

Putting the Person in Personalization: Towards Virtual User Identification via Contrastive Session Pair Discrimination

Athiya Deviyani*

Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
adeviyan@cs.cmu.edu

Kevin Zielnicki
Netflix

Los Gatos, California, USA
kzielnicki@netflix.com

Cheng Ju

Netflix
Los Gatos, California, USA
cju@netflix.com

Patric Glynn
Netflix

Los Gatos, California, USA
pglynn@netflix.com

Abstract

Profile sharing is prevalent in streaming services, with surveys indicating that 40% of profiles are shared among multiple users. This challenges personalization and measurement systems that assume a one-to-one mapping between profiles and users. We developed and deployed a contrastive learning approach to distinguish between streaming sessions belonging to the same or different users without requiring direct identity labels. Our system leverages product-generated signals-such as continue-watching patterns and concurrent streaming detection-to construct training data. The resulting session-level embeddings capture meaningful distinctions between user behaviors, paving the way for more precise audience measurement and improved personalization in streaming services.

CCS Concepts

• Information systems → Recommender systems.

Keywords

Personalization, Contrastive Learning, Streaming Services, Representation Learning, Profile Sharing

1 INTRODUCTION

Streaming services face a fundamental challenge in understanding who is actually watching content, as the relationship between profiles and real viewers has become increasingly complex. Internal surveys indicate that 40% of profiles are shared between multiple users. In fact, profile sharing and coupling behaviors from multiple individuals are among the most critical challenges of the recommendation problem in streaming services [1][9]. Most existing recommendation models [12][15][31] implicitly assume a one-to-one mapping between a profile’s watch behavior and a single user and generate recommendation results based on the associated watch behavior. This is illustrated in Figure 1. Therefore, when dealing with profile sharing, these recommender systems may have sub-optimal performance [9][20][7] as the unexpected additional user behaviors introduce noise. This challenge becomes more acute as streaming platforms expand beyond passive video consumption into interactive experiences like games and live events.

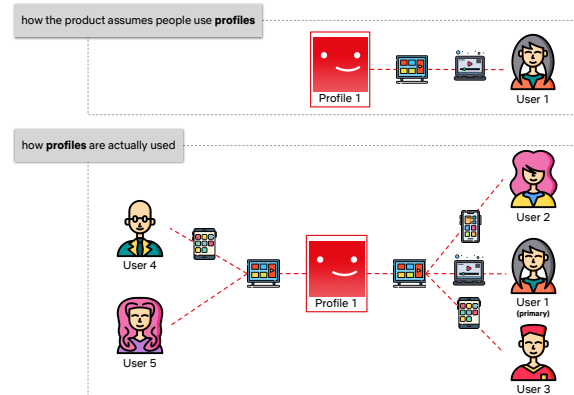


Figure 1: How the product assumes people use profiles vs. how profiles are actually used

While the impact of profile sharing is clear, developing a practical solution faces several fundamental challenges. Obtaining high-quality ground truth about user identity is extremely difficult, even at the coarse profile level. Users rarely explicitly indicate when they’re sharing profiles or watching together. However, streaming platforms generate numerous weak or indirect signals through product interactions - from continue watching patterns to device usage to concurrent streams. The challenge lies in effectively leveraging these weak signals to build robust user identification models. Similar to Qin et al. [21], we use the term ‘weak’ under the following connotations: it is not an actual user identification label, which is not always accurate, and the data is sparse (e.g., very few users watch from more than one TV or mobile device). Additionally, different streaming applications require identity information at vastly different granularities and latencies. Monthly Active Users (MAU) measurement operates at the profile/account level with daily/weekly updates, while real-time personalization demands session-level user identification with sub-second latency. Building separate models for each application would be prohibitively expensive in terms of development resources and maintenance overhead. This necessitates a unified approach that can serve multiple downstream applications efficiently.

