4-2024

Context-Aware REpresentation: Jointly learning item features and selection from triplets

Rodrigo ALVES

Antoine LEDENT
Singapore Management University, aledent@smu.edu.sg

Context Aware REpresentation: Jointly Learning Item Features and Selection from Triplets

Rodrigo Alves and Antoine Ledent

***Abstract*—In areas of machine learning such as cognitive modeling or recommendation, user feedback is usually context dependent. For instance, a website might provide a user with a set of recommendations, and observe which (if any) of the links were clicked by the user. Similarly, there is growing interest in the so-called "odd-one-out" learning setting, where human participants are provided with a basket of items and asked which is the most dissimilar to the others. In both of those cases, the presence of all the items in the basket can influence the final decision. In this paper, we consider a classification task where each input consists of three items (a triplet), and the task is to predict which of the three will be selected. Our aim is not only to return accurate predictions for the selection task, but to additionally provide interpretable feature representations for both the context and for each individual item. To achieve this we introduce CARE, a specialized neural network architecture that yields Context Aware REpresentations of items based on observations of triplets of items alone. We demonstrate that, in addition to achieving state-of-the-art performance at the selection task, our model is able to produce meaningful representations both for each item, as well for each context (triplet of items). This is done using only triplet responses: CARE has no access to supervised item-level information. In addition, we prove parameter counting generalisation bounds for our model in the i.i.d. setting, demonstrating that the apparent sample sparsity arising from the combinatorially large number of possible triplets is no obstacle to efficient learning.**

***Index Terms*—Cognitive Models of Knowledge, Collaborative Filtering, Recommender Systems, Odd-one-out Problem.**

I. Introduction

Humans evaluate the properties of objects by considering a variety of criteria, ranging from (1) physical appearance and functionality to (2) abstract and intangible attributes [1], [2]. Their decisions are also typically influenced by cultural and social relationships, which leads to the establishment of comparable judgements within particular environments [3], [4].

The mathematical modeling of human mental representations of object concepts is a fundamental open problem in cognitive science. Here "mental representation" does not refer to how concepts are biologically represented via brain activity or neuronal connections. Instead, it refers to a means of associating objects with vectors that geometrically capture how humans consider objects. For example, one would want the vector representing "dog" to be closer to "cat" than "wrench". A common approach to finding these representations is to collect a large number of human responses on some task (regarding the properties of a diverse set of objects), and then propose models to retrieve embedded representations. For example, several works focused on collecting and describing object features related to different setups, such as food [5], object pictures [6] and language semantics [7].

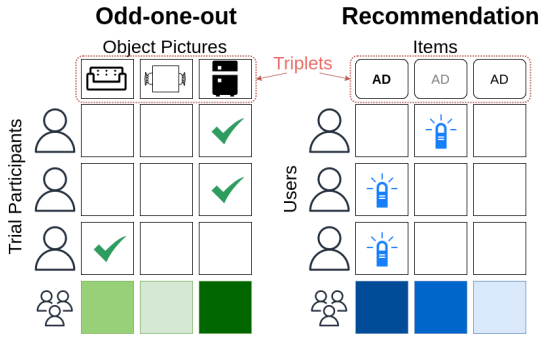

Figure 1: Comparison between odd-one-out task and recommendation task

The machine learning community has recently taken some interest in modelling mental representations of objects. For instance, SPoSE [8], [9] and VICE [2] are recently developed methods that attempt to profile object representations based on human judgements. These methods are trained on responses to the *odd-one-out* triplet task where a human subject is presented with three randomly chosen objects, e.g., a table, a sofa, and a refrigerator, and is instructed to select the object least like the others, or equivalently the pair that are the most similar (see Figure 1). With many responses to this task these methods are capable of learning a vector for each object that captures the semantic properties of the objects.

Outside of cognitive science, the odd-one-out task naturally occurs in many recommender systems (RS) scenarios. For instance, an RS may select K advertisements for a website and then display them to a cold-start user, who will typically select the most relevant one (odd-one-out item) according to their own judgment. Based on the user selections, the RS profiles items and learns similarities with the aim of performing more accurate recommendations in the future. Notably, the vast majority of recommender system situations, including implicit or explicit feedback [10], collaborative or content-based filtering [11], [12], cold and warm-start [13], [14], can be viewed as odd-one-out tasks. For $K = 3$, the "recommended triplets" consist of the three ads being shown and is similar to the triplet odd-one-out task that occurs in cognitive science (see Figure 1, right). For the odd-one-out task the composition of the triplet items might influence the

Accepted for publication in Transactions of Neural Networks and Learning Systems (TNNLS), DOI: 10.1109/TNNLS.2024.3383246

users' judgments. Thus, integrating the composition of the triplet, which we will call *context*, into a method may help to enhance object representations and, potentially, improve prediction accuracy.

In this paper, we investigate how to model and extract interpretable embeddings that characterize the users' representations of items that integrate context. To this end, we propose CARE (which stands for "Context Aware REpresentations"). Our model is an artificial neural network that includes a permutational layer (PL) (see Figure 2) that receives three latent vectors (one for each item in the triplet) as input. The PL is invariant to the order of the input vectors: any permutation of the three items in a triplet yields the same output of the PL. Unlike previous models for the odd-one-out task, this architecture allows us to learn a *context embedding* that summarizes the content of each triplet along with the embeddings of individual items. Unlike SPoSE and VICE, the odd-one-out item is set to be the item whose feature representation is the furthest away from the context vector (the distance serves the role of a classification score). The item and context representations are learned jointly by attaching the cross-entropy loss to the final odd-one-out predictions and observations. We enforce component wise positivity and apply ℓ^1-norm regularization to the embeddings layer in order to enhance the interpretability of the recovered item embeddings.

Our contributions are described as follows:

- we introduce CARE, an end-to-end model which jointly learns item and context representations from triplets of observations with odd-one-out responses, without ever having access to item-level supervised information;
- we demonstrate through a series of experiments on four datasets that our method exhibits state-of-the art performance at the odd-one-out selection task from triplets;
- we experimentally investigate the behavior of our item level feature representations and demonstrate that we are able to recover interpretable qualitative information about item space *without using any item-level supervision*;
- we prove generalization bounds for the triplet selection task in an i.i.d. setting, illustrating that the sample complexity of the model is still comparable to the number of parameters of the model, despite the fact that the total number of triplets is combinatorially large.

II. Related Works

Triplet-based learning settings encompass various applications. For example, the triplet-based odd-one-out task has been used not only to establish a framework for assessing how artificial intelligence mechanisms can interactively estimate pairwise similarity between objects [15] and solve IQ problems [16], but it has also found significant utility in the realm of cognitive science [2]. Furthermore, in [17], the authors employ a reinforcement learning algorithm to address the odd-one-out task, specifically focusing on triplets of geometric figures characterized by their shapes, colors, and textures. Beyond the odd-one-out task, triplet losses have been used as a contrastive supervision signal to train other learning tasks in computer vision [18]. Likewise, in sentiment analysis, the task of *triplet extraction* consists in extracting relational triplets of concepts from a sentence, which could be framed as identifying an optimal one out of many candidate triplets [19], [20].

In cognitive science, this task plays a crucial role in exploring human mental representations, forming an important strand of research [21]–[24]. However, other methods have also been employed by cognitive scientists to collect data for studying mental representations. In this context, previous studies have asked subjects to provide descriptors for an object [25] or to select which descriptors from a list best describe an object [26]. In other cases, subjects were asked to report the most similar pair of objects using a Likert scale (ranging from 1 to 10) [27]. The odd-one-out triplet task offers significant advantages over these approaches: it is straightforward to answer, it is insensitive to subject differences in rating scale (e.g., subject A's rating of 7 may be equivalent to subject B's rating of 10), and it does not make any a priori assumptions about which properties are important.

To our best knowledge, two models have been proposed for learning representation vectors from odd-one-out triplet responses: SPoSE [9] and VICE [2]. Both algorithms are based on a model of similarity using a softmax function. Given objects indexed by i, j, k with representations x_i, x_j, x_k, the probability of $\{a, b\} \subset \{i, j, k\}$ being selected as the most similar pair is given by

$$P\left(\{a,b\} \mid \{i,j,k\}\right) = \frac{\exp(x_a^T x_b)}{\exp(x_i^T x_j) + \exp(x_j^T x_k) + \exp(x_i^T x_k)}.$$

We observe that CARE does not use this type of similarity-based model. Given a collection of odd-one-out triplet responses, both SPoSE and VICE learn object representations that are enforced to be nonnegatve and are encouraged to be sparse by ℓ^1 regularization. VICE is a Bayesian approach relying on Variational Inference to extract item representations. Like SPoSE, VICE was also found to return semantically meaningful dimensions for the object representations on the odd-one-out task on datasets such as THINGS [6], [8], [28]. On THINGS, VICE's was found to require far less data than SPoSE to achieve similar predictive performance. Later works have found the representations from VICE to be consistent with results from other object similarity tasks [29]. Note that unlike the present work, neither of these contributions experimented with applications of triplet-based learning settings to Recommender Systems contexts.

Pairwise Learning: A natural predecessor of triplet losses and triplet learning strategies such as the odd-one-out task we study is found in **pairwise learning**, with incorporates the theoretical study of learning settings where the loss function depends on two inputs instead of one. There is a wide array of research on the theoretical and practical properties of such models [30]–[35].

Recommender systems build internal representations of user and item behaviors to predict which items users are likely to enjoy [36]–[42]. A vast range of research focuses on explaining and interpreting such user and item representations. For instance, previous works analyzed user and item properties via social network analysis [43], [44], natural language

processing [45]–[47], or feature extraction in content-based methods [48], [49]. Others relied on user and (or) item-based collaborative filtering approaches [50]–[55] to extract information from interactions alone. For a detailed survey, we suggest [56]. Our method belongs to the last category in the sense that we explore patterns in the interactions between user-items in an RS to extract item features. However, differently from most of the previously proposed methods, CARE is user-agnostic by focusing on cold-start recommendation sessions. In addition, to the best of our knowledge, we are the first to consider the triplet-based learning setting and its analogy to the odd-one-out task from cognitive sciences in a recommendation context. This makes the quality of our feature representations particularly noteworthy since they are obtained relying solely on *triplet-based* observations *without* access to the *user IDs* or even any item-level information: in particular, the above methods cannot be directly compared to ours either qualitatively (w.r.t. interpretability) or quantitatively (w.r.t. performance).

