


Systems biology

# Boost-RS: boosted embeddings for recommender systems and its application to enzyme–substrate interaction prediction

Xinmeng Li<sup>1</sup>, Li-Ping Liu<sup>1,\*</sup> and Soha Hassoun <sup>1,2,\*</sup>

<sup>1</sup>Department of Computer Science, Tufts University, Medford, MA 02155, USA and <sup>2</sup>Department of Chemical and Biological Engineering, Tufts University, Medford, MA 02155, USA

\*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on September 21, 2021; revised on February 6, 2022; editorial decision on March 26, 2022

## Abstract

**Motivation:** Despite experimental and curation efforts, the extent of enzyme promiscuity on substrates continues to be largely unexplored and under documented. Providing computational tools for the exploration of the enzyme–substrate interaction space can expedite experimentation and benefit applications such as constructing synthesis pathways for novel biomolecules, identifying products of metabolism on ingested compounds, and elucidating xenobiotic metabolism. Recommender systems (RS), which are currently unexplored for the enzyme–substrate interaction prediction problem, can be utilized to provide enzyme recommendations for substrates, and vice versa. The performance of Collaborative-Filtering (CF) RSs; however, hinges on the quality of embedding vectors of users and items (enzymes and substrates in our case). Importantly, enhancing CF embeddings with heterogeneous auxiliary data, specially relational data (e.g. hierarchical, pairwise or groupings), remains a challenge.

**Results:** We propose an innovative general RS framework, termed Boost-RS that enhances RS performance by ‘boosting’ embedding vectors through auxiliary data. Specifically, Boost-RS is trained and dynamically tuned on multiple relevant auxiliary learning tasks Boost-RS utilizes contrastive learning tasks to exploit relational data. To show the efficacy of Boost-RS for the enzyme–substrate prediction interaction problem, we apply the Boost-RS framework to several baseline CF models. We show that each of our auxiliary tasks boosts learning of the embedding vectors, and that contrastive learning using Boost-RS outperforms attribute concatenation and multi-label learning. We also show that Boost-RS outperforms similarity-based models. Ablation studies and visualization of learned representations highlight the importance of using contrastive learning on some of the auxiliary data in boosting the embedding vectors.

**Availability and implementation:** A Python implementation for Boost-RS is provided at <https://github.com/HassounLab/Boost-RS>. The enzyme-substrate interaction data is available from the KEGG database (<https://www.genome.jp/kegg/>).

**Contact:** [liping.liu@tufts.edu](mailto:liping.liu@tufts.edu) and [soha.hassoun@tufts.edu](mailto:soha.hassoun@tufts.edu)

## 1 Introduction

Understanding the rich functionality of enzymes is fundamental in advancing biochemistry, molecular and synthetic biology and many other application domains. Enzymes were assumed *specific*, catalyzing a specific substrate; however, there is now wide consensus that enzymes are *promiscuous*, catalyzing many substrates, including substrates that the enzymes did not evolve to catalyze (Khersonsky and Tawfik, 2010). Our ability to analyze this inherent promiscuity has proved instrumental in guiding the direct evolution of novel proteins (Romero and Arnold, 2009), elucidating metabolism in natural and engineered organisms (Porokhin *et al.*, 2021), and creating novel synthesis pathway to produce valuable therapeutics and

commodity molecules (Bowie *et al.*, 2020). Despite progress in protein function annotation and modeling protein–ligand interactions (mostly focused on drug–ligand interactions), and manual and automated curation efforts, there remains large gaps in our knowledge of enzyme capabilities. Computational tools that predict enzyme promiscuity on molecules can augment existing knowledge, guide biological and biomedical applications and reduce costly experimental efforts.

Computational approaches for predicting enzyme–substrate interactions target different applications. Physics-based models, including molecular docking and molecular dynamic simulations, attempt to identify the most favorable binding mode of a ligand with a

given target protein. These methods require 3D models of both protein and molecule, and require significant compute time, making these methods suitable for detailed analysis of a small number of interactions. Rule-based methods predict site of metabolism or products of enzymatic transformations on a query molecule. Most such methods, however, utilize hand-curated biotransformation rules (e.g. Ridder and Wagener, 2008), or applicable to only specific enzymes (e.g. Tyzack and Kirchmair, 2019), thus limiting their general applicability. Machine-learning (ML) approaches have taken advantage of available enzymatic data and solve many important questions such the likelihood of enzymatic transformations between a compound pair, e.g. support vector machines (Kotera *et al.*, 2013), graph embedding (Jiang *et al.*, 2021), identifying enzyme commission numbers that act on molecules, e.g. using hierarchical classification of enzymes on molecules (Visani *et al.*, 2021), and predicting the likelihood of a sequence catalyzing a reaction or quantifying the affinity of sequences on substrates using Gaussian processes (Mellor *et al.*, 2016). Once trained, ML models provide quick evaluation and are suited for many bioengineering and biological applications that require the exploration of the vast interaction space.

To expand the use of ML in predicting enzyme–substrate interactions, we investigate the use of recommender systems (RSs) to recommend enzymes that are likely to act on specific substrates, and/or compounds that are suited as substrates for an enzyme. RSs are heavily utilized in industrial applications. For example, more than 50% of all AI training cycles at Facebook are devoted to training deep learning recommendation models (Acun *et al.*, 2021). RS, however, were not used prior for predicting enzyme–substrate interactions. Previously, RS were used for predicting protein–drug interactions (Bagherian *et al.*, 2021). Many such techniques use collaborative filtering (CF) in the form of matrix factorization (MF), e.g., Multiple Similarities Collaborative Matrix Factorization (MSCMF) (Zheng *et al.*, 2013), Probabilistic Matrix Factorization (Mnih and Salakhutdinov, 2008) and Neighborhood Regularized Logistic Matrix Factorization (NRLMF) (Liu *et al.*, 2016).