Existing approaches to address these problems utilize unsupervised clustering techniques to try and identify users within a profile

*Work done while at Netflix

and use the results to make recommendations [14][19][26][28][29][30]. Other works attempt to train end-to-end models to learn latent user representations under the assumption that there are a fixed number of latent user [7][8][17][23][27]. As noted by Qin et al. [21], these methods overlook exploring how to choose and utilize the indicator of current user identification (useful user behavior information from a session) and the supervision signal of user identification. To this end, to identify users from profile-sharing occurrences, the authors design a session-aware co-attention mechanism that leverages candidate items and session behaviors to attend to historical behaviors in a fine-grained manner.

We present a novel approach that leverages the contrastive learning paradigm to identify virtual users by learning session-level embeddings from behavioral patterns without requiring direct user labels. Our contributions encompass several key innovations:

- We propose a session-level embedding approach using contrastive learning to identify distinct users from their viewing patterns without needing explicit labels. Unlike traditional methods that rely on profile assignments or device fingerprinting, our approach learns to recognize patterns across entire viewing sessions. We show that the system can effectively learn from weak supervision signals despite extreme class imbalance (>90% positive pairs) through careful architectural choices and dynamic loss weighting. Our framework successfully separates user patterns in embedding space even with noisy training signals.
- We establish that a single unified embedding space can effectively serve multiple downstream applications with different timing and granularity needs. Through comprehensive evaluation against ground truth survey data, we achieve 96% AUC in distinguishing between same-user and different-user session pairs. By visualizing the learned embeddings projected to a lower-dimensional space, we observe that they naturally cluster by key behavioral attributes (geographic location, device usage, viewing patterns), revealing clear separation along behavioral dimensions that align with known user characteristics.

2 RELATED WORK

One of the biggest challenges in designing recommender systems for streaming services is the account-sharing phenomena [1][9]. This is due to the fact that existing recommendation models assume a one-to-one mapping between user and account [12][15][31]. Thus, recommendation results are generated based on watching behavior on the account, and may suffer from sub-optimal performance due to noise generated by unexpected watch behavior from additional users [9][20][7]. Streaming services have attempted to circumvent this issue by introducing the notion of ‘profiles’ within accounts to allow users to have separate spaces for different users (or groups of users) within a shared account. However, internal surveys suggest that 40% of profiles are still shared among multiple users. This may be caused by unfavorable pricing incentives, which tend to charge users more for adding multiple profiles to a single account.

This phenomenon suggests the need for quantitative measures to detect profile sharing and explicit user identification. One of the most common approaches is the two-stage model approach used to

improve recommender systems, which first identifies users behind a shared account and then makes recommendations based on identified users. One of the earliest works by Zhang et al. [29] frames the problem as a subspace clustering problem. Wang et al. [26] and Yang et al. [28] extended this approach by analyzing sub-sequence similarity to determine whether sessions belong to the same user. Several works [14][19][30] leverage unsupervised clustering techniques to identify users. While these methods have proven to be effective, they do not leverage explicit feedback data and ignore important sequential patterns in user behaviors, limiting their applicability to streaming services where user sessions inform each other and contain implicit information on user behavior.

Another common approach is to use end-to-end learnable models to learn latent user representations under the assumption that there is a fixed number of latent users. For example, Ma et al. [17] and Sun et al. [23] propose variants of Gated Recurrent Units (GRUs) to filter out information that does not belong to a latent user, while Guo et al. [7] and Guo et al. [8] use the last behavior as an indicator of the current user’s identity. Wen et al. [27] further advance this approach by proposing a self-attention based user identification network that aggregates latent user representations using the last session. Although effective, these models rely on strong assumptions about the fixed number of latent users behind an account and depend solely on click labels as supervision signals, which may not be sufficient for accurate user identification. This limitation is particularly problematic in streaming services, where user behavior is highly dynamic and often lacks explicit feedback.

These methods overlook exploring how to choose and utilize the indicator of current user identification (useful user behavior information from a session) and the supervision signal of user identification, as noted by Qin et al. [21]. To this end, to identify users from profile-sharing occurrences, they proposed a framework with a session-aware co-attention mechanism that leverages candidate items and session behaviors to attend to historical behaviors in a fine-grained manner.