Several studies attempt to extract context vectors from *session content* by employing, for instance, convolutional neural networks [57] or recurrent neural networks [58], [59]. In this regard, the most similar approach to ours is [57], which models session content as time series, focusing on the progression through the items in the order in which they are viewed. In sharp contrast, our context vector summarizes the triplet's content in a permutation-invariant fashion by relying on a permutation layer. This disentangles the order information from the context vector, simplifying interpretation. Thus, while related works are focused on predicting future interactions in session-based settings, our approach is much better targeted at delivering interpretable observations in a more general "grouped sample" setting such as click prediction.

Self-supervised methods for recommendation systems: Contrastive learning [60]–[63] is a broad class models which aim to learn useful representations in data in a self-supervised way. This can be done either through direct comparisons [32], [64] or by maximizing the mutual information between learnt features and the samples [60], [65]. Alternatively, the Information Bottleneck approach [66], [67] maximizes the mutual information with the labels whilst minimizing the mutual information with the samples, thus ensuring that only the information most closely related to the classification task is retained. Whilst such techniques have historically focused on image data [60], [61], there has been a growing interest in their application in the context of Graph Neural Networks and Recommendation Systems [65], [68]. In particular, [69] uses the Information Bottleneck approach to learn three disentangled representation of the users and items through the interaction graph and variants of it relying on edge dropping and node dropping. The method is further incorporated in an involved jointly trained model which performs the interaction prediction task. Furthermore, LightGCL [70] uses a contrastive loss learn representations from the interaction graphs by using a LightGCN [71] architecture over the original graph as a teacher model and a simplified model relying on a spectrally-truncated version of the message-passing layer as a student. However, the self-supervised aspect of these works merely refers to the optimal extraction of user and item representations from interaction graphs without explicit supervisions regarding the nature of individual items (e.g. movie content, genre, synopsis). Indeed, whilst some of these methods do make use of item-wise pairwise losses in training ([70], [71]), they still utilize the full matrix of interactions between users and items to construct their embeddings. In particular, they still rely on information at the level of individual items and even (user, item) pairs by comparing every pair of items for each individual user. In contrast, our work relies only on triplet-level training data: only a choice of item out of three for the triplets in the training set is provided. This corresponds to a much weaker form of supervision for several reasons: (1) None of our models have access to the user ID, (2) Not all combinations of items are available in the training data, and (3) comparisons are over triplets rather than pairs.

III. Model and properties

A. CARE model

In this section, we define our model mathematically and discuss its basic properties. CARE consists in three main components: (1) a permutational Layer; (2) a feed-forward neural network; and (3) a distance-based classifier. The permutational layer performs a set of operations on the feature representations of the three items to obtain a context vector Z summarizing an entire triplet. These operations enforce permutational invariance between the components of the triplets, whilst encoding the similarity between the spaces in which the component vectors live. Next, a feed forward neural network $\bar{\phi}$ is applied to Z to obtain a *context vector* ρ, which represents our final feature representation of the triplet.

Finally, our model picks the item whose feature representation's Euclidean distance to the context vector is the greatest. Thus, if the input to our model is (e_1, e_2, e_3), our output will be whichever of e_1, e_2 and e_3 is deemed to be the "odd-one-out", i.e. the item which is the least similar to the other two. A particularly interesting aspect of our method is that this whole operation can be trained in an end-to-end fashion with the feature representations of each item being trainable parameters. Thus, our model can learn both item-level feature representations and triplet level *context vectors* from the observation of triplet level labels only.

The next three subsections each correspond to one of the components of the model. The model is also illustrated in Figure 2. Finally, Subsection III-B contains a discussion of the generalization behaviour of triplet based methods such as ours.

Permutational Layer: We write n for the number of items/objects. We also write E for the set of all such items. Therefore $n = |E|$. For each triplet (e_1, e_2, e_3), the label, which represents which of the three items should be picked, is represented by y. The (label,triplet) pairs are drawn i.i.d. from some distribution $\mathcal{P}$ over $\{1, 2, 3\} \times E^3$. Our model assigns a separate learnable d-dimensional feature vector $x_e \in \mathbb{R}^d$. We refer to this process as the **feature assignment layer**. We collect all the vectors x_e as the columns of a (trainable) matrix $\chi \in \mathbb{R}^{d \times n}$. For each triplet (e_1, e_2, e_3), the output of our feature assignment layer is the matrix $X := (x_{e_1}, x_{e_2}, x_{e_3})$.

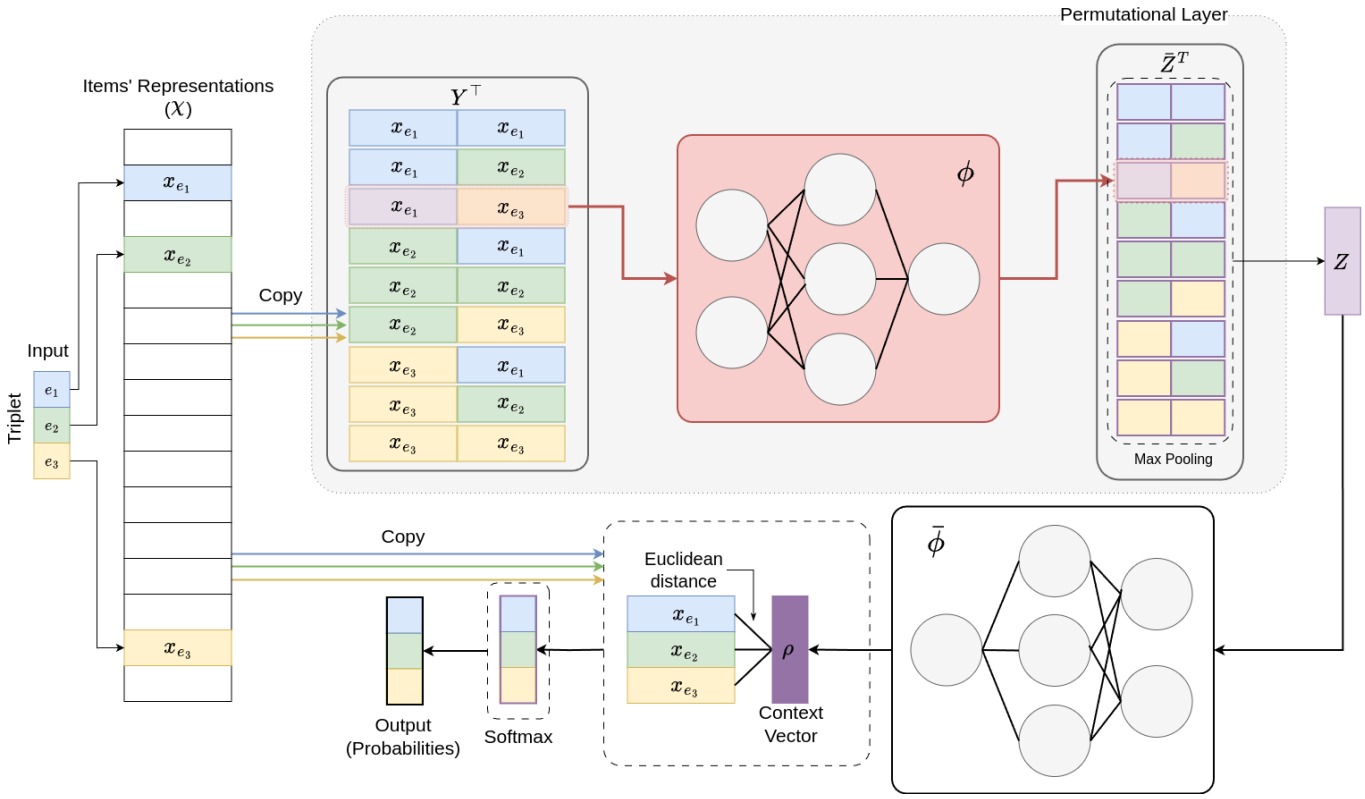

Figure 2: Architecture of CARE. Only χ (which contains the embeddings of each item) and the neural networks ϕ and $\bar{\phi}$ are learning parameters. In the first step of the permutation layer (top left of the picture), all the pairwise combinations of the three item representations are constructed. The resulting pairs are all fed through a trainable neural network ϕ, whose outputs are then subjected to max pooling. Then, another neural network $\bar{\phi}$ is used to extract a context vector from the result. This context vector is then compared to the original embeddings via the euclidean distance to obtain a prediction of the odd-one-out. More details are provided in the text.

Note that since the feature representations are trainable, we can assume without loss of generality that E is the set of basis vectors of $\mathbb{R}^n$. This means that by identifying the ith object with the vector $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$ (where the 1 is in the ith position), we can interpret the feature assignment layer as multiplication by X on the left. Indeed, if $e_1 = i, e_2 = j, e_3 = k$, we have $X := (x_{e_1}, x_{e_2}, x_{e_3}) = \chi e$, where $e \in \mathbb{R}^{n\times 3}$ is formally interpreted as the input to our network and consists of three columns which are the indicator vectors for positions i, j, k respectively.

