here, we further investigate how well the learned matrix $UV^\top$ scales with respect to the ground truth by studying the evolution of the optimal alignment coefficient α.

In addition to direct error metrics, we analyze the behavior of the scaling coefficient α, defined as:

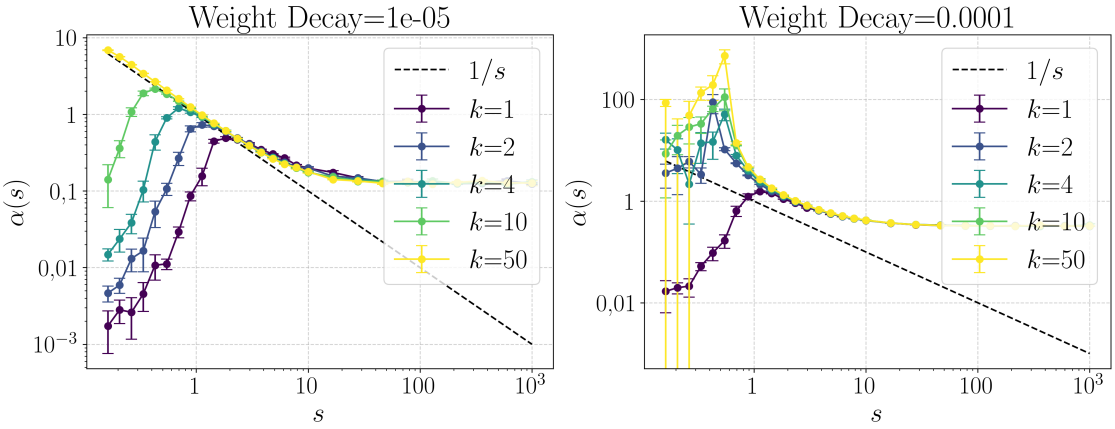
$$\alpha = \frac{\langle UV^\top, X \rangle}{\|UV^\top\|_F^2}$$

which corresponds to the optimal multiplicative scalar aligning the learned reconstruction $UV^\top$ with the ground-truth matrix X. Figure 26a and its full-range counterpart 26b show that, for moderate values of s, the scaling coefficient α behaves approximately like $1/s$ around the minimum of the reconstruction error. This approximation degrades when s becomes too small or too large, especially under strong regularization. These deviations highlight the importance of properly tuning both the scaling parameter s and the weight decay to achieve accurate reconstruction.

Moreover, we observe that for larger values of k, the alignment between α and $1/s$ becomes tighter. This suggests that increasing the number of comparisons improves the model's ability to linearly rescale the ground-truth matrix.



(a) Zoomed-in: α vs s



(b) α vs s over wider range

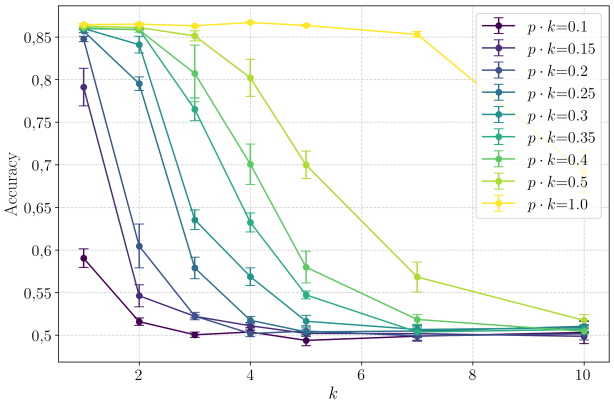
Figure 26: Scaling coefficient α as a function of s, for various values of k. The dotted line represents $1/s$.

A.6 Constant Label Budget: Effect of k under Fixed $p \cdot k$

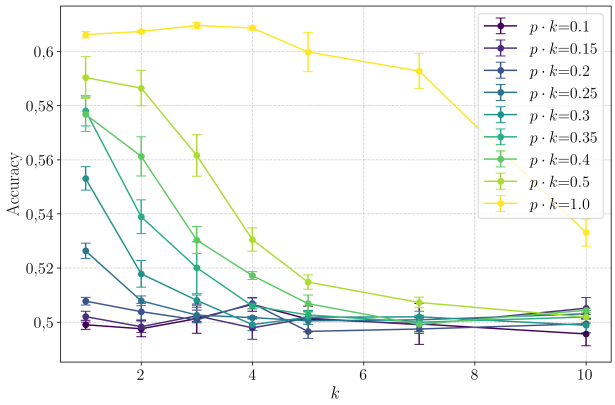
This section expands on the analysis presented in Section 7, where we explored the joint impact of sparsity p and repetition k. We now isolate the effect of k by fixing the product $p \cdot k$, which serves as a proxy for the total amount of supervision available.

Our goal is to assess whether a constant label budget leads to a stable optimal number of comparisons per user (k), especially under different values of the scaling factor s. Figures 27 and 28 below summarize the accuracy and reconstruction error for fixed $p \cdot k$, across several scaling settings. As we can see, the validation accuracy does not exhibit a clear maximum across varying values of k, even when the product $p \cdot k$ remains constant. This holds across all tested scaling factors: whether we look at high scaling (a), low scaling (b), or intermediate scaling (c).