In contrast to these approaches, our work adopts an end-to-end contrastive learning framework that makes fewer assumptions and effectively leverages additional behavioral signals to enhance user identification. Over the years, contrastive learning has been a well-established methodology to learn meaningful representations from weakly labeled data [2][5][11][10]. By learning session-level embeddings from weak supervision signals (e.g., continue-watching patterns, concurrent streaming), our approach avoids the need for explicit user labels or fixed assumptions about the number of users. This allows us to capture subtle behavioral patterns and distinguish between users sharing the same account, even in noisy and imbalanced data.

3 METHODOLOGY

Our approach consists of two main components: (1) a framework for generating and validating training signals from product features, and (2) a contrastive learning model that learns session-level embeddings. Figure 2 shows the overview.

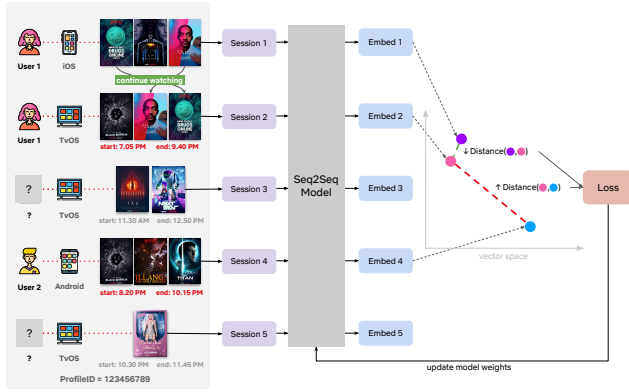


Figure 2: End-to-end framework overview

3.1 Label selection and validation

A key challenge in virtual user identification is the lack of direct ground truth labels about which sessions belong to the same user. To address this, we develop a systematic approach for generating training signals from product-generated session features, where a session is defined from when the user logs in until when the user is inactive for a period of time. In this section, we discuss and investigate the potential of each feature associated with a user session as a weak signal that indicates whether a pair of sessions comes from the same or different user.

3.1.1 Label selection. We identify several product-generated signals indicating whether two sessions likely belong to the same or different users. For positive pairs (same user), we have the following candidate labels:

- **Continue watching:** indicates whether a session contains an instance where a user continues watching a movie or show from the same timestamp. For example, a session indicates a user watches 30 minutes of a movie at 2 PM, then continues from the 30-minute mark at 8 PM. We believe this strongly indicates the same user as people typically resume their content, not others.
- **Same mobile device:** multiple sessions originating from the same mobile device identifier. We believe that mobile devices tend to be personal rather than shared. However, this label might be less reliable for tables shared among family members.

Consequently, for negative pairs (different users), we have the following candidate labels:

- **Concurrent streaming:** indicates whether multiple sessions actively stream different content at the same time, for example, two different shows being watched simultaneously under the same profile. We believe this is a strong indicator of different users, as one person cannot actively watch multiple streams.
- **Different mobile devices:** sessions from different mobile device identifiers. We believe that this could be a strong

signal when devices are used regularly over time. However, this might be less reliable for occasional or temporary device usage.

- **Different location categories:** indicates whether a session comes from a primary or a secondary location.
- **Different device categories:** indicates whether a session comes from a primary or a secondary device.
- **Country ISO code:** country code derived from the client's IP in the order it appeared in a user session. In the case of one user session containing multiple countries, the country that appears most often is chosen.

3.1.2 Label validation. We validate these signals against ground-truth data from a preexisting user survey designed to measure profile sharing. The survey was designed with the following high-level user research goals in mind, namely, to develop a one-time measure of profile sharing for validating user counting models at the session level for several recent sessions, to create a dataset that enables us to train improvements to user counting models and informs efforts around leveraging individual user identification to improve personalization, and to focus on identifying false negatives (i.e. identifying users as not profile-sharing, but they are), such as how often the concurrent streaming rule misses actual profile sharing behavior. The survey was sent to 15,000 users in US, JP, IN, DE, BR, where 40% of the users is an oversample of profile-sharing accounts (accounts with concurrent streaming from the primary profile from more than one device in the past 28 days), and the remainder 60% from the general user population.

While the survey covers several questions intended to provide more insights toward the aforementioned user research goals, we focus on the results to one specific question:

Did you personally watch the following show episode or film between Saturday April 13th and Monday April 15th? Select that you have watched even if you only watched a part of it.