Next, our model performs a **permutational layer**, which consists in three operations. The first part of the permutational layer consists in collecting all pairwise concatenations of the three inputs (including self concatenations), which are then collected as rows of the outputs. More formally, the output of the first part of the permutational layer is a matrix $Y \in \mathbb{R}^{2d\times 9}$ where the first (resp. second, third) column of Y is the vector $\mathrm{concat}(x_1, x_1)$ (resp. $\mathrm{concat}(x_1, x_2)$, $\mathrm{concat}(x_1, x_3)$). Similarly, the fourth (resp. fifth, sixth) column of Y is the vector $\mathrm{concat}(x_2, x_1)$ (resp. $\mathrm{concat}(x_2, x_2)$, $\mathrm{concat}(x_2, x_3)$) and the seventh, eighth and ninth columns of Y are the vectors $\mathrm{concat}(x_3, x_1)$, $\mathrm{concat}(x_3, x_2)$ and $\mathrm{concat}(x_3, x_3)$ respectively. Observe that this layer can be interpreted as a sequence of matrix multiplications as follows

$$Y = \begin{pmatrix} X & 0 \\ 0 & X \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix}, \tag{1}$$

where

$$P_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \tag{2}$$

and

$$P_2 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{3}$$

Next, for the second part of the premutational layer, we apply a "**network in network**": each column of Y is fed through a given fully-connected neural network ϕ with D_1 parameters. The output of the network in a network component is denoted by $\widetilde{Z} \in \mathbb{R}^{d\times 9}$:

$$\widetilde{Z}_{\bullet,k} = \phi(Y_{\bullet,k}) \tag{4}$$

for all $k \leqslant 9$.

This network in network component is analogous to a series of one dimensional convolutions, and has been explored in various settings, notably in the Inception architecture [72].

Note that since we arrange the spacial components as the columns of our matrix Z, the network in network component can also be explicitly written in terms of the weights matrices of the network ϕ. Let $W^1, \dots, W^{L_1}$ denote the weight matrices corresponding to each layer of the network ϕ. Let also σ denote our component-wise activation function, which we choose as Relu. Thus, Equation (4) can also be written:

$$\widetilde{Z} = \phi(Y) = \sigma(W^{L_1}\sigma(W^{L_1-1}(\dots\sigma(W^1 Y)\dots))). \quad (5)$$

Next, we perform a max pooling operation over the spacial dimension, which we denote by $\sigma_{\max}$, leading to the final output Z, which is obtained from $\bar{Z}$ by taking a row-wise maximum so that $\mathbb{R}^d \ni Z$ is

$$Z = \sigma_{\max}(\bar{Z}) = \sigma_{\max}(\sigma(W^{L_1}\sigma(W^{L_1-1}(\dots\sigma(W^1 Y)\dots)))). \quad (6)$$

Equivalently, $Z_i = \max_k \widetilde{Z}_{i,k}$ for all $i \leqslant d$. Thus, $\mathbb{R}^d \ni Z$ is the final output of our permutational layer. Permutational layers have been used previously to encode permutational invariance when modeling interacting particle systems [73]. Note that is it trivial to check that Z is invariant to permutations of the input vectors (e_1, e_2, e_3), since all pairwise concatenations appear in Y in any case, and the row-wise maximum removes any notion of order between the columns of Y.

Fully-connected component and context vector: Next, we feed the output Z of the permutational layer through a second neural network $\bar{\phi}$ with D_2 parameters $\overline{W}^1 \dots \overline{W}^{L_2}$ and non linearity $\sigma =$ Relu. The output is a vector

$$\rho := \bar{\phi}(Z) = \sigma\overline{W}^{L_2}\sigma(\overline{W}^{L_2-1}(\dots\sigma(\overline{W}^1 Z)\dots))). \quad (7)$$

This vector ρ is what we refer to as a **context vector**: it is a deep feature representation of the triplet which is invariant to permutations of the triplet. In addition, the structure of the permuation layer implies that the function which maps the triplet to the context vector maintains a notion of individuality between the three inputs through the use of the network ϕ, which is individually applied to each column of Y. Thus, the permutation layer represents a well-principled way to encode the knowledge that the input consists in a triplet of three objects/feature representations living in the same space, with permutational invariance between them.

Euclidian distance-based classifier: The aim of our model is to identify the odd-one-out, i.e. the item with the least similarity to the other two as possible. To achieve this, we pick the item whose feature representation is the furthest away from that of the context vector. Formally, our model consists in a score for each possible item, each being equal to the squared distance to the context vector:

$$f(e_1, e_2, e_3) = (\|\rho - x_1\|, \|\rho - x_2\|, \|\rho - x_3\|). \quad (8)$$

At prediction time, we predict the odd-one-out to be the item e_i where

$$i = \text{argmax}\,(\|\rho - x_1\|, \|\rho - x_2\|, \|\rho - x_3\|). \quad (9)$$

In addition, probabilities for each item can also be obtained via a softmax layer, similarly to classic classification problems.

Note that the use of a distance-based classifier at the last layer makes our model more intuitive and increases the potential for interpretability of the feature representations obtained for each item (which can be recovered from the matrix χ), as well as the context vector ρ.

Finally, in our experiments, we further increase the interpretability of our model by ensuring component-wise positivity of the feature representations.

B. On the generalization performance of triplet based models

Writing n for the number of items, the total number of possible triplets is $\binom{n}{3}$, which is $O(n^3)$. For example, in the MovieLens dataset, we have 1600 items, which gives around 6.8×10^8 possible triplets. Thus, it is clear that it is impossible to sample a significant proportion of the total set of triplets. However, a careful understanding of classic results from Statistical Learning Theory shows that only the total function class capacity of our model is relevant: as long as the number of parameters is not too large (and certain Lipschitzness constraints are satisfied), the number of samples (triplets) required to train the model properly (the *sample complexity*) will scale roughly like the number of parameters in the model. In our case, this number is $dn + D_1 + D_2$, which scales linearly in both the number of the of parameters of the hidden networks D_1, D_2 and the number of items n. This is irrespective of the total number of triplets. Indeed, we have the following theorem:

Theorem 3.1: Let $c > 0$ and $B, R > 0$ be fixed constants. Assume that the activation function is RELU and that the triplets are drawn i.i.d. with a distribution $\mathcal{P}$ over $\{1,2,3\} \times E^3$. For any $\delta > 0$, with probability greater than $1 - \delta$ over the draw of the training set, if the weights of our neural network satisfy

$$\|\chi\|_{\text{Fr}} + \sum_{\ell=1}^{L_1} \|W^\ell\|_{\text{Fr}} + \sum_{\ell=1}^{L_2} \|\overline{W}^\ell\|_{\text{Fr}} \leqslant R \quad (10)$$

$$\|\chi\|_{\text{Fr}} \prod_{\ell=1}^{L_1} (\|W^\ell\|_{\text{Fr}} + 1) \prod_{\ell=1}^{L_2} (\|\overline{W}^\ell\|_{\text{Fr}} + 1) \leqslant \Gamma, \quad (11)$$

then for large enough N, we also have the following generalization bound, where ℓ_Λ represents the margin loss [1]:

$$\mathbb{E}\left(\ell_\Lambda(y, f(e_1, e_2, e_3)\right) - \widehat{\mathbb{E}}\left(\ell_\Lambda(y, f(e_1, e_2, e_3))\right) \quad (12)$$

$$\leqslant \sqrt{\frac{(dn + D_1 + D_2)\log\left(\frac{R\Gamma}{\Lambda}\right) + \log\left(\frac{1}{\delta}\right)}{N}}, \quad (13)$$

where C is some constant and $\widehat{\mathbb{E}}$ represents the empirical expectation over the training set of (label, triplets) pairs available.

Proof 3.1: See Appendix.

Recall that here, N is the number of triplets sampled, n is the number of items, and f is the neural network function defined above. Note that for relatively shallow networks (i.e. if the

[1] $\ell_\Lambda(y, f) = 0$ if the classification margin is $\geqslant \Lambda$, $\ell_\Lambda(y, f) = 0$ for miss classified triplets, see also Equation (17) in the Appendix.

number of layers is fixed, e.g. 10), $R\Gamma$ can be considered a non-exponential quantity [2], meaning that bound scales roughly like $\widetilde{O}\left(\sqrt{\frac{dn+D_1+D_2}{N}}\right)$: the dominant term in the *sample complexity* of our model is $\widetilde{O}\left(dn + D_1 + D_2\right)$, with hidden logarithmic factors of the constraints on the norms and the margin requirement.

A noteworthy point is that the complexity only scales linearly in n (the number of objects/items), despite the fact that we are using a triplet-based model and the number of triplets that can be formed (which is $\begin{pmatrix} n \\ 3 \end{pmatrix}$) grows polynomially in n. This illustrates that the apparent sample sparsity arising from the combinatorially large number of possible triplets is no obstacle to efficient learning: if the number of items grows by a factor of 2, the number of randomly sampled triplets required to reach comparable accuracy only grows by a factor of 2, even though the number of triplets grows roughly by a factor of 8.

Remarks:

1) Our bounds involve the quantities R and Γ, which are upper bounds on the norms of the parameters. Although they must be selected in advance, it is trivial to modify the results to replace them by the observed values of the quantities on the left hand sides of Equations (10) and (11). This can be done via a union bound [74], [75] and is usually left to the reader [76], [77].
2) For our results, it is crucial that the triplets be sampled in an i.i.d. fashion and uniformly randomly over the set of all possible triplets involving all items. Indeed, if instead a subset of k items were to be drawn, and all possible triplets involving those k items be used for training, we anticipate the generalization error to still only decay like $\frac{1}{\sqrt{k}}$ even if all $O(k^3)$ triplets are used. This is by analogy with the pairwise learning literature, which considers pairs of instances rather than triplets (see, e.g. [31]).

IV. Experiments

A. Datasets

We compare CARE to the baselines using four datasets. We aim to learn interpretable individual item features, without ever having access to side information related to the individual items. This will be achieved through a model capable of reliably isolating one item from a group of three based on a context-dependent rule. For the **MovieLens**,**GoodReads** and **Douban** datasets the supervision comes from the odd-one-out task of predicting the item with a rating that differs from the others (cf. details below). In the **Outbrain** dataset, each sample naturally consists of a sequence of three advertisements presented to a user, from which the user selects one item to click on. Note that Outbrain functions as an advertising service across diverse websites. Unlike the other datasets, where users rate movies while seeking information in the movie domain, Outbrain's items may not necessarily align with the content of the website the user is accessing. For instance, when a user is reading an article about politics, they could be presented with a set of three different tourism activities. These advertisements are displayed collectively, and the user's decision to click on a particular item is influenced by the composition of the exhibited advertisements. In this case, our model seeks to capture embeddings that represent the contextual composition of the advertisement set, and the selected "odd-one-out" item is the one that most effectively captures the user's attention at that specific moment. A comprehensive description of our two datasets and sampling procedure is provided below.