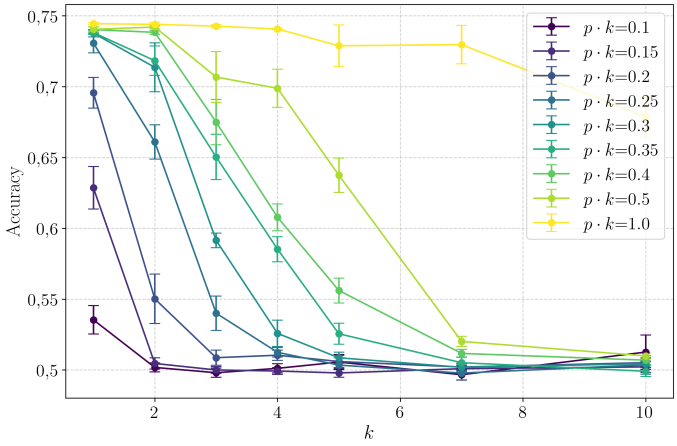
A similar pattern can be observed in the reconstruction error plots (Figure 28). Across all subfigures—(a), (b), and (c)—there is no evident value of k that systematically minimizes the reconstruction error.



(a) Validation accuracy for $s = 8$

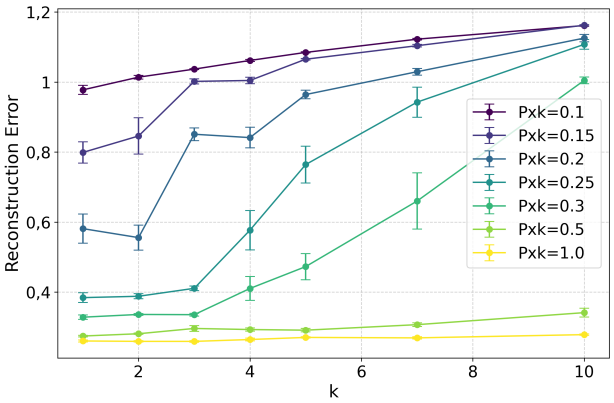


(b) Validation accuracy for $s = 1$

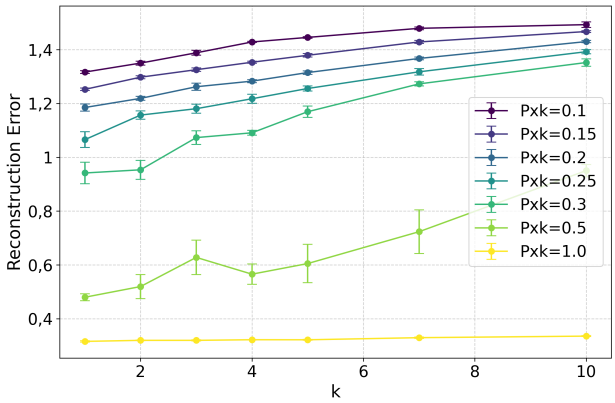


(c) Validation accuracy for $s = 3$

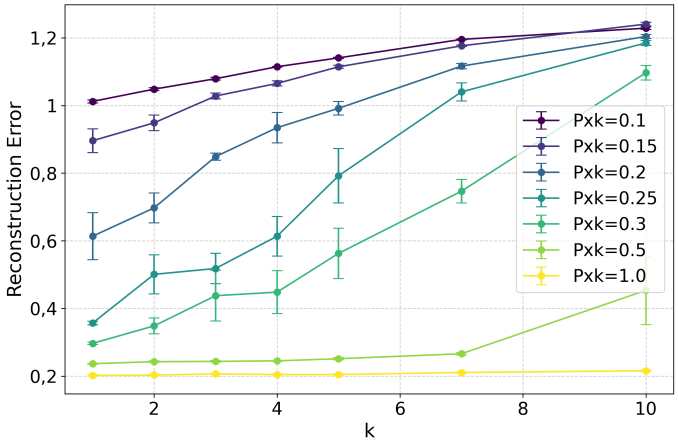
Figure 27: Validation accuracy as a function of the number of comparisons k, grouped by the product $p \cdot k$ to ensure constant label density.
Each subfigure shows a different scaling factor s: (a) $s = 8$, (b) $s = 1$, (c) $s = 3$.
Fixed parameters: $n = m = 1000$, $d = 2$, $l_r = 10^{-3}$, $w_d = 10^{-5}$, $reps = 5$.



(a) Reconstruction error for $s = 8$



(b) Reconstruction error for $s = 1$



(c) Reconstruction error for $s = 3$

Figure 28: Reconstruction error as a function of the number of comparisons k, grouped by constant information density $p \cdot k$.
Subfigures vary the scaling factor s: (a) $s = 8$, (b) $s = 1$, (c) $s = 3$.
Fixed parameters: $n = m = 1000$, $d = 2$, $l_r = 10^{-3}$, $w_d = 10^{-5}$, $reps = 5$.

References

[1] Yehuda Koren, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems". In: *Computer* 42.8 (2009), pp. 30–37.

[2] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[3] Steffen Rendle et al. *BPR: Bayesian Personalized Ranking from Implicit Feedback*. 2012. arXiv: 1205.2618 [cs.IR]. URL: https://arxiv.org/abs/1205.2618.

[4] Dohyung Park et al. *Preference Completion: Large-scale Collaborative Ranking from Pairwise Comparisons*. 2015. arXiv: 1507.04457 [stat.ML]. URL: https://arxiv.org/abs/1507.04457.