The user is then presented with a series of target titles (show episodes or film watched on primary profile on April 13-April 15) and decoy titles (show episodes or film that are not watched in the last two weeks). The user will answer either "Yes" or "No" to a question regarding whether or not they have watched a particular content within the specified time frame. Let A, and B be two target titles presented to the user. If the user says "Yes" to target title A and target title B, then it is highly likely that the sessions that correspond to A and B belong to the same user (positive pair). Conversely, if the user says "Yes" to target title A and "No" to target title B, then it is highly likely that the sessions that correspond to A and B do not belong to the same user (negative pair). However, if the user says "No" to both target title A and target title B, we cannot infer anything from this information, as there can be more than one other user on the same profile.

To perform label validation, we checked each individual candidate labels for positive and negative pairs discussed in the previous section against the ground truth survey data. In our labeling heuristic, we do not immediately label a session-pair as a negative pair if we do not see an indication that it is a positive pair (for example, if a session-pair does not contain a continue watching signal, it does not automatically indicate that they are a negative pair, i.e. sessions

that belong to different users). Therefore, we validate the different candidate labels based on their recall values, as shown on Table 1.

Label	Description	Recall
Positive	Continue watching	0.9185857931
	Same mobile device	0.2344346345
	Concurrent streaming	0.2736842105
Negative	Different mobile devices	0.3689839572
	Different location categories	0.1516227181
	Different device categories	0.1523232323
	Country ISO code	0.3018867925

Table 1: Recall values for different labels and descriptions

From our validation results, we can observe that there is high confidence in continue watching. However, we do not observe the same from any other candidate labels. Nevertheless, we still need negative labels to train the model using contrastive learning. Therefore, we decided to use the concurrent streaming feature as a negative signal as we have high confidence obtained from previous product understanding and user behavior research. From this, we proceeded with rule-based labeling of the session features and dropped these features and any related features from the training data to avoid any leakage. We obtained a total of 16824 positive pairs and 2848 negative pairs.

3.2 Modeling

3.2.1 Data. We leverage profile session features, which contain information regarding user activity within a session associated with a single profile. We also incorporated additional features from external sources, such as title-to-embedding and title-to-genre mappings. Below are some of the key features in our data:

- **Device:** device ID, device frequency, device type
- **Location:** location frequency, country ISO code
- **Time:** day, hour, timestamp gap, session duration, watch duration
- **Title:** title embedding, genre
- **Recommendation:** impressions from each row, playback from each row
- **Navigation:** search, profile gate, movie details
- **Feedback:** thumb rating, add to list
- **Playback:** continue watching play, my list play, episode selector play
- **Cluster:** multi-household usage cluster

The majority of the features utilized in the model are binary to ensure robustness. The exceptions are the title embeddings and the timestamp gap, which are dense features. Impression counts and watch durations have been transformed into binary features using binning, with cutoffs at the 0, 0.33, 0.75, 0.95, and 1.00 percentiles to handle the long tail. The device and location features are encoded based on their frequency of occurrence (where devices that occur frequently are primary devices while devices that don't are secondary devices). This binary encoding provides an advantage where device and location features are standardized across profiles, i.e., not uniquely identifying different profiles.

Additionally, title embeddings are weighted by the logarithm of the watch duration when multiple show episodes or movies are viewed within a session. Let E_i be the embedding of title i and D_i be the watch duration of title i . We define the weighted embedding W for a session formally as follows:

$$W = \frac{\sum_{i=1}^n E_i \cdot \log(D_i)}{\sum_{i=1}^n \log(D_i)} \quad (1)$$

We also only take into account titles that have been watched for more than 2 minutes, mainly because watching a title for more than 2 minutes represents a stronger expression of interest, and titles viewed for less than 2 minutes often do not have trained embeddings (possibly because they are trailers or highlight reels). For titles viewed longer than 2 minutes, there are still roughly 15% that still lack their corresponding embeddings. In these cases, we generate random embeddings to fill the gaps.