MovieLens[3], **GoodReads** and **Douban**: In MovieLens and Douban [78], users are members of a social network, items are movies and the entry of a rating matrix r_{ij} is the rating given by user i to movie j, in a scale $\{1, 2, 3, 4, 5\}$. The GoodReads [79] (Romance) dataset serves as the counterpart to MovieLens and Douban in the realm of books, utilizing a rating scale of $\{0, 1, 2, 3, 4, 5\}$. For these datasets, assume a valid triplet any set of three items $\{a, b, c\}$ with $a \neq b$, $|r_{ia} - r_{ic}| > 1$, $|r_{ib} - r_{ic}| > 1$ and $|r_{ia} - r_{ib}| \leqslant 1$. Note that, for instance, $\{r_{ia}, r_{ib}, r_{ic}\}$ equals to $\{1, 1, 4\}$ or $\{5, 4, 2\}$ are valid $\{a, b, c\}$ triplets, while $\{3, 2, 4\}$ or $\{2, 2, 2\}$ are not. The reason is to construct triplets in accordance with the one-odd-out trials: our triplets always contain items a and b rated by user i as *significantly* more similar than c. Thus c is consistently the odd-one-out item here. *Sampling procedure*: To sample triplets we used the following sampling procedure: (1) randomly split the observed entries of rating matrix R into three groups and then construct matrices $R_{\text{train}}, R_{\text{validation}}$ and R_{test} (with $60\%, 20\%$ and 20% of the original entries, respectively); (2) from R_{train} sample among all valid (a, b, c) triplets with equal probability to build the training set; (3) repeat the process with $R_{\text{validation}}$ and R_{test} to build the validation and test sets. Thus, we employ a stricter separation between training, validation and test sets than the i.i.d. setting. This is so that the quality of our feature representations can be compared favorably even to those obtained through direct item-level supervision in Recommender Systems Settings.

Outbrain[4]: In Outbrain the input data consists of a large number of recommendation batches containing a small set of items $A = \{a_1, a_2, \cdots, a_k\}$, with $k \in \{2, 3, \cdots, 12\}$ which is presented to the user. The output is the unique $a \in A$ on which the user *clicked*. *Sampling procedure*: to build a triplet dataset from the original dataset we performed the following: (1) drop all instances with $k = 2$; (2) if $k > 2$, we keep the clicked item and randomly selected two among the non-clicked ones. (3) then, we randomly sampled 8.5 million triplets. (4) finally, we filtered the dataset by keeping triplets composed of items that appear at list five times.

Remark: Our model is able to recognize odd-one-out components in multiple settings. It should be observed, however, that the tasks performed using MovieLens and Outbrain differ significantly from an interpretation perspective. We interpret the context vector, in both cases, as a summary of the triplet's content.

[2]because it only has implicit exponential dependence in the number of layers L assuming the norms of the weights don't vary too much between layers

[3]Available in: https://grouplens.org/datasets/movielens/

[4]Available in: https://www.kaggle.com/c/Outbrain-click-prediction/data

Table I: Performance comparison of our methods vs baselines on the real datasets. Metric: accuracy

	Douban	GoodReads	MovieLens	Outbrain
SPoSE-Similarity	0.5698 ± 0.0004	0.4582 ± 0.0011	0.4447 ± 0.0012	0.5726 ± 0.0004
SPoSE-Distance	0.5470 ± 0.0005	0.4583 ± 0.0010	0.4210 ± 0.0013	0.5322 ± 0.0043
VICE	0.5687 ± 0.0002	0.4591 ± 0.0003	0.4462 ± 0.0012	0.5724 ± 0.0003
CaRe	$\mathbf{0.5701 \pm 0.0006}$	$\mathbf{0.4658 \pm 0.0008}$	$\mathbf{0.4516 \pm 0.0016}$	$\mathbf{0.5794 \pm 0.0002}$
#Items	2.5K	115K	1.6K	2.5K
#Triplets Train/Val/Test	5.0 M/500K/500K	5.0 M/500K/500K	2.5M/240K/240K	7.6M/425K/425K

B. Baselines

As explained above, although there are many baselines which apply to the recommendation setting and rely on *user* and item-level information, they do not apply to our learning setting which consists in choosing an item from a group of three. Thus, we only compare to baselines which apply to this triplets-based scenario. The three baselines we consider are **SPoSE-Similarity**, **SPoSE-Distance** and **VICE**, and the corresponding methods are explained in detail below.

- Sparse Positive Similarity Embedding **(SPoSE) - Similarity** [8], [9]: SPoSE assumes that the decision in a given odd-one-out trial is explained as a function of the similarity between the embedding vectors of the three concepts presented. To infer embedding vectors from data, the method maximizes similarities between the vectors of the objects of the triplet that are not the odd-one-out.
- Sparse Positive Similarity Embedding **(SPoSE) - Distance** [8], [9]: SPoSE assumes that the decision in a given odd-one-out trial is explained as a function of the distance between the embedding vectors of the three concepts presented. To infer embedding vectors from data, the method minimizes the distance between the vectors of the objects of the triplet that are most similar.
- Variational Interpretable Concept Embeddings (**VICE**) [2]: VICE is an Bayesian method for embedding object concepts in a vector space using data collected from humans in a triplet odd-one-out task. VICE uses variational inference to provide representations of object concepts with uncertainty estimates for the embedding values. VICE incorporates spike-and-slab regularization, reinforcing the reduction of weights. Notably, the spike-and-slab prior imposes a more substantial penalty on larger weights compared to smaller ones, which is conducive to gradient-based optimization techniques. This characteristic makes VICE especially well-suited for modeling scenarios. The parameters involved in the regularization, namely π_{spike}, σ_{spike}, and σ_{spike}, undergo cross-validation. We determine their values using the grid outlined in the subsection "Hyperparameter grid" of Section E in the appendix of [2]).

C. Parameter Setting and Tuning Schemes

In this section we will describe the parameter settings and tuning schemes for our model and the baselines. SPoSE variants involve tuning two hyperparameters: the dimension of the embedding vectors for the three concepts, denoted as p, and a regularization parameter λ controlling the norm of each concept embedding x_i. The regularization parameter λ was cross validated by searching over the set $\{10^{-6}, 5 \times 10^{-6}, \cdots, 0.1, 0.5, 1\}$ and we initialize model with $p = 100$ dimensions. Regarding VICE, the parameters involved in the regularization, namely π_{spike}, σ_{spike}, and σ_{spike}, undergo cross-validation. We determine their values using the grid outlined in the subsection "Hyperparameter grid" of Section E in the appendix of [2]. In all cases, the cited parameters and sections follow the notation of the respective papers.

In our own model, we have tuned the hyperparameter λ in a range that adheres to the same structure as described for λ in **SPoSE** and $D \in \{25, 50, 100\}$. We first run the all the models (the baselines and ours) once for 50 epochs with early stopping with a tolerance of 5 epochs and output the combination that provides the best validation accuracy. We then run each model for 20 times (50 epochs with 5 epochs of tolerance). The results displayed in Table I consist of averages of accuracy over the 20 experiments. We implement our model and **SPoSE** via classic gradient-based methods implemented in Tensorflow 2. For **VICE** we used the code provide by the authors[5]. The number of items and our *train-test setup* for each dataset can be found in Table I.

D. Analysis of the Results

Table I summarizes the datasets' statistics and the results of the real-world data experiments. Although the model's interpretability assessment is qualitative, comparing the accuracy of odd-one-out item prediction of our method with state-of-the-art methods that handle the same problem is a consistent indicator of our method's effectiveness. We compared the performance of our method to all previous baselines and datasets. Observe that our method demonstrates state-of-the-art performance, achieving the highest accuracy in the analyzed datasets, with the added benefit of interpretability through the context vector.

V. Interpretability

This section will analyze the interpretability of the representation vectors our model assigns to each triplet and to each individual item. These can help us to understand what properties CARE learns from the triplet data. Our analysis is conducted using MovieLens and Outbrain datasets.

A. Item Embeddings

Via its *feature assignment layer*, our model encodes item representations into the columns of a matrix $\chi \in \mathbb{R}^{d \times n}$. To