As the performance of CF hinges on learned embeddings of the users and items, prior RS techniques aimed to utilize auxiliary (side) data to learn improved embeddings. Many techniques enhance CF using similarities among proteins and among drugs, e.g. MSCMF (Zheng *et al.*, 2013) or neighborhood regularization, e.g. NRLMF (Liu *et al.*, 2016), and REMAP (Lim *et al.*, 2016), with the goal of minimizing distances between a protein (or a drug) and its nearest neighbors in the latent space. Other RS aim to integrate auxiliary data by fusing knowledge graphs (e.g. Wang *et al.*, 2021), or integrating multi-source data (e.g. Gao *et al.*, 2018; Zhu *et al.*, 2017). In practice, auxiliary data is complex and often exhibits multiple relational aspects: item labels may be hierarchical, and users may share a group label (zip code, building address or profession). The common practice to concatenate auxiliary data with the learned embedding does not necessarily maximally exploit the relational aspect of the data. A general methodology for computing enhanced embeddings based on relational data and other complex heterogeneous auxiliary data therefore remains a challenge.

We present in this article a novel technique, Boost-RS, for enhancing the performance of RS by ‘boosting’ the embedding vectors through auxiliary learning tasks. Boost-RS integrates the primary CF task with boosting tasks that aim to upgrade the embedding vectors based on available heterogeneous auxiliary data. The integration of user and item attributes addresses the interaction matrix sparsity issue and has already shown RS performance improvements (Bagherian *et al.*, 2021; Sun *et al.*, 2019). To minimize negative transfer from the auxiliary tasks to the main task, the CF and the boosting tasks are dynamically weighted (Liu *et al.*, 2019). Each auxiliary task is designed to maximally utilize the available auxiliary data. Importantly, to learn from relational data, Boost-RS uses contrastive learning, which contrasts positive and negative samples to learn discriminative representations in a self-supervised manner. Contrastive learning is applied through triplet loss (Weinberger and Saul, 2009), where the model is trained to produce representations such that, for a given anchor example, a positive example is closer to the anchor than a negative example.

We demonstrate Boost-RS’s effectiveness by applying it to the enzyme–substrate interaction prediction problem. Through multi-tasking, Boost-RS integrates the primary CF task with boosting tasks that aim to upgrade the embedding vectors based on available heterogeneous auxiliary data. For enzymes, we exploit the Enzyme Commission (EC) hierarchical relationships and the enzyme functional orthologs. For substrates, we utilize the molecular fingerprints and substrate–substrate biotransformation relationships due to functionally similar enzymes. We formulate a hierarchical loss on the EC relationship. We use contrastive learning on the enzyme functional orthologs and the biotransformation relationships. Our main contributions are:

- Creating a flexible and generalizable framework, Boost-RS, that enriches the embedding vectors for CF-based RSs via multi-tasking on individual and relational heterogeneous auxiliary data.
- Showing that applying multi-tasking on contrastive learning on relational data may outperform other techniques such as concatenation with learned embeddings and multi-labels learning.
- Demonstrating the generality of the Boost-RS framework by showcasing its applicability to three recent neural network baseline CFs: Deep Matrix Factorization (DMF) (Xue *et al.*, 2017), Neural Graph Collaborative Filtering (NGCF) (Wang *et al.*, 2019) and Neural Matrix Factorization (NMF) (He *et al.*, 2017).
- Showing that Boost-RS outperforms state-of-the-art similarity-based Graph Regularized Generalized Matrix Factorization (GRGMF) RSs (Zhang *et al.*, 2020).

## 2 Methods

### 2.1 Dataset

We apply Boost-RS to the enzyme–substrate interaction prediction task to recommend substrates to enzymes that are most likely to interact and vice versa. Our dataset is culled from biochemical reactions in the KEGG database (Kanehisa and Goto, 2000). As most biochemical reactions are reversible, no distinction is made between substrates and products, and hence interacting molecules are referred to as compounds or substrates interchangeably. Reactions form a bipartite graph that can be captured as a binary interaction matrix between enzymes and substrates (Fig. 1A), where each row represents a compound, and each column represents an enzyme. A matrix entry is set to 1 if the compound and enzyme participate the same reaction, therefore representing a positive interaction instance. An entry is set to 0 in case there is no catalogued enzyme–substrate interaction. Compounds that are common to many enzymatic reactions, including cofactors such as ATP and NADH and metals, are excluded from the matrix. To ensure valid data splits, enzymes or compounds with a single entry in the interaction matrix are also excluded. In total, 17 627 enzyme–compound interactions are collected. They involve 4768 enzymes and 6397 compounds. As all non-positive interactions are unknown, they are assumed as negative interactions and their corresponding matrix entries are set to zero. The interaction matrix is therefore sparse with 0.06% positive entries.

### 2.2 Auxiliary data

Four attributes are collected or derived from the KEGG database and are used for training auxiliary tasks:

**Enzyme Commission (EC) numbers.** Each enzyme is associated with an EC number (Webb, 1992) that comprises four numbers, separated by dots, starting with a number that broadly represents the enzyme class, then the sub-class, sub-subclass and a final number that reflects the specificity of the enzyme toward a small group of substrates. As the EC numbers are hierarchical, the auxiliary learning task of EC prediction can be formulated as such. The EC numbers have 7, 94 and 164 distinct labels in the first three respective