3.2.2 Model architecture and training paradigm. The model architecture is a sequence-to-sequence model [4][24], where each session is represented as a token in the sequence model. with the session-level features encoded into a fixed-length vector. Then, we propose multiple weak-supervised objectives to train the sequence model, with one of them being the session-pair level task involving contrastive learning for session pairs. The output of the model is a session-level embedding. The embeddings corresponding to sessions from the same user should ideally have a smaller distance. In the downstream applications, the sessions should be able to be clustered into multiple groups with each standing for a virtual user.

We train the model on the rule-based pseudo-labeled profile session features described in Section 3.2.1 and perform evaluation on the ground-truth survey data described in Section 3.1.2. We observe that there is a significant data imbalance between the number of positive and negative pairs, where there are only roughly 67,000 negative pairs and over 530,000 positive pairs. We discuss mitigation techniques in Section 3.2.4.

The final training pipeline incorporates both the rule-based labeling heuristic as well as the model training and evaluation procedure.

3.2.3 Loss function. Contrastive learning is a machine learning approach that contrasts positive pairs (similar instances) against negative pairs (dissimilar instances) to learn meaningful data representations [2][10][11][5]. By bringing positive pairs closer and pushing negative pairs farther apart in the representation space, the model captures important features and relationships. This method is used to determine if two sessions belong to the same user. Unlike sequence-to-sequence models, contrastive learning effectively models session similarity. The approach leverages pair-wise information to refine session-level embeddings, improving representation quality without needing pairwise computations during inference.

To this end, we use the following contrastive loss defined by Chopra et al. [5] in our training framework:

$$(Y) (D_W)^2 + (1 - Y) \max(0, m - D_W)^2 \quad (2)$$

- Y : A binary label indicating whether the pair is positive ($Y = 1$) or negative ($Y = 0$).

- D_W : The (cosine) distance between the pair in the feature space.
- m : The margin that defines the minimum distance for negative pairs.

For positive pairs, the loss encourages minimizing the distance between them as much as possible, effectively pulling them together in the feature space. For negative pairs, the loss enforces a minimum margin. If the pair is already sufficiently far apart (greater than the margin), no additional penalty is imposed. However, if they are too close, the loss increases quadratically with how much they violate the margin, thereby pushing them farther apart.

3.2.4 Countering data imbalance. As discussed in Section 3.2.1, we observe an imbalanced proportion between positive and negative pairs when using our rule-based labeling heuristic on the training data. Therefore, we focus our efforts on trying to make the model learn better by countering the effects of data imbalance using the following techniques:

- (1) **Simple architecture:** due to the limited number of negative pairs and labeled session pairs in general, we opted for a straightforward approach by training a simple LSTM [13] and a Transformer [25] with fewer layers and heads. This decision was made to avoid overfitting and to ensure that the model remains efficient and interpretable. By using simpler models, we can focus on the quality of the data and the effectiveness of the training process rather than the complexity of the architecture.
- (2) **Margin parameter tuning:** the margin parameter plays a crucial role in determining the quality of the representations learned by the model. A larger margin can effectively push negative pairs farther apart, which helps in better distinguishing between positive and negative pairs. However, this might lead to slower convergence or even overfitting if the margin is too large. Conversely, a smaller margin might not provide enough separation between negative pairs, potentially leading to less distinct representations.
- (3) **Hard negative mining:** in many cases, focusing on the hardest negative pairs, which are those with the smallest margin violations, can make the learning process more efficient and effective [22]. However, our approach is limited by the fact that we can only learn from profiles that contain both positive and negative session pairs. This constraint means that a significant portion of the data is not utilized for training, as only a few profiles have both types of session pairs available.
- (4) **Loss reweighting:** several methods have been proposed to address class imbalance in contrastive learning by reweighting the loss function based on label frequencies, such as the Focal loss [16], Class-balanced loss [6], and Label-Distribution-Aware-Margin (LDAM) loss [3], and so on. Following the ideas from this work, we use the following modified contrastive loss functions:
 - **Static:** We introduce two new tunable hyperparameters, w^+ and w^- , corresponding to the positive and negative weights. Given the large imbalance in the data, with a limited number of negative pairs per profile, we can penalize the model more heavily when a

negative pair is too close. The revised contrastive loss function is formally defined as follows:

$$w^+ \cdot (Y) (D_W)^2 + w^- \cdot (1 - Y) \max(0, m - D_W)^2 \quad (3)$$

Since the data imbalance presented at roughly 90-10 positive-negative, we experimented with $w^+ = [0.1, 0.01]$ and $w^- = [0.9, 0.99]$.