[5]Available at: https://github.com/LukasMut/VICE

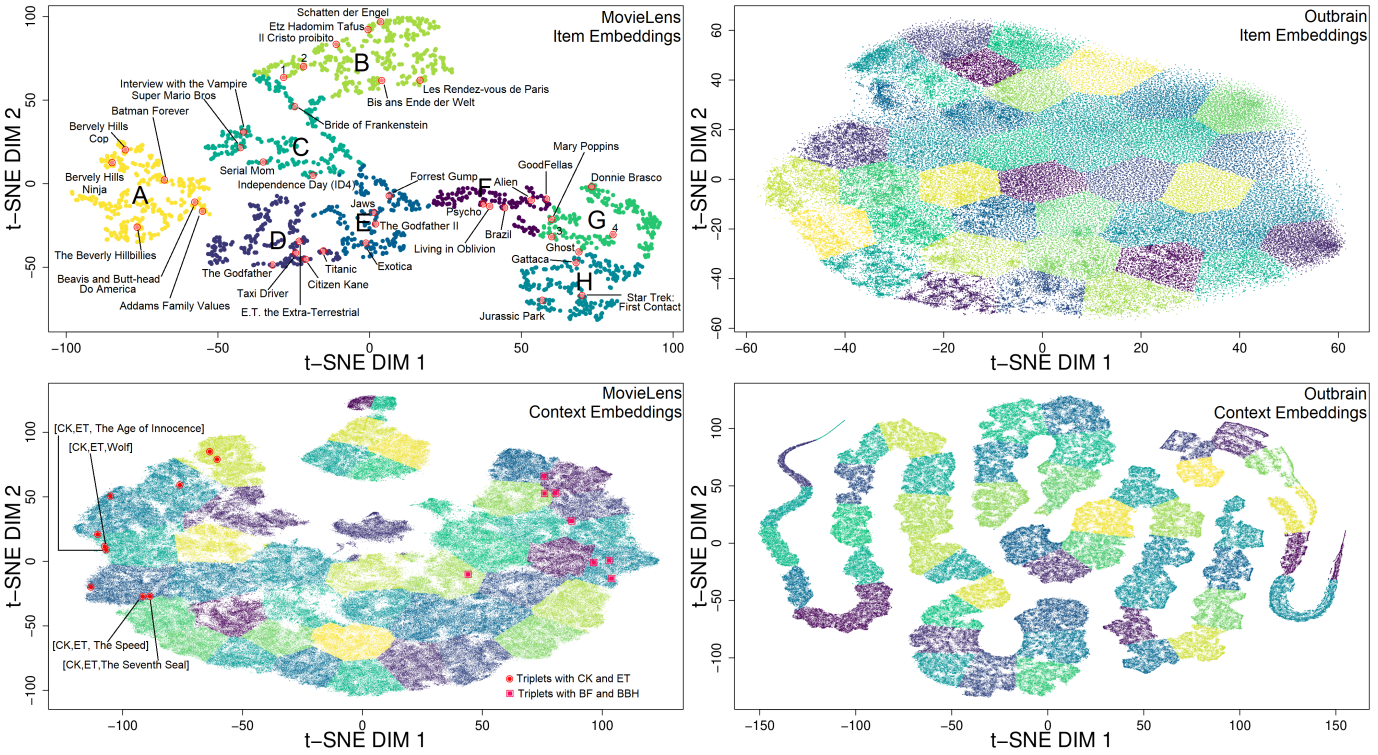

Figure 3: **Top:** Clustering of item embeddings visualized using t-SNE. **Bottom:** Clustering of **context embeddings** derived from our model's feature representations, also visualized using t-SNE. Data from the Movielens (left) and Outbrain (right) datasets are used in both instances. Key: *CK: Citizen Kane*, *ET: E.T. The Extra-Terrestrial*, *BF: Batman Forever*, *BBH: Beavis and Butt-head Do America*.

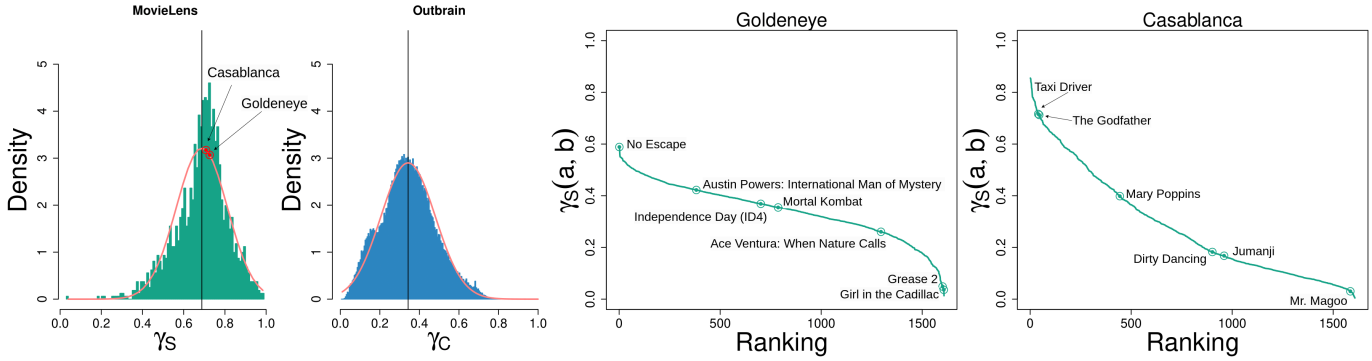

Figure 4: Right: Distribution of the indicators γ_S (for MovieLens) and γ_C (for Outbrain). Left: γ_S disentanglements of items *Goldeneye* and *Casablanca* of MovieLens.

visually investigate the behavior of those representations, we first reduce their dimensionality to two by using t-SNE. After that, we perform the k-means clustering algorithm with the so-called elbow method [80] to select the number of clusters $k \in \{2, 3, \cdots, 100\}$ that effectively minimizes the sum of the squared distances within-cluster. Finally, we plot the graph with the "optimal" k. Results can be seen in the top two graphs of Figure 3. The chosen number of clusters for MovieLens and Outbrain are eight and forty, respectively. For the MovieLens dataset side information is present and can be used to help understand the clusters. Unfortunately, item descriptions are not available for the Outbrain dataset.

For MovieLens we find that the movies' representations are indeed meaningful. For example, the far-left yellow cluster (B) on the top-left image in Figure 3 contains several adult, comedy movies, such as *Beverly Hills Cop III*, *Beavis and Butt-head Do America*, *I Know What You Did Last Summer*. The bottom-left cluster contains serious, classic drama movies like *The Godfather*, *Citizen Kane*, and *Taxi Driver*. Adjacent to the navy cluster (D), we find a dark blue one (E) with, for instance, *Forrest Gump*, *The Godfather: Part II*, and *Jaws*. Note that both of these groups are made up of blockbusters that are typically well-liked by the majority of users that interacted with them in MovieLens. CARE effectively captures

the proximity of these two groups. However, one may argue that the latter is composed of movies with relatively more recent success, also justifying our model's decision to group them into two different clusters. Finally, it is noteworthy that the light green group (B, middle top of the graph) contains a substantial sample of "foreign" (non-American) movies such as *Les Rendez-vous de Paris*, *Bis ans Ende der Welt*, and *Il Cristo proibito*. In contrast to the previous two clusters (also opposed in CARE's geography), this group contains rather niche moves, and is accordingly well separated from the other groups in the geometrical representations elicited by our model.

B. Context Embeddings

Our model produces a context vector ρ for each triplet (observed or non-observed). To explore patterns in the context vector distribution of MovieLens and Outbrain, we randomly selected 5×10^5 triplets and applied the same dimensionality reduction method to their context vectors as was used for item embeddings in Subsection V-A. The resulting visualization is displayed in the bottom half of Figure 3.

The red dots concentrated on the left of the MovieLens graph correspond to triplets containing both *Citizen Kane* and *E.T. the Extra-Terrestrial* (both movies' *item representations* belong to Cluster D – top left graph of Figure 3). Since such triplets have two items in common, it is expected that they are not too far from each other, as observed. We also expect them to be closer or further away depending on the nature of the third element, which we indeed observe, further highlighting the relevance of the representations. Indeed, note the two *adjacent* triplets highlighted at the bottom of the plot respectively have, as their third components, the movies *The Seventh Seal* and *The Speed* (datapoints 1 and 2, respectively), whose item embeddings also belong to the same cluster (B). Similar behavior is shown for the data points marked at the top (left) of the graph, with the third item being the movies *Wolf* and *The Age of Innocence* (datapoints 3 and 4, respectively – Cluster H).

Furthermore, such behavior can also be observed in other regions of the space of context vectors: the highlighted data points on the right side of the graph correspond to triplets containing both *Batman Forever* and *Beavis and Butt-head Do America*, which likewise belong to the same cluster (A) in item space (but a different one from that of *Citizen Kane* and *E.T. the Extra-Terrestrial* (D)).

C. Distance Scale γ

The output of CARE is always computed by applying the softmax function to the distances between the context vector ρ and each item vector of the triplet to identify the "oddest" item. However, the precise interpretation of the choice varies according to the task it is solving. Consider a triplet $(a, b, c) \in E^3$, with a, b, and c distinct. In Outbrain, the output score corresponding to the item c refers to the probability of c being clicked on. In MovieLens, however, it refers to the probability of a and b being frequently judged to be of the *most similar quality* by many users. By fixing two items a and b and iterating among all possible c's, we define the score

$$\gamma_S(a, b) = \frac{\sum_{c \neq \{a,b\}} \mathbb{P}(\text{c is oddest item})}{n - 2}. \tag{14}$$

The value of $\gamma_S(a, b)$ can be seen as the probability of a and b being the most similar in a triplet where c is sampled from $E \backslash \{a, b\}$. Define $\gamma_S(a)$ as follows

$$\gamma_S(a) = \frac{\sum_{b \neq a} \gamma_S(a, b)}{n - 1}. \tag{15}$$

Here $\gamma_S(a)$ is the probability of a being selected as one of the most similar (or equivalently, *not* being selected as the odd-one-out output of our model) in a triplet when b and c are sampled from $E \backslash \{a, b\}$. Thus, in the MovieLens dataset, we can interpret $\gamma_S(a)$ as an indicator of how many "similar movies " lie in the neighborhood of a in feature space. On the other hand, in the Outbrain dataset, the feature representations yielded by our model are more targeted to the concrete task of predicting which item is *clicked on*, which complicates the interpretation of $\gamma_S(a)$ as a measure of nearby similarity, since a user clicking on a when presented with the triplet (a, b, c) does not guarantee that b and c are truly similar. However, we can concretely interpret $\gamma_C(a) := 1 - \gamma_S(a)$ as the probability of a being clicked on, which is naturally an interpretable quantity of interest.

Due to the prohibitive computational burden of estimating $\gamma_S(a)$ via all valid triplets in the dataset, we compute an estimate based on a random sample of 10000 valid triplets for each item a. The histograms on the left of Figure 4 show the results. The distributions of $\gamma_S(a)$ (for MovieLens) and $\gamma_C(a) := 1 - \gamma_S(a)$ (for Outbrain) are similar to normal distributions centered at approximately 0.66 and 0.33 (respectively), as could be expected. To better understand the underlying behavior of $\gamma_S(a)$ and $\gamma_S(a, b)$, we computed the set $\Gamma_S(a) := \{\gamma_S(a, b) : b \in E\}$ for $a =$ Goldeneye and for $a =$ Casablanca. The two last graphs of Figure 4 show the resulting distributions. The x-axes of the graphs correspond to the ranking of $\gamma_S(a, b) \in \Gamma_S(a)$, ordered from the largest to the lowest value of $\gamma_S(a, b)$. Note that although $\gamma_S(\text{Goldeneye})$ and $\gamma_S(\text{Casablanca})$ have comparable values, they their distributions are different. A very small number of movies (including *No Escape*) have a very high probability of being considered as most similar together with *Goldeneye*. In sequence, we can then observe a large set of movies with a moderate similarity to Goldeneye, corresponding to a low-rate decay of $\gamma_S(a, b)$. Finally, the line ends with a huge decay: our model detects that there is a very small set of movies, including b=*Grease 2* and b=*Girl in the Cadilac* with a very low similarity $\gamma_S(\text{Goldeneye}, b)$ to Goldeneye. However, for *Casablanca* we can observe an almost quadratic decay, indicating a broader range of degrees of similarity to *Casablanca* in the items in the catalog. We also highlight the *very* high probability of *Casablanca* being classified as the most similar to *Taxi Driver* in a triplet, whilst the opposite behaviour is observed when considered the movie *Mr Magoo* instead.