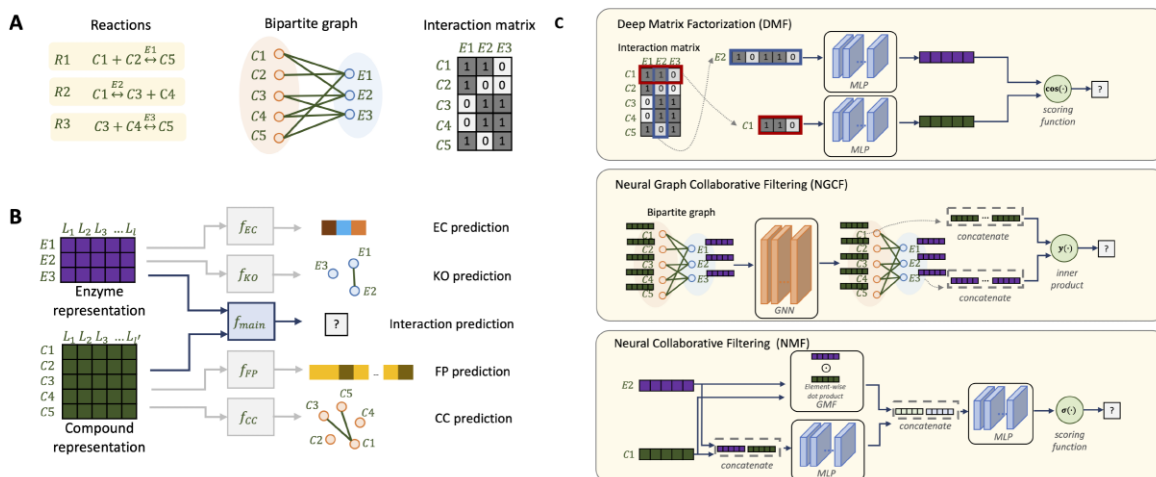


Fig. 1. Boost-RS framework for enzyme-substrate recommendation prediction. (A) Interaction matrix construction from enzymatic reactions. For example for  $E1$ , three positive interactions are added to the matrix. (B) The Boost-RS framework that integrates the main task of interaction prediction with related auxiliary tasks. (C) Collaborative filtering models used as baselines

fields of EC. We only consider the first three fields of EC, as the fourth index typically denotes specific substrates and cofactors.

**Functional Orthologs (KO) numbers.** Another enzyme attribute is its KEGG functional orthology (KO) number (Kanehisa et al., 2016). A particular KO designation, the letter K followed by five numerical digits, is assigned to a group of genes sharing similar functionality. KO numbers can therefore be considered as a ‘group’ attribute. We utilize 5575 sets of KO designations.

**Molecular fingerprints (FP).** Based on descriptions for molecules in the KEGG database, molecular attributes in the form of MACCS fingerprints (FP) (Durant et al., 2002) are calculated using RDKit.

**Compound-compound biotransformations (CC).** The KEGG database provides biotransformation patterns, designated as RClasses, shared by multiple substrate-product or compound-compound (CC), pairs. CC pairs under the same RClass are transformed by enzymes with similar functionality (e.g. hydroxylation or methylation). CC relationships give rise to a compound-centric graph, akin to a social network. This graph is *not* a similarity network as CC pairs are not necessarily similar: a compound may undergo significant molecular changes under some enzymatic transformations (e.g. transferases, ligases).

## 2.3 The Boost-RS model

### 2.3.1 Main task: interaction prediction

The main task of the Boost-RS framework (Fig. 1B) is ‘recommending’ compounds to enzymes (or enzymes to compound). Per the interaction matrix, an entry  $y_{ij}$  is 1 for the positive interaction set,  $P$  and  $y_{ij}$  is 0 for the negative interaction set,  $N$ . As in a standard recommendation task, the main task function,  $f_{main}(\cdot, \cdot)$ , predicts the probability  $\hat{y}_{ij}$  of interaction using learned enzyme and compound representations. We denote the representations for compound  $i$  and enzyme  $j$  as  $\mathbf{v}_i$  and  $\mathbf{u}_j$ , respectively. The probability of interaction,  $\hat{y}_{ij}$ , and task loss,  $\mathcal{L}_{main}$ , are then defined as:

$$\hat{y}_{ij} = f_{main}(\mathbf{v}_i, \mathbf{u}_j), \quad (1)$$

$$\mathcal{L}_{main} = \sum_{i,j} BCE(\hat{y}_{ij}, y_{ij}). \quad (2)$$

In the simplest form,  $f_{main}(\mathbf{v}_i, \mathbf{u}_j) = \mathbf{v}_i^T \mathbf{u}_j$ . In more complex forms,  $f_{main}(\mathbf{v}_i, \mathbf{u}_j)$  is calculated using neural networks. The parameters for  $f_{main}$  and the representations of enzymes and compounds are updated by minimizing the task loss,  $\mathcal{L}_{main}$ . One such loss is Binary Cross Entropy (BCE) which measures the difference between the actual interaction value and the predicted value. Some base CF models (Section 2.3.4) improves over this loss. The embedding vectors,  $\mathbf{v}_i$

and  $\mathbf{u}_j$ , are critically important for the recommendation task. To boost performance, we inject task-relevant auxiliary data for compounds and enzymes into the embedding vectors via multi-task learning.

### 2.3.2 Auxiliary tasks

Each auxiliary task calculates attribute probabilities with an auxiliary task function and updates representations and parameters based on the task loss. To address the differing characteristic in relational attributes, auxiliary losses are defined on individual, hierarchical, group and pairwise attributes. While we describe the details relevant to the specific substrate and enzyme attributes, the framework easily accommodates other attributes with individual and relational attributes.

Using FP as an individual attribute, we denote the function for FP prediction as  $f_{FP}$ . As each compound fingerprint is a binary vector, we use BCE to evaluate the prediction accuracy for each vector entry. The FP prediction function and the FP task loss are defined as:

$$\hat{y}_i^{FP} = f_{FP}(\mathbf{v}_i), \quad (3)$$

$$\mathcal{L}_{FP} = \frac{1}{|C|} * \frac{1}{l_{FP}} \sum_{i \in C} BCE(\hat{y}_i^{FP}, y_i^{FP}), \quad (4)$$

where  $C$  is the set of compounds, and  $l_{FP}$  is the length of the fingerprint vector.