- **Dynamic:** we recognize the limitations of hard-coding hyperparameter values to our pipeline, as different slices of training data both from past sessions and future sessions might exhibit different levels of imbalances. Therefore, in similar spirit to LDAM loss [3], we propose a variant of weighted max-margin loss where we dynamically assign the positive and negative weights based on the number of positive and negative pairs available in the current batch. Let N^+ and N^- be the number of positive pairs and negative pairs in a batch respectively, we formally define the dynamic max-margin loss as follows:

$$\omega^+ \cdot (Y) (D_W)^2 + \omega^- \cdot (1 - Y) \max(0, m - D_W)^2 \quad (4)$$

Where,

$$\omega^+ = \max \left(1 - \frac{N^+}{N^+ + N^-}, \epsilon \right) \quad (5)$$

$$\omega^- = \max \left(1 - \frac{N^-}{N^+ + N^-}, \epsilon \right) \quad (6)$$

We have included the smoothing parameter ϵ set at 0.01, where a batch contains many positive or negative pairs.

3.2.5 Evaluation. We evaluate the resulting model embeddings through quantitative metrics for assessing discriminative performance and qualitative analyses to understand the semantic structure of the learned embeddings. This dual approach enables us to validate both the model’s ability to distinguish between users and the meaningfulness of its learned representations. We outline each of the evaluation methodologies below:

- **Quantitative performance metrics:** the primary quantitative evaluation metric is the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), which measures the model’s ability to distinguish between same-user and different-user session pairs. For any pair of sessions (s_1, s_2) , we first compute the cosine distance D between their respective embeddings (e_1, e_2) . This distance D serves as our similarity metric between sessions, where smaller distances suggest sessions from the same user and larger distances suggest different users. To evaluate the model’s discriminative ability, we classify session pairs by comparing d against a threshold t : if $D \leq t$: classify as same-user pair, and classify as a different-user pair otherwise. The ROC curve is constructed by varying the distance threshold t and plotting the ratio of correctly classified same-user pairs against incorrectly classified different-user pairs. The AUC score, computed as the area under this curve, provides a threshold-independent assessment of the model’s

discriminative capabilities. An AUC of 1.0 indicates perfect separation between same-user and different-user pairs, while 0.5 indicates random chance performance.

- **Distance distribution analysis:** to understand the model's ability to separate same-user and different-user sessions in embedding space, we analyze the distributions of cosine distances between session pairs. Let $\mathcal{S}$ be the set of all sessions in our validation set. We define positive pairs $\mathcal{P}$ and negative pairs $\mathcal{N}$ as:

$$\mathcal{P} = (s_i, s_j) \in \mathcal{S} \times \mathcal{S} \mid \text{user}(s_i) = \text{user}(s_j) \quad (7)$$

$$\mathcal{N} = (s_i, s_j) \in \mathcal{S} \times \mathcal{S} \mid \text{user}(s_i) \neq \text{user}(s_j) \quad (8)$$

For each pair, we compute the cosine distance: $d(s_i, s_j) = 1 - \cos(e_i, e_j)$ where e_i and e_j are the embeddings of sessions s_i and s_j respectively. This gives us two distributions

$$D_p = d(s_i, s_j) \mid (s_i, s_j) \in \mathcal{P} \quad (9)$$

$$D_n = d(s_i, s_j) \mid (s_i, s_j) \in \mathcal{N} \quad (10)$$

We visualize these distributions using frequency histograms to assess the degree of separation achieved by the model. A successful model should yield D_p clustered near zero (indicating similar embeddings for same-user sessions) and D_n clustered closer to 2 (indicating dissimilar embeddings for different-user sessions), with minimal overlap between the distributions.