VI. Conclusion

In this work we have presented an adaption of the odd-one-out task from cognitive science to recommender systems. For this we introduced CARE, a model that performs the odd-one-out task by constructing a context from a triplet and selecting the item representation that is the furthest from the context. We find that this method achieves state-of-the-art accuracy whilst producing highly interpretable representations. Though this work is only an initial foray, we feel that this integration of cognitive science models and recommender systems will be a fruitful direction for future research. Additionally, the question of how CARE mechanisms can be harnessed to enhance recommendation algorithms accuracy remains an open research direction as well as explore the one-odd-out problem for groups of more than three items.

References

[1] B. J. Devereux, L. K. Tyler, J. Geertzen, and B. Randall, "The centre for speech, language and the brain (cslb) concept property norms," *Behavior research methods*, vol. 46, pp. 1119–1127, 2014.

[2] L. Muttenthaler, C. Y. Zheng, P. McClure, R. A. Vandermeulen, M. N. Hebart, and F. Pereira, "Vice: Variational interpretable concept embeddings," *arXiv preprint arXiv:2205.00756*, 2022.

[3] S. Pan and T. Ding, "Social media-based user embedding: A literature review," *arXiv preprint arXiv:1907.00725*, 2019.

[4] T. Ingold, "Culture and the perception of the environment," in *Bush base, forest farm*, pp. 38–56, Routledge, 2002.

[5] M. Carrington, A. G. Liu, C. Y. Zheng, L. Muttenthaler, F. Pereira, J. Avery, and A. Martin, "Naturalistic food categories are driven by subjective estimates rather than objective measures of food qualities," in *Society for Neuroscience Annual Meeting, Washington DC.*, 2021.

[6] M. N. Hebart, O. Contier, L. Teichmann, A. Rockter, C. Y. Zheng, A. Kidder, A. Corriveau, M. Vaziri-Pashkam, and C. I. Baker, "Things-data: A multimodal collection of large-scale datasets for investigating object representations in brain and behavior," *bioRxiv*, pp. 2022–07, 2022.

[7] E. M. Buchanan, K. D. Valentine, and N. P. Maxwell, "English semantic feature production norms: An extended database of 4436 concepts," *Behavior Research Methods*, vol. 51, pp. 1849–1863, 2019.

[8] M. N. Hebart, C. Y. Zheng, F. Pereira, and C. I. Baker, "Revealing the multidimensional mental representations of natural objects underlying human similarity judgements," *Nature human behaviour*, vol. 4, no. 11, pp. 1173–1185, 2020.

[9] C. Y. Zheng, F. Pereira, C. I. Baker, and M. N. Hebart, "Revealing interpretable object representations from human behavior," *arXiv preprint arXiv:1901.02915*, 2019.

[10] G. Jawaheer, M. Szomszor, and P. Kostkova, "Comparison of implicit and explicit feedback from an online music recommendation service," in *proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems*, pp. 47–51, 2010.

[11] R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in *Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop*, vol. 30, pp. 47–56, Barcelona, 2000.

[12] W. Juan, L. Yue-xin, and W. Chun-ying, "Survey of recommendation based on collaborative filtering," in *Journal of Physics: Conference Series*, vol. 1314, p. 012078, IOP Publishing, 2019.

[13] N. A. Abdullah, R. A. Rasheed, M. H. N. M. Nasir, and M. M. Rahman, "Eliciting auxiliary information for cold start user recommendation: A survey," *Applied Sciences*, vol. 11, no. 20, p. 9608, 2021.

[14] R. Sethi and M. Mehrotra, "Cold start in recommender systems—a survey from domain perspective," in *Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2020*, pp. 223–232, Springer, 2021.

[15] J. Sinapov and A. Stoytchev, "The odd one out task: Toward an intelligence test for robots," in *2010 IEEE 9th International Conference on Development and Learning*, pp. 126–131, IEEE, 2010.

[16] J. Mańdziuk and A. Żychowski, "Deepiq: A human-inspired ai system for solving iq test problems," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.

[17] A. K. Lampinen, N. Roy, I. Dasgupta, S. C. Chan, A. Tam, J. Mcclelland, C. Yan, A. Santoro, N. C. Rabinowitz, J. Wang, *et al.*, "Tell me why! explanations support learning relational and causal structure," in *International Conference on Machine Learning*, pp. 11868–11890, PMLR, 2022.

[18] J. Yu, C. Zhu, J. Zhang, Q. Huang, and D. Tao, "Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 2, pp. 661–674, 2020.

[19] H. Fei, Y. Ren, Y. Zhang, and D. Ji, "Nonautoregressive encoder–decoder neural framework for end-to-end aspect-based sentiment triplet extraction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 5544–5556, 2023.

[20] J. Wang, L. Zhang, J. Liu, K. Ma, W. Wu, X. Zhao, Y. Wu, and Y. Huang, "Tgin: Translation-based graph inference network for few-shot relational triplet extraction," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[21] K. Fukuzawa, M. Itoh, S. Sasanuma, T. Suzuki, Y. Fukusako, and T. Masui, "Internal representations and the conceptual operation of color in pure alexia with color naming defects," *Brain and Language*, vol. 34, no. 1, pp. 98–126, 1988.

[22] R. Robilotto and Q. Zaidi, "Limits of lightness identification for real objects under natural viewing conditions," *Journal of Vision*, vol. 4, pp. 9–9, 09 2004.

[23] S. J. Crutch, S. Connell, and E. K. Warrington, "The different representational frameworks underpinning abstract and concrete knowledge: Evidence from odd-one-out judgements," *Quarterly Journal of Experimental Psychology*, vol. 62, no. 7, pp. 1377–1390, 2009.

[24] P. E. Ruiz, "Building and solving odd-one-out classification problems: A systematic approach," *Intelligence*, vol. 39, no. 5, pp. 342–350, 2011.

[25] K. McRae, G. S. Cree, M. S. Seidenberg, and C. Mcnorgan, "Semantic feature production norms for a large set of living and nonliving things," *Behavior Research Methods*, vol. 37, pp. 547–559, Nov 2005.

[26] J. R. Binder, L. L. Conant, C. J. Humphries, L. Fernandino, S. B. Simons, M. Aguilar, and R. H. Desai, "Toward a brain-based componential semantic representation," *Cogn Neuropsychol*, vol. 33, pp. 130–174, June 2016.

[27] J. C. Peterson, J. T. Abbott, and T. L. Griffiths, "Evaluating (and improving) the correspondence between Deep Neural Networks and Human Representations," *Cogn. Sci.*, vol. 42, no. 8, pp. 2648–2669, 2018.

[28] M. N. Hebart, A. H. Dickter, A. Kidder, W. Y. Kwok, A. Corriveau, C. Van Wicklin, and C. I. Baker, "Things: A database of 1,854 object concepts and more than 26,000 naturalistic object images," *PloS one*, vol. 14, no. 10, p. e0223792, 2019.

[29] L. Muttenthaler, J. Dippel, L. Linhardt, R. A. Vandermeulen, and S. Kornblith, "Human alignment of neural network representations," *arXiv e-prints*, p. arXiv:2211.01201, Nov. 2022.

[30] Y. Lei, M. Liu, and Y. Ying, "Generalization guarantee of sgd for pairwise learning," in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 21216–21228, Curran Associates, Inc., 2021.

[31] Y. Lei, A. Ledent, and M. Kloft, "Sharper generalization bounds for pairwise learning," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 21236–21246, Curran Associates, Inc., 2020.

[32] Y. Lei, T. Yang, Y. Ying, and D.-X. Zhou, "Generalization analysis for contrastive representation learning," *arXiv preprint arXiv:2302.12383*, 2023.

[33] Y. Zhou, H. Chen, R. Lan, and Z. Pan, "Generalization performance of regularized ranking with multiscale kernels," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 5, pp. 993–1002, 2015.

[34] S. Duan, S. Yu, and J. C. Príncipe, "Modularizing deep learning via pairwise learning with kernels," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 4, pp. 1441–1451, 2022.

[35] X. Yang, Y. Guo, M. Dong, and J.-H. Xue, "Toward certified robustness of distance metric learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2022.

[36] Z. Li, H. Liu, Z. Zhang, T. Liu, and N. N. Xiong, "Learning knowledge graph embedding with heterogeneous relation attention networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3961–3973, 2021.

[37] W. Liu, Y. Zhang, J. Wang, Y. He, J. Caverlee, P. P. Chan, D. S. Yeung, and P.-A. Heng, "Item relationship graph neural networks for e-commerce," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4785–4799, 2021.

[38] H.-S. Sheu, Z. Chu, D. Qi, and S. Li, "Knowledge-guided article embedding refinement for session-based news recommendation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 7921–7927, 2021.

[39] D. Wu, M. Shang, X. Luo, and Z. Wang, "An l1-and-l2-norm-oriented latent factor model for recommender systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5775–5788, 2022.

[40] L. Xia, C. Huang, Y. Xu, P. Dai, and L. Bo, "Multi-behavior graph neural networks for recommender system," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[41] R. Alves, A. Ledent, and M. Kloft, "Uncertainty-adjusted recommendation via matrix factorization with weighted losses," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2023.

[42] T. Huang, R. Zhao, L. Bi, D. Zhang, and C. Lu, "Neural embedding singular value decomposition for collaborative filtering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 6021–6029, 2021.