For EC prediction, we capitalize on the EC’s hierarchical structure, and denote the function for EC prediction as  $f_{EC}$ . The loss for each enzyme is based on the cross entropy loss on each of its field set,  $FS_1, FS_2, FS_3$  for the first, the first two and the first three fields of the EC number, respectively. The prediction function and the cumulative task loss for the EC attribute is therefore:

$$\hat{y}_{j,FS_k}^{EC} = f_{EC}(\mathbf{u}_j), \quad (5)$$

$$\mathcal{L}_{EC} = \frac{1}{|E|} \sum_{j \in E} \sum_{k=1}^3 w_k \cdot CE(\hat{y}_{j,FS_k}^{EC}, y_{j,FS_k}^{EC}), \quad (6)$$

where  $E$  is the set of enzymes, and  $k$  is indicating which field set,  $FS$ , in EC are we calculating the cross entropy loss on its distinct labels. And  $w_k$  is the weight for the  $FS_k$  field set in EC, and, after some tuning, is set to  $\frac{1}{3}$ , therefore equalizing each field’s contribution to the loss.

When using group attributes and pairwise relationship for the auxiliary tasks, we denote functions for KO prediction and CC prediction as  $f_{\text{KO}}$  and  $f_{\text{CC}}$ , respectively. We utilize triplet loss, with the intention of pulling the representation of samples in the same set closer together in the embedding space and pushing away the representation of a sample outside the set. A function,  $d(\cdot)$ , measures the distance between a pair of representations. The CC loss function is defined as:

$$\mathcal{L}_{\text{CC}} = \frac{1}{|C|} \sum_{i, P_i, N_i \in C} \max(0, d(\mathbf{v}_i, \mathbf{v}_{P_i}) - d(\mathbf{v}_i, \mathbf{v}_{N_i}) + \gamma), \quad (7)$$

where  $i$  an anchor compound,  $P_i$  is a compound in the set of compounds has CC pairwise relationship with compound  $i$ ,  $N_i$  is a compound in a set of compounds that does not have an CC relationship with compound  $i$ , and  $\gamma$  is a positive margin between the positive  $P_i$  and negative  $N_i$  samples.

The KO loss function is defined similarly to the CC loss function, except that the anchor, positive and negative samples are derived from the KO group relationships defined on the enzymes. The contrastive loss on CC reflect a pairwise relationship within the compound-centric relational graph, while the loss on KO reflects the group label distinction.

### 2.3.3 Dynamic training

The Boost-RS loss,  $\mathcal{L}_{\text{Boost-RS}}$ , is the weighted sum of losses of the main task and auxiliary tasks:

$$\mathcal{L}_{\text{Boost-RS}} = \alpha_{\text{main}} \mathcal{L}_{\text{main}} + \alpha_{\text{aux}} \mathcal{L}_{\text{aux}}, \quad (8)$$

where  $L_{\text{aux}}$  is the additive losses across the auxiliary tasks,  $\alpha_{\text{main}}$  and  $\alpha_{\text{aux}}$  are weights for the main task and auxiliary tasks, respectively.

The weights of task losses directly influence the performance of interaction prediction. We use a dynamic strategy to balance the weights among the main task and auxiliary tasks. The abridged linear schedule (Belharbi *et al.*, 2016) emphasizes auxiliary tasks in the early training epochs and shifts the focus to the main task in later epochs. We assign weights for the main and auxiliary tasks losses as follows:

$$\alpha_{\text{main}} = \min\left(\frac{t}{T}, 1\right); \alpha_{\text{aux}} = \max(1 - \alpha_{\text{main}}, 0), \quad (9)$$

where  $t$  is the current training iteration, and  $T$  is time point where the focus shifts completely to the main task.

### 2.3.4 Deep-learning baseline RSs

We use three neural-network RSs (Fig. 1C) for the interaction prediction task: DMF (Xue *et al.*, 2017), NGCF (Wang *et al.*, 2019) and NMF (He *et al.*, 2017). Each RS has its own characteristic and may be better suited for some applications. We apply Boost-RS to each of these models.

The inputs to DMF are the rows and columns of the interaction matrix. Two separate Multi-layer Perceptron (MLP) networks are trained through  $f_{\text{main}}$  to learn compound and enzyme representations. The similarity of the enzyme and compound representations are computed using cosine similarity and outputted as the probability of the interaction,  $\hat{y}$ . Normalized cross entropy loss is used to compute the loss between  $y$  and  $\hat{y}$ .

To compute  $f_{\text{main}}$ , NGCF first applies graph neural network to the bipartite interaction graph. Graph neural networks (GNNs) are utilized to learn node representations. GNNs can account for different order neighbors including first order neighbors, second order neighbors and so on. Node representations for enzyme (compound) nodes that are learned for each level of neighbors are then concatenated. The inner product of the enzyme and compound representations is then computed. To favor assigning higher predictions for observed interactions than for unobserved interactions, NGCF utilizes the Pairwise Bayesian Personalized Ranking loss (not shown in figure).

For NMF,  $f_{\text{main}}$  has GMF and MLP working in parallel and are then followed with a scoring layer. GMF and MLP each learn independent representations. A function  $\sigma$  calculates the interaction probability based on the concatenation of the learned representations. BCE loss is used to compute the loss between  $y$  and  $\hat{y}$ .

## 3 Results

### 3.1 Experimental setup

We divide positive interactions into training, validation and test sets at a ratio of 7:2:1. During training and validation, negative interactions are randomly sampled from the unknown interactions at a negative sampling ratio, which is a hyperparameter that varies across models. During testing, all unknown interactions are assumed negative. Positive interactions in the training set are excluded during sampling.