- **Embedding space visualization:** To analyze the semantic structure of the learned embedding space, we employ Uniform Manifold Approximation and Projection (UMAP) [18] for dimensionality reduction. Let $E \in \mathbb{R}^{n \times d}$ be the matrix of n session embeddings in d dimensions. UMAP learns a low-dimensional representation $E' \in \mathbb{R}^{n \times 2}$ by optimizing: $E' = \text{UMAP}(E; k = 15, \delta = 0.1)$ where k denotes the number of neighbors considered for local structure preservation and δ represents the minimum distance between points in the low-dimensional space. For each session s_i , we examine external attributes that are not used to label the session pairs. Let $T(s_i) = t_1, \dots, t_k$ be the set of training features including continue watching patterns, and $A(s_i) = a_1, \dots, a_m$ be the set of attributes not used during training such as geographic location $g_i \in G$ and device type $d_i \in D$. The visualization function f maps each reduced embedding e'_i to a point in 2D space: $f : e'_i \mapsto (x_i, y_i, A(s_i))$ where (x_i, y_i) represents the spatial coordinates and $A(s_i)$ determines the visual encoding (e.g., color, shape) of the point. This visualization enables qualitative assessment of the embeddings' semantic structure while providing interpretable insights into the model's learned representations and potential biases. For more interpretable results, we balance the number of positive and negative pairs in the test data.

4 RESULTS

4.1 Model performance

Figures 6 and 7 show the distance distribution histogram and AUC curve for different configurations of architectures and loss functions described in Section 3.2.

We began our experiments by implementing a Transformer model with 6 heads and 8 layers using the max-margin loss, as shown in Figure 6. However, we observed that the model was not training effectively, with the embeddings collapsing to the same values for each session. This issue is likely due to the Transformer architecture being too complex and underfitting the data. Transformers are known to be data-hungry, and the significant class imbalance—where over 90% of the pairs are positive, with most profiles only having positively labeled session pairs—exacerbates this challenge.

To mitigate the effects of data imbalance, we initially applied weighting to the max-margin loss to bias it towards the negative pairs. Unfortunately, this approach did not yield improved performance. We then tested our hypothesis that the architecture was too complex by reducing the number of heads and layers. Simplifying the architecture resulted in a better AUC score and improved separation between the distances of positive and negative pairs. However, some negative pairs were still misclassified, indicating room for further improvement.

Given the results in Figure 6, which suggested that the Transformer architecture was too complex for our use case, we decided to experiment with a 3-layer LSTM. The LSTM architecture captures the contextual nature of profile sessions without the added complexity. As shown in Figure 7, even with the base max-margin loss, the LSTM achieved a decent AUC score and separation between positive and negative pair distances.

We continued experimenting with various techniques to counter data imbalance, as described in Section 3.2.4. Through hyperparameter tuning, we determined that the optimal margin value is 1.5. All techniques employed to address data imbalance appeared to enhance the AUC score and visually improved the separation between positive and negative pair distances, demonstrating their viability as solutions.

Both figures highlight the importance of loss weighting and hard negative mining in improving model performance, especially in the context of data imbalance. When combined with these techniques, simpler models can achieve results comparable to more complex architectures, offering potential computational efficiency benefits. The results underscore the value of tailoring the training process to the specific challenges of the dataset, such as class imbalance, to enhance the model's discriminative capabilities.

These findings validate our approach and demonstrate the potential for efficient and effective virtual user identification in streaming services. The ability of simpler models to achieve results comparable to more complex architectures, when combined with these techniques, highlights the importance of tailoring the training process to the specific challenges of the dataset. These findings validate our approach and demonstrate the potential for efficient and effective virtual user identification in streaming services.

4.2 Embedding analysis

The UMAP embeddings offer a visual representation of the session-level embeddings learned by our model, providing valuable insights into the semantic structure and clustering of user sessions. The