[43] H. Park, H. Jeon, J. Kim, B. Ahn, and U. Kang, "Uniwalk: Explainable and accurate recommendation for rating and network data," *arXiv preprint arXiv:1710.07134*, 2017.

[44] L. Quijano-Sanchez, C. Sauer, J. A. Recio-Garcia, and B. Diaz-Agudo, "Make it personal: a social explanation system applied to group recommendations," *Expert Systems with Applications*, vol. 76, pp. 36–48, 2017.

[45] A. Greenstein-Messica, L. Rokach, and M. Friedman, "Session-based recommendations using item embedding," in *Proceedings of the 22nd international conference on intelligent user interfaces*, pp. 629–633, 2017.

[46] F. Costa, S. Ouyang, P. Dolog, and A. Lawlor, "Automatic generation of natural language explanations," in *Proceedings of the 23rd international conference on intelligent user interfaces companion*, pp. 1–2, 2018.

[47] T. Tran, K. Lee, Y. Liao, and D. Lee, "Regularizing matrix factorization with user and item embeddings for recommendation," in *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 687–696, 2018.

[48] X. W. Zhao, Y. Guo, Y. He, H. Jiang, Y. Wu, and X. Li, "We know what you want to buy: a demographic-based system for product recommendation on microblogs," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1935–1944, 2014.

[49] F. Vasile, E. Smirnova, and A. Conneau, "Meta-prod2vec: Product embeddings using side-information for recommendation," in *Proceedings of the 10th ACM conference on recommender systems*, pp. 225–232, 2016.

[50] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, 2001.

[51] A. Ledent, R. Alves, and M. Kloft, "Orthogonal inductive matrix completion," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[52] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, pp. 173–182, 2017.

[53] W. Cheng, Y. Shen, L. Huang, and Y. Zhu, "Incorporating interpretability into latent factor models via fast influence analysis," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 885–893, 2019.

[54] G. Peake and J. Wang, "Explanation mining: Post hoc interpretability of latent factor models for recommendation systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2060–2069, 2018.

[55] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, and R. Luo, "Heterogeneous graph contrastive learning for recommendation," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pp. 544–552, 2023.

[56] Y. Zhang, X. Chen, *et al.*, "Explainable recommendation: A survey and new perspectives," *Foundations and Trends® in Information Retrieval*, vol. 14, no. 1, pp. 1–101, 2020.

[57] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 565–573, 2018.

[58] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[59] D. Jannach and M. Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," in *Proceedings of the eleventh ACM conference on recommender systems*, pp. 306–310, 2017.

[60] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," *arXiv preprint arXiv:1808.06670*, 2018.

[61] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 776–794, Springer, 2020.

[62] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

[63] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18661–18673, 2020.

[64] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[65] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," *arXiv preprint arXiv:1809.10341*, 2018.

[66] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[67] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," *arXiv preprint arXiv:1810.00826*, 2018.

[68] F.-Y. Sun, J. Hoffmann, V. Verma, and J. Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," *arXiv preprint arXiv:1908.01000*, 2019.

[69] C. Wei, J. Liang, D. Liu, and F. Wang, "Contrastive graph structure learning via information bottleneck for recommendation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 20407–20420, 2022.

[70] X. Cai, C. Huang, L. Xia, and X. Ren, "Lightgcl: Simple yet effective graph contrastive learning for recommendation," *arXiv preprint arXiv:2302.08191*, 2023.

[71] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648, 2020.

[72] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[73] N. Guttenberg, N. Virgo, O. Witkowski, H. Aoki, and R. Kanai, "Permutation-equivariant neural networks applied to dynamics prediction," *arXiv preprint arXiv:1612.04530*, 2016.

[74] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 6240–6249, Curran Associates, Inc., 2017.

[75] A. Ledent, W. Mustafa, Y. Lei, and M. Kloft, "Norm-based generalisation bounds for deep multi-class convolutional neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 8279–8287, May 2021.

[76] P. M. Long and H. Sedghi, "Generalization bounds for deep convolutional neural networks," *arXiv preprint arXiv:1905.12600*, 2019.

[77] S. Arora, S. Du, W. Hu, Z. Li, and R. Wang, "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks," in *International Conference on Machine Learning*, pp. 322–332, PMLR, 2019.

[78] F. Zhu, C. Chen, Y. Wang, G. Liu, and X. Zheng, "Dtcdr: A framework for dual-target cross-domain recommendation," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1533–1542, 2019.

[79] M. Wan, R. Misra, N. Nakashole, and J. McAuley, "Fine-grained spoiler detection from large-scale review corpora," *arXiv preprint arXiv:1905.13416*, 2019.

[80] C. Yuan and H. Yang, "Research on k-value selection method of k-means clustering algorithm," *J*, vol. 2, no. 2, pp. 226–235, 2019.

[81] P. M. Long and H. Sedghi, "Size-free generalization bounds for convolutional neural networks," in *International Conference on Learning Representations*, 2020.

[82] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Adaptive Computation and Machine Learning, Cambridge, MA: MIT Press, 2 ed., 2018.

[83] E. Giné and A. Guillou, "On consistency of kernel density estimators for randomly censored data: Rates holding uniformly over adaptive intervals," *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, vol. 37, pp. 503–522, 07 2001.

[84] E. Platen, "Pollard, d.:convergence of stochastic processes. (springer series in statistics). springer-verlag, new york - berlin - heidelberg - tokyo 1984, 216 pp., 36 illustr., dm 82," *Biometrical Journal*, vol. 28, no. 5, pp. 644–644, 1986.

[85] M. Talagrand, "Sharper bounds for gaussian and empirical processes," *The Annals of Probability*, vol. 22, no. 1, pp. 28–76, 1994.

[86] M. Talagrand, "New concentration inequalities in product spaces," *Inventiones mathematicae*, vol. 126, pp. 505–563, Nov 1996.

APPENDIX

First, recall the following proposition which relates the complexity of a function class to its number of parameters.

Proposition A.1: [81]–[86] Let G be a set of functions from a domain Z to $[0, M]$ such that for some $B > 5$ and for some $d \in \mathbb{N}$ and for some norm $\|\cdot\|_1$ on $\mathbb{R}^d$, there exists a map from the unit ball in $\mathbb{R}^d$ (w.r.t. $\|\cdot\|_1$) to G which is B-Lipschitz with respect to the norms $\|\cdot\|_1$ and $\|\cdot\|_\infty$. For large enough n and for any distribution P over Z, if S is sampled n times independently form P, for any $\delta > 0$, we have with probability $\geqslant 1 - \delta$ that for all $g \in G$,

$$\mathbb{E}_{z\sim P}(g(z)) \leqslant \widehat{\mathbb{E}}_S(g) + CM\sqrt{\frac{d\log(B) + \log(1/\delta)}{n}},$$

where C is some constant.

Back to our situation, if we sample triplets of data, with both our model and our loss function being allowed to depend on the full triplet, we can still apply the above theorem by replacing the sampling distribution P (over individual samples) by a sampling distribution $\mathcal{P}$ over triplets. This yields the following

Proposition A.2: Let $\mathcal{F}$ be a set of functions from a domain Z^3 to $[0, 1]^3$, and let $\ell : (\{1, 2, 3\}, [0, 1]^3) \to [0, 1]$ be a loss function with Lipschitz constant $\frac{1}{\Lambda}$.

Assume there exists a $B > 5$, some $d \in \mathbb{N}$, some norm $\|\cdot\|_1$ on $\mathbb{R}^d$, and a map from the unit ball in $\mathbb{R}^d$ (w.r.t. $\|\cdot\|_1$) to $\mathcal{F}$ which is B-Lipschitz with respect to the norms $\|\cdot\|_1$ and $\|\cdot\|_\infty$. For large enough n and for any distribution $\mathcal{P}$ over the set of combinations of triplets and labels $\{1, 2, 3\} \times Z^3$, if a set of triplets and labels $S = \{(y^i, (e_1^i, e_2^i, e_3^i)) : i \leqslant N\}$ is sampled N times independently form $\mathcal{P}$, for any $\delta > 0$, we have with probability $\geqslant 1 - \delta$ that for all $f \in \mathcal{F}$,

$$\begin{aligned}&\mathbb{E}_{(y,(e_1,e_2,e_3))\sim\mathcal{P}}(\ell(y, f(e_1, e_2, e_3))\\ &\leqslant \widehat{\mathbb{E}}_S(\ell(y, f(e_1, e_2, e_3)) + \sqrt{\frac{d\log\left(\frac{B}{\Lambda}\right) + \log\left(\frac{1}{\delta}\right)}{N}},\end{aligned} \tag{16}$$

where C is some constant.

Proof A.3 (Proof of Proposition A.2): This follows directly from applying proposition A.1 to the composite function class $\mathcal{G} := \ell\circ\mathcal{F}$. Indeed, this function class satisfies the assumptions of the statement of proposition A.1 with Lipschitz constant $\frac{B}{\Lambda}$ and $M = 1$.

A canonical example of such a $1/\Lambda$-Lipschitz loss function would be the following *margin loss*, which we use in Theorem 3.1:

$$\begin{aligned}\ell(y, f(e_1, e_2, e_3)) = 0 \quad &\text{if} \quad \mu(y, f(e_1, e_2, e_3)) \geqslant \Lambda\\ 1 \quad &\text{if} \quad \mu(y, f(e_1, e_2, e_3)) \leqslant 0 \quad \text{and}\\ = 1 - \frac{\mu(y, f(e_1, e_2, e_3))}{\Lambda} \quad &\text{otherwise}\end{aligned} \tag{17}$$

where $\mu(y, f(e_1, e_2, e_3)) :=$

$$\max_{i\leqslant 3} f(e_1, e_2, e_3)_i - \max_{i\leqslant 3, i\neq y} f(e_1, e_2, e_3)_i. \tag{18}$$

Note that in Proposition A.2, both the Lipschitz constant of the parametrization and the margin parameter Λ only show up in logarithmic terms. Furthermore, in the case of a standard CNN, the Lipschitz constant of the parametrization has been shown (cf. proof of Lemmas 2.5 and 3.4) to scale like the product of the spectral norms of the weights of each layer.