As the test dataset is imbalanced, where the ratio of positive to the assumed negative interactions is less than 0.01%, evaluation metrics are selected to reflect the ability of RS to rank positive interactions ahead of negative ones. Average precision (AP) computes the average precision after each predicted positive interaction in the ranked order list provided by RS. AP is utilized for model selection. To place lower emphasis on the exact ranking of known interactions, R-Precision computes the precision after all R positive interactions have been identified in the ranked order list. The overall performance at distinguishing between positive and negative interactions is reported using the area under the receiver operating characteristic curve (AUC). We additionally report the Mean AP (MAP) and the R-Precision across the enzymes and the substrates. As each enzyme and substrate had a varied number of positives, we also report the MAP for the top 3 items (MAP@3) and the precision on the top one item (Precision@1).

For the three baseline RSs, we follow the authors' guidelines on hyperparameter tuning. The range of hyperparameter search is specified as follows. The negative sampling ratio for training set is selected from {1, 5, 10, 15, 20, 25, 30}. The margin in the triplet loss,  $\gamma$ , is selected from {0.5, 1.0, 1.5}. The dimension of the embedding is selected from {128, 256, 512, 1024} based on average precision, where the optimal dimension is 256 for Boost-RS. The dimension of the two hidden layers of MLPs of  $f_{\text{main}}(\cdot)$ ,  $f_{\text{FP}}(\cdot)$ ,  $f_{\text{EC}}(\cdot)$  predictor is selected from {128, 256, 512}. We optimize our models with the Adam optimizer (Kingma and Ba, 2015) with learning rates selected among  $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ . We apply dropout at a rate selected from {0.0, 0.3, 0.5} and L2 norm at a weight selected from  $\{10^{-2}, 10^{-3}, \dots, 10^{-6}\}$ . For the abridged linear dynamic weighting strategy, we allow a maximum of 3000 iterations, with  $T=2000$ . During the first 2000 iterations, the model shifts linearly from training the auxiliary tasks to the interaction prediction task. Training is stopped early if there is no improved MAP on the validation set in 500 consecutive iterations.

### 3.2 Evaluating Boost-RS on baseline models

We evaluate the performance gain (Table 1A) when implementing Boost-RS for the three baselines Boost-RS significantly boosts the performance of every baseline across all metrics. NMF, which is the best performing baseline, gains 74%, 68% and 10% on MAP, R-Precision and AUC, respectively, when combined with Boost-RS. For this work, we use Boost-NMF as our 'Boost-RS' model and use NMF as a baseline model for the rest of the experiments, unless noted otherwise.

### 3.3 Multi-label learning via multi-tasking or concatenation

We create a compound (enzyme) binary multi-label vector for CC (KO), where each entry of the vector indicates the presence or absence of an RClass (KO) designation. The length of the vector is the number of distinct CCs (KOs). For CC, the multi-label vector has 3163 entries, where each compound is involved on average with 1.03 distinct CCs. For KO, the multi-label vector has a length of



**Table 1.** Interaction prediction performance evaluation. Boost-RS performance is bolded.

	Overall			Enzymes				Compounds			
	AP	R-Precision	AUC	MAP	R-Precision	MAP@3	Precision@1	MAP	R-Precision	MAP@3	Precision@1
A. Baselines and their boosted models											
DMF	0.154	0.344	0.869	0.328	0.251	0.261	0.255	0.281	0.282	0.334	0.332
Boost-DMF	0.192	0.401	0.954	0.374	0.258	0.273	0.294	0.309	0.296	0.362	0.374
NGCF	0.169	0.328	0.810	0.333	0.274	0.278	0.261	0.277	0.297	0.347	0.326
Boost-NGCF	0.223	0.503	0.959	0.552	0.295	0.446	0.393	0.405	0.488	0.566	0.490
NMF	0.280	0.380	0.880	0.339	0.322	0.286	0.309	0.332	0.320	0.362	0.378
<b>Boost-RS (Boost-NMF)</b>	<b>0.488</b>	<b>0.638</b>	<b>0.968</b>	<b>0.595</b>	<b>0.510</b>	<b>0.506</b>	<b>0.545</b>	<b>0.568</b>	<b>0.546</b>	<b>0.620</b>	<b>0.637</b>
B. Group data treated as individual attributes and incorporated into RS via either multi-tasking or concatenation											
Boost-RS_Multi-label	0.404	0.492	0.936	0.419	0.411	0.353	0.421	0.452	0.395	0.441	0.490
NMF-Concat_Multi-label	0.396	0.485	0.950	0.430	0.406	0.363	0.413	0.441	0.408	0.454	0.480
C. Interaction prediction with each auxiliary task using Boost-RS											
Boost-RS(KO)	0.296	0.377	0.857	0.321	0.315	0.280	0.321	0.349	0.319	0.347	0.381
Boost-RS(FP)	0.309	0.402	0.914	0.370	0.333	0.307	0.327	0.350	0.340	0.388	0.395
Boost-RS(EC)	0.344	0.432	0.880	0.337	0.377	0.294	0.372	0.399	0.325	0.364	0.438
Boost-RS(CC)	0.419	0.548	0.936	0.527	0.447	0.447	0.467	0.492	0.487	0.553	0.546
D. Interaction prediction with each auxiliary data using NMF-Concat_Multi-label											
NMF-Concat(KO)	0.285	0.386	0.872	0.346	0.327	0.296	0.319	0.343	0.337	0.375	0.384
NMF-Concat(FP)	0.287	0.386	0.870	0.338	0.329	0.286	0.318	0.344	0.324	0.368	0.386
NMF-Concat(EC)	0.292	0.390	0.879	0.339	0.333	0.289	0.324	0.349	0.320	0.361	0.392
NMF-Concat(CC)	0.322	0.408	0.868	0.351	0.343	0.297	0.351	0.372	0.342	0.380	0.412
E. Interaction prediction comparing Boost-RS framework against similarity-based method											
GRGMF(FP+EC)	0.189	0.407	0.946	0.362	0.282	0.266	0.293	0.307	0.281	0.361	0.387
Boost-RS(FP+EC)	0.349	0.456	0.931	0.376	0.377	0.314	0.385	0.408	0.352	0.398	0.453

Note: The best model (Boost-RS) is based on NMF and it exploits auxiliary data via multi-task learning, including hierarchical learning on EC, individual attribute learning on FP and contrastive viewing of KO and CC.