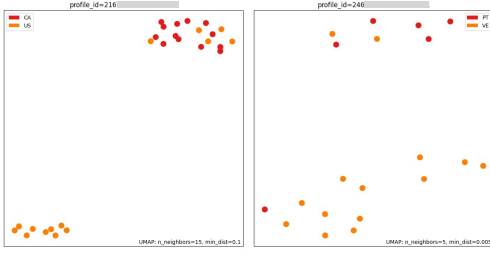


Figure 3: UMAP with Country ISO code labels

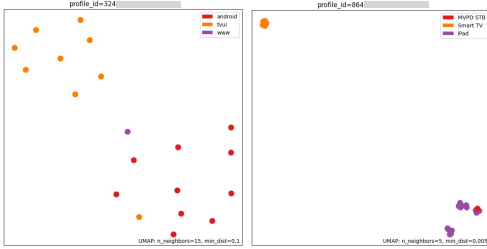


Figure 4: UMAP with Device labels

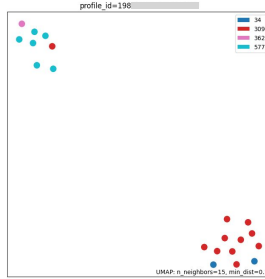


Figure 5: UMAP with MHU cluster ID labels

visualizations reveal distinct clusters of session embeddings, indicating that the model effectively captures the underlying patterns in user behavior.

The UMAP embeddings naturally cluster by key behavioral attributes such as geographic location, device usage, and household cluster. This clustering suggests that the model captures meaningful dimensions of user behavior, which are critical for personalization and user identification. It is important to note that we do not observe the same clustering behavior for an attribute across profiles. For example, if we observe clustering based on country ISO codes for one profile, it doesn't mean we will observe the same clustering on another. This is because profile-sharing behavior is unique for each user: sometimes, they share with cohabitants and stream from different devices. Other times, they might share with family members who live farther away.

Profile observations labeled by country ISO codes in Figure 3 show apparent geographic clustering of user sessions. In the visualization, sessions from different countries form distinct clusters, suggesting that geographic location strongly influences session behavior patterns. This geographic separation emerges naturally from

the learned embeddings despite the model not explicitly trained on location information, indicating that users from different regions exhibit distinguishable viewing behaviors.

In Figure 4, the visualization on the left shows three distinct clusters corresponding to different device types. Android sessions (red) form a tight cluster in the lower right, while TV UI sessions (orange) cluster in the upper left. The single web session (purple) appears isolated, suggesting distinct behavioral patterns across these platforms. This clear separation indicates that users tend to interact with content differently depending on their device, a pattern the model has learned without explicit device-type supervision. The visualization on the right demonstrates similar device-based separation across different viewing platforms. Here, we observe Smart TV sessions forming a dense cluster, while iPad sessions cluster separately with a few MVPD STB sessions nearby. The distinct grouping by device type across different profiles suggests this is a robust pattern learned by our model.

Finally, the visualization in Figure 5 reveals an interesting alignment with household cluster IDs (34, 309, 362, 577). Sessions naturally cluster according to these household groupings, forming two main clusters: one in the upper portion of the plot and another in the lower right. This emergent alignment between our learned embeddings and independently derived household clusters provides strong validation that our model captures genuine user patterns.

The consistency of these clustering patterns across different attributes - geographic location, device type, and household groupings - provides strong evidence that our contrastive learning approach has learned meaningful representations of user behavior. The model discovers natural groupings that align with multiple user characteristics without explicit supervision for any of these attributes, suggesting it has successfully captured underlying patterns in user behavior that can be used for virtual user identification.

5 DISCUSSION

Our experimental results demonstrate several key insights about virtual user identification through session embeddings. First, the challenge of extreme class imbalance (>90% positive pairs) proved to be a significant barrier to model performance, particularly with complex architectures. While Transformers are powerful for sequence modeling, our findings show that architectural simplicity combined with careful handling of class imbalance can be more effective for this task. The success of the LSTM model with dynamic loss weighting suggests that capturing temporal patterns in viewing sessions does not require the full complexity of self-attention mechanisms.

The empirical results highlight an important trade-off between model complexity and data characteristics. Despite having access to millions of streaming sessions, the inherent imbalance in our training signals required us to prioritize robust learning over model capacity. This aligns with our goal of building a practical system that can handle the diverse requirements of streaming applications, from daily user counting to real-time personalization.

The emergence of interpretable patterns in our embedding space is particularly noteworthy. Without explicit supervision, the model learned to organize sessions along multiple behaviorally meaningful dimensions - geographic location, device types, and household