Indeed, we have the following simplified version of one of the main results of [81]:

Proposition A.4 (Simplified version of Lemmas 2.5 and 3.6 in [81]): Consider a neural network f defined by the weights $\theta = (\mathrm{A}^1, \mathrm{A}^2, \ldots, \mathrm{A}^L)$, with a componentwise 1-Lipschitz activation function σ, so that

$$f_\theta(x) = \sigma(\mathrm{A}^L\, \sigma(\mathrm{A}^{L-1} \ldots \sigma(\mathrm{A}^1\, x) \ldots)). \tag{19}$$

Let $\theta = (\mathrm{A}^1, \ldots, \mathrm{A}^L), \tilde{\theta} = (\widetilde{\mathrm{A}}^1, \widetilde{\mathrm{A}}^2, \ldots, \widetilde{\mathrm{A}}^L)$ be two possible values of the parameter θ_2 with the property that

$$\prod_{\ell\leqslant L} \|\widetilde{\mathrm{A}}^\ell\|, \prod_{\ell\leqslant L} \|\,\mathrm{A}^\ell\,\| \leqslant \Gamma.$$

We have the following inequality for any x with $\|x\| \leqslant c$:

$$\left\|f_\theta(x) - f_{\tilde{\theta}}(x)\right\| \leqslant \Gamma c \sum_{\ell=1}^{L} \|\,\mathrm{A}^\ell - \widetilde{\mathrm{A}}^\ell\|, \tag{20}$$

where $\|\cdot\|$ denotes the spectral norm.

Proof A.5 (Proof of Proposition A.4): The proof is a simplified version of that of Lemma 2.5 in [81].

Since $\mathrm{A}^l = \widetilde{\mathrm{A}}^l$ for all $l \neq \ell$, we can write $g_\theta = g_{down} \circ g_{\mathrm{A}^\ell} \circ g_{up}$ and $g_{\tilde{\theta}} = g_{down} \circ g_{\widetilde{\mathrm{A}}^\ell} \circ g_{up}$ for two functions g_{up} and g_{down} (depending on $\mathrm{A}^l = \widetilde{\mathrm{A}}^l$ for all the $l \neq \ell$) where $g_{\widetilde{\mathrm{A}}^\ell}$ represents the operation $x \mapsto \sigma(W^\ell x)$ for any $\ell \leqslant L$. We then have

$$\begin{aligned}&\left\|f_\theta(x) - f_{\tilde{\theta}}(x)\right\|\\ &= \left\|g_{down} \circ g_{\mathrm{A}^\ell} \circ g_{up}(x) - g_{down} \circ g_{\widetilde{\mathrm{A}}^\ell} \circ g_{up}(x)\right\|\\ &= \left\|g_{down} \circ \left(g_{\mathrm{A}^\ell} - g_{\widetilde{\mathrm{A}}^\ell}\right) \circ g_{up}(x)\right\|_\ell \|x\|\\ &\leqslant \prod_{l=1}^{\ell-1} \|\,\mathrm{A}^l\,\| \|\,\mathrm{A}^\ell - \widetilde{\mathrm{A}}^\ell\| \prod_{l=\ell+1}^{L} \|\,\mathrm{A}^l\,\| \|x\|\\ &\leqslant c\Gamma \|\,\mathrm{A}^\ell - \widetilde{\mathrm{A}}^\ell\|.\end{aligned} \tag{21}$$

Thus, writing out a telescoping sum, we have

$$\begin{aligned}&\left\|f_\theta(x) - f_{\tilde{\theta}}(x)\right\|\\ &\leqslant \sum_{\ell=1}^{L} \left\|f_{(\mathrm{A}^1,\ldots,\mathrm{A}^\ell,\widetilde{\mathrm{A}}^{\ell+1},\ldots,\widetilde{\mathrm{A}}^L)}(x) - f_{(\mathrm{A}^1,\ldots,\mathrm{A}^{\ell-1},\widetilde{\mathrm{A}}^\ell,\ldots,\widetilde{\mathrm{A}}^L)}(x)\right\|\\ &\leqslant c\Gamma \sum_{\ell=1}^{L} \|\,\mathrm{A}^\ell - \widetilde{\mathrm{A}}^\ell\|,\end{aligned} \tag{22}$$

as expected.

Using this, we can finally provide the proof of our main Theorem:

Proof A.6 (Proof of Theorem 3.1, stated in the main text):

First, note that in Propositions A.1 and A.2, if the map goes from a ball of radius R instead of 1, we can just replace the norm $\|\cdot\|$ by $\frac{\|\cdot\|}{R}$ to reach the same result with the constant B replaced by BR.

In our situation, the parameters $(\chi, W^1, \dots, W^{L_1}, \overline{W}^1, \dots, \overline{W}^{L_2})$ live in a space $\mathbb{R}^{dn+D_1+D_2}$ of dimension $dn + D_1 + D_2$. On that space, we can define a norm by

$$\begin{aligned}\|\theta\| &:= \|(\chi, W^1, \dots, W^{L_1}, \overline{W}^1, \dots, \overline{W}^{L_2})\| \\ &:= \|\chi\|_{\mathrm{Fr}} + \sum_{\ell=1}^{L_1} \|W^\ell\|_{\mathrm{Fr}} + \sum_{\ell=1}^{L_2} \|\overline{W}^\ell\|_{\mathrm{Fr}}. \end{aligned} \tag{23}$$

Claim: For any two sets of parameters

$$\theta_1 = (\chi_1, W_1^1, \dots, W_1^{L_1}, \overline{W}_1^1, \dots, \overline{W}_1^{L_2})$$

and

$$\theta_2 = (\chi_2, W_2^1, \dots, W_2^{L_1}, \overline{W}_2^1, \dots, \overline{W}_2^{L_2}),$$

we have, for any triplet (e_1, e_2, e_3) and label y:

$$\begin{aligned}&|\ell(y, f_{\theta_1}(e_1, e_2, e_3)) - \ell(y, f_{\theta_1}(e_1, e_2, e_3))| \\ &\leqslant [1 + 2(\|P_1\|_{\mathrm{Fr}} + \|P_2\|_{\mathrm{Fr}})]\|\theta\| = 13\|\theta\| \end{aligned} \tag{24}$$

where the matrices P_1, P_2 are defined in equations (1) and (2).

Proof of Claim: As long as we can view the operations performed by our model as a combination of linear operations element wise activation functions, we can apply Proposition A.4. However, the matrices A^l from Proposition must be the matrices representing the full linear operation of each layer expressed as a map from a vector space to another. In our model, we used matrix notation *with the input also being a matrix*: the linear component of each layer is expressed as $X_{out} = WX_{in}$ where X_{in} is a matrix. For instance, at the first layer, $X = \chi e$, where $e \in E^3$, and at the next layer

$$Y = \begin{pmatrix} X & 0 \\ 0 & X \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix}.$$

To apply Proposition A.4, we need to compute the spectral norms of the matrices A^ℓ, which correspond to a translation of the operation $X_{out} = WX_{in}$ in vector notation:he corresponding A^l matrix in the notation of Proposition A.4 is the matrix $\mathrm{op}(W)$ such that $\mathrm{unfold}(X_{out}) = \mathrm{op}(W)(\mathrm{unfold}(X_{in}))$.

Thus, at the first layer, we need the spectral norm of the operation

$$\mathbb{R}^{n\times 3} \ni e \to \chi e \in \mathbb{R}^{n\times 3}, \tag{25}$$

viewed as a map from the vector space $\mathbb{R}^{3n}$ to itself.

Fortunately, the spectral norm can be bounded directly bounded by the Frobenius norm of χ, i.e. $\|\chi\|_{\mathrm{Fr}}$. This is because for any matrices A, B, $\|AB\|_{\mathrm{Fr}} \leqslant \|A\|_{\mathrm{Fr}}\|B\|_{\mathrm{Fr}}$, which implies $\|\chi e\|_{\mathrm{Fr}} \leqslant \|\chi\|_{\mathrm{Fr}}\|e\|_{\mathrm{Fr}}$, or equivalently $\|\mathrm{unfold}(\chi e)\| \leqslant \|\chi\|_{\mathrm{Fr}}\|\mathrm{unfold}(e)\|$, which holds for any e and shows that $\|\mathrm{op}(\chi)\| \leqslant \|\chi\|_{\mathrm{Fr}}$. A similar argument holds for every layer, which explains our use of Frobenius norms for the weights W^ℓ of each layer.

Note that to be able to compare the context vector with the features in X at the distance-based classifier step, we need to create a skip connection from X to the output ρ. Thus the output of layer 2 must be thought of as (X, Y) (rather than Y) and the output of the permutation layer as (X, Z) (rather than Z) etc. However, it is still straightforward to bound the spectral norms of the relevant operators in terms of the Frobenius norms of the weights. Indeed, we certainly have

$$\left\| \begin{pmatrix} X & 0 \\ 0 & X \end{pmatrix} \right\|_{\mathrm{Fr}} \leqslant 2\|X\|_{\mathrm{Fr}} \qquad \text{and then} \tag{26}$$

$$\begin{aligned}\|(X, Y)\|_{\mathrm{Fr}} &\leqslant \|X\|_{\mathrm{Fr}} + 2\|X\|_{\mathrm{Fr}}(\|P_1\|_{\mathrm{Fr}} + \|P_2\|_{\mathrm{Fr}}) \\ &\leqslant \|X\|_{\mathrm{Fr}}(1 + 2\|P_1\|_{\mathrm{Fr}} + 2\|P_2\|_{\mathrm{Fr}}) \\ &= 13\|X\|_{\mathrm{Fr}}. \end{aligned} \tag{27}$$

Similarly, the norm at the ℓth layer of ϕ (resp. $\bar{\phi}$) is bounded by $1 + \|W^\ell\|_{\mathrm{Fr}}$ (resp. $1 + \|\overline{W}^\ell\|_{\mathrm{Fr}}$). This concludes the proof of Claim 1.

To finish the proof of the Proposition, we only need to apply Proposition A.2 with the norm defined in Equation (23).