5575 entries, where each enzyme has on average 1.37 distinct KO designations.

To evaluate the use of contrastive loss in Boost-RS, we construct a model (Boost-RS\_Multi-label) that uses weighted BCE loss to predict multi-label vectors. Both Boost-RS and Boost-RS\_Multi-label therefore predict the KO and CC labels using multi-tasking, but with different loss formulations. To further assess the value of Boost-RS’s multi-tasking abilities on the KO and CC labels, we construct another model (NMF-Concat\_Multi-label), where we concatenate the GMF and MLP outputs of NMF with the outputs of MLP layers that encode the KO and CC data. The comparisons (Table 1B) utilize multi-label KO and CC data along with hierarchical EC and FP attributes. Boost-RS outperforms both models. That is, using contrastive loss with multi-tasking is the better strategy. Boost-RS outperforms Boost-RS\_Multi-label as contrastive learning explicitly enforces negative pairs to have distinct representations, while multi-label training does not. Both Boost-RS and Boost-RS\_Multi-label outperform NMF-Concat\_Multi-label may be due to the sparsity of the KO and CC multi-label vectors, where most vector entries representing the KO and CC labels are zero. Importantly, a flexible framework such as Boost-RS allows the judicious selection of the appropriate losses to boost the embeddings (last row of Table 1A).

We use t-SNE (Van der Maaten and Hinton, 2008) to visualize learned enzyme and compound representations (Fig. 2) using the various techniques (NMF, NMF-Concat\_Multi-label, Boost-RS\_Multi-label and Boost-RS). Enzyme representations are shown to the left of each sub-panel. Compound representations are shown to the right of each sub-panel, where an edge is added between two compound representations if the two compounds are related via a CC. For the enzyme representations, each dot is colored with an EC class. Enzymes in sub-panels C and D form the most distinguishable clusters when compared with sub-panels A and B as both multi-label and contrastive learning perform well on KO (see next section). Across the sub-panels, compound representations initially show no evident pattern (sub-panel A), but display more defined clusters with the progression toward sub-panel D. For the compound

representation (right figure of the D sub-panel), there is a node grouping for compounds that lack a CC relation (on the left side) as evident by the absence of any edges connecting these compounds.

### 3.4 Contributions of individual auxiliary tasks

We characterize the contribution of each auxiliary task to the performance of Boost-RS independently of other tasks (Table 1C). We also contrast each such contribution against using the same data via concatenation with the baseline NMF model (Table 1D). For both Boost-RS and NMF-Concat, each auxiliary task contributes positively to predicting the overall interactions, indicating that these auxiliary tasks are relevant to the main task and provide additional information beyond what is captured within CF. For Boost-RS, CC contributes the most (50% improvement on the overall AP), while KO, FP and EC show more modest improvements (6%, 10% and 23% respectively). For NMF-Concat, CC improves the baseline NMF by 15% AP, while KO, FP and EC show limited improvements (2%, 3% and 4%, respectively). Boost-RS(CC) improves enzyme and compound MAPs significantly (enzyme MAP by 55% and compound MAP by 48%). These results indicate that each auxiliary data improves the learning of compound and enzyme representations.

Boost-RS consistently improves performance on each tasks over NMF-concat, with the exception of the KO prediction task on select metrics other than overall AP, Precision1 for enzymes and MAP for the compounds. Despite this performance variation and other experimental evaluations, learning KO using contrastive learning results in the best boosting performance (last row in Table 1A) as our model selection is based on AP. We attribute this varied performance to the characteristics of the KO auxiliary data. For KO, contrastive loss is applied on 779 paired relationships, where there are multiple pairwise relationships involving all pairs of enzymes under the same KO group label. In contrast, for CC, contrastive loss is applied to 7915 paired relationships derived from pairwise compound-compound transformations.

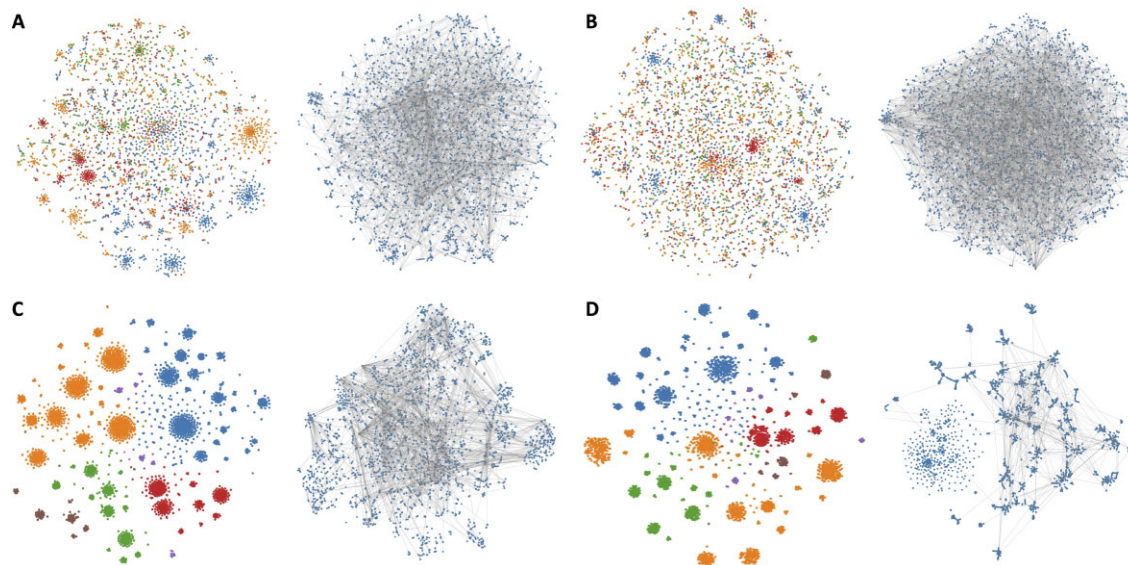


Fig. 2. Visualization using t-SNE for learned representation of enzymes and compounds, shown to the left and right of each sub-panel, respectively. (A) Baseline NMF. (B) Baseline with multi-label KO and CC concatenation (NMF-Concat\_Multi-label). (C) Boost-RS with the auxiliary task of learning multi-label KO and CC (Boost-RS\_Multi-label). (D) Boost-RS with triplet loss on KO and CC (Boost-RS)

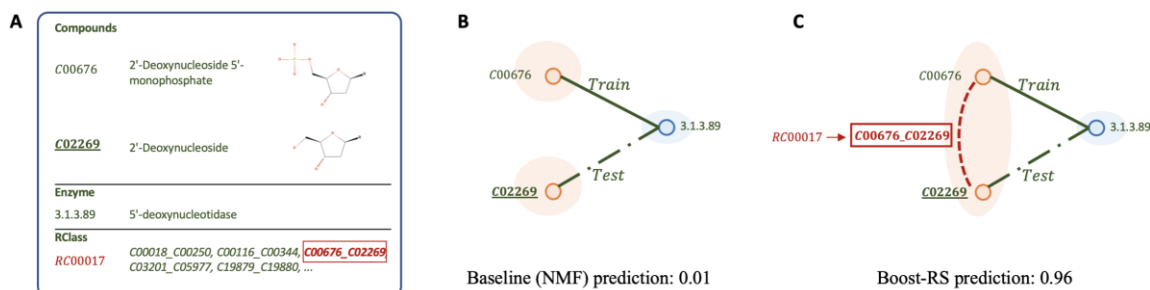


Fig. 3. Example that shows how Boost-RS exploits CC relationships derived from RClass relationships in KEGG. (A) Legend and KEGG data. RClass RC00017 is associated with multiple CC pairs, including C00676 and C02269. (B) NMF prediction is not aware of the relationships between C00676 and C02269, and results in a 0.01 likelihood of interaction between C02269 and enzyme 3.1.3.89. (C) Boost-RS exploits the CC relationships and results in an improved prediction

### 3.5 Boost-RS versus similarity-based models

We compare Boost-RS with a recent RS, Graph Regularized Generalized Matrix Factorization (GRGMF) (Zhang *et al.*, 2020). In addition to implementing MF, GRGMF learns latent node representations based on their neighborhood similarity. GRGMF therefore provides an alternative model for incorporating EC and FP auxiliary data. GRGMF takes as input a pairwise compound-similarity matrix and a pairwise enzyme-similarity matrix, where we use EC numbers and FP to obtain Jaccard similarity scores. CC and KO relationships cannot be readily integrated with GRGMF. We therefore evaluate Boost-RS when using only EC and FP as auxiliary data. Boost-RS outperforms GRGMF in most metrics, except for the AUC of GRGMF (Table 1E). The results show that the Boost-RS framework can effectively capture the auxiliary data.

### 3.6 Boost-RS exploits and advances biological knowledge

We present an example (Fig. 3) to highlight how Boost-RS utilizes biological knowledge to improve performance over NMF. When trained without regard to CC relationships, NMF cannot exploit such relationships and predicts low interaction probabilities between a compound and an enzyme in the test set. However, as Boost-RS exploits CC relationships, it predicts the interaction correctly. In the example, RClass RC00017 links many pairs of metabolites, including deoxynucleosides C00676 and C02269. The latter pairing is due to several reactions in the database (not shown in the figure).

Without integrating this CC pairing information into the RS, NMF predicts an interaction score of 0.01. In contrast, Boost-RS predicts a score of 0.96. Indeed, C00676 interacting with enzyme 3.1.3.89 is in our training set, while the interaction between enzyme 3.1.3.89 and C02269 was present in the test set.

We also investigate how Boost-RS improves MAP per enzyme class when compared to NMF. Apart from EC 7, which has relatively fewer instances in both training and test sets, Boost-RS improves the MAP for enzyme classes EC 1-6. The largest MAP improvement is for EC 5 isomerases (by 0.45), and the smallest improvement is for EC 2 transferases (by 0.17).

## 4 Conclusion

Our proposed framework, Boost-RS, offers an elegant and generalizable model for boosting learned representations with heterogeneous auxiliary data for CF RSs. Dynamically, training Boost-RS on multiple tasks allows evolving their relative weights along the learning epochs. The learning tasks are applicable to various individual and relational attributes. While intended as a general framework, we here demonstrated the utility of Boost-RS for enzyme-substrate interaction prediction task. We demonstrated Boost-RS on three CF baseline models. We identified four auxiliary data (molecular fingerprints, enzyme commission numbers, functional orthologs and bio-transformation relationships), and proved their relevance for enhancing interaction prediction. While we assumed all non-positive

interactions as negative, Boost-RS performance may be improved by the addition of known inhibitory data as hard negative data (Visani et al., 2021). Importantly, we showed the flexibility of Boost-RS framework through multi-task learning allows the integration of various auxiliary data modalities such as individual attributes, group attributes, pairwise relationship. Replacing relational attributes with their multi-label representations (not using contrastive-learning loss), or naively concatenating their multi-label representations to the embedding at the input (not using multi-task learning) cannot achieve the same performance as with Boost-RS. We compared Boost-RS with similarity-based RS models and showed that Boost-RS outperforms GRGMF when utilizing the same data. Because of its demonstrated advantages, generality and elegance in integrating attributes with CF, the Boost-RS framework may prove beneficial for a diverse set of application. Further, the use of the trained auxiliary machinery might prove useful in addressing the cold-start problem, common across RSs.

## Funding

This research was supported by NSF [Award 1909536]; NSF [Award 1908617] to L.-P.L. The research was also supported by the NIGMS of the National Institutes of Health [Award R01GM132391]. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

*Conflict of Interest:* none declared.

## References

- Acun, B. et al. (2021) Understanding training efficiency of deep learning recommendation models at scale. In: *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, Seoul, Korea (South). IEEE, pp. 802–814.
- Bagherian, M. et al. (2021) Machine learning approaches and databases for prediction of drug–target interaction: a survey paper. *Brief. Bioinform.*, **22**, 247–269.
- Belharbi, S. et al. (2016) Deep multi-task learning with evolving weights. In: *ESANN, Bruges, Belgium*.
- Bowie, J.U. et al. (2020) Synthetic biochemistry: the bio-inspired cell-free approach to commodity chemical production. *Trends Biotechnol.*, **38**, 766–778.
- Durant, J.L. et al. (2002) Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.*, **42**, 1273–1280.
- Gao, L. et al. (2018) Recommendation with multi-source heterogeneous information. In: *IJCAI International Joint Conference on Artificial Intelligence, Stockholm, Sweden*.
- He, X. et al. (2017) Neural collaborative filtering. In: *Proceedings of the 26th International Conference on World Wide Web, Perth, Australia*. pp. 173–182.
- Jiang, J. et al. (2021) Learning graph representations of biochemical networks and its application to enzymatic link prediction. *Bioinformatics*, **37**, 793–799.
- Kanehisa, M. and Goto, S. (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, **28**, 27–30.
- Kanehisa, M. et al. (2016) KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Res.*, **44**, D457–D462.
- Khersonsky, O. and Tawfik, D.S. (2010) Enzyme promiscuity: a mechanistic and evolutionary perspective. *Annu. Rev. Biochem.*, **79**, 471–505.
- Kingma, D.P. and Ba, J. (2015) Adam: a method for stochastic optimization. In: *ICLR, San Diego, CA, United States*.
- Kotera, M. et al. (2013) KCF-S: KEGG chemical function and substructure for improved interpretability and prediction in chemical bioinformatics. *BMC Syst. Biol.*, **7**, 1–17.
- Lim, H. et al. (2016) Large-scale off-target identification using fast and accurate dual regularized one-class collaborative filtering and its application to drug repurposing. *PLoS Comput. Biol.*, **12**, e1005135.
- Liu, Y. et al. (2016) Neighborhood regularized logistic matrix factorization for drug–target interaction prediction. *PLoS Comput. Biol.*, **12**, e1004760.
- Liu, S. et al. (2019) Loss-balanced task weighting to reduce negative transfer in multi-task learning. *Proc. AAAI Conf. Artif. Intell.*, **33**, 9977–9978.
- Mellor, J. et al. (2016) Semisupervised gaussian process for automated enzyme search. *ACS Synth. Biol.*, **5**, 518–528.
- Mnih, A. and Salakhutdinov, R.R. (2008) Probabilistic matrix factorization. In: *Advances in Neural Information Processing Systems, Vancouver, Canada*.
- Porokhin, V. et al. (2021) Analysis of metabolic network disruption in engineered microbial hosts due to enzyme promiscuity. *Metab. Eng. Commun.*, **12**, e00170.
- Ridder, L. and Wagener, M. (2008) SYGMA: combining expert knowledge and empirical scoring in the prediction of metabolites. *ChemMedChem Chem. Enabling Drug Discov.*, **3**, 821–832.
- Romero, P.A. and Arnold, F.H. (2009) Exploring protein fitness landscapes by directed evolution. *Nat. Rev. Mol. Cell Biol.*, **10**, 866–876.
- Sun, Z. et al. (2019) Research commentary on recommendations with side information: a survey and research directions. *Electron. Commer. Res. Appl.*, **37**, 100879.
- Tyzack, J.D. and Kirchmair, J. (2019) Computational methods and tools to predict cytochrome p450 metabolism for drug discovery. *Chem. Biol. Drug Des.*, **93**, 377–386.
- Van der Maaten, L. and Hinton, G. (2008) Visualizing data using T-SNE. *J. Mach. Learn. Res.*, **9**, 2579–2605.
- Visani, G.M. et al. (2021) Enzyme promiscuity prediction using hierarchy-informed multi-label classification. *Bioinformatics*, **37**, 2017–2024.
- Wang, X. et al. (2019) Neural graph collaborative filtering. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, United States*. pp. 165–174.
- Wang, Y. et al. (2021) Multitask feature learning approach for knowledge graph enhanced recommendations with ripplenet. *PLoS One*, **16**, e0251162.
- Webb, E.C. (1992) *Enzyme Nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes*. San Diego, Academic Press.
- Weinberger, K.Q. and Saul, L.K. (2009) Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, **10**, 207–244.
- Xue, H.-J. et al. (2017) Deep matrix factorization models for recommender systems. In: *IJCAI International Joint Conference on Artificial Intelligence, Melbourne, Australia*, Vol. 17, pp. 3203–3209.
- Zhang, Z.-C. et al. (2020) A graph regularized generalized matrix factorization model for predicting links in biomedical bipartite networks. *Bioinformatics*, **36**, 3474–3481.
- Zheng, X. et al. (2013) Collaborative matrix factorization with multiple similarities for predicting drug–target interactions. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, United States*. pp. 1025–1033.
- Zhu, J. et al. (2017) Broad learning based multi-source collaborative recommendation. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore*. pp. 1409–1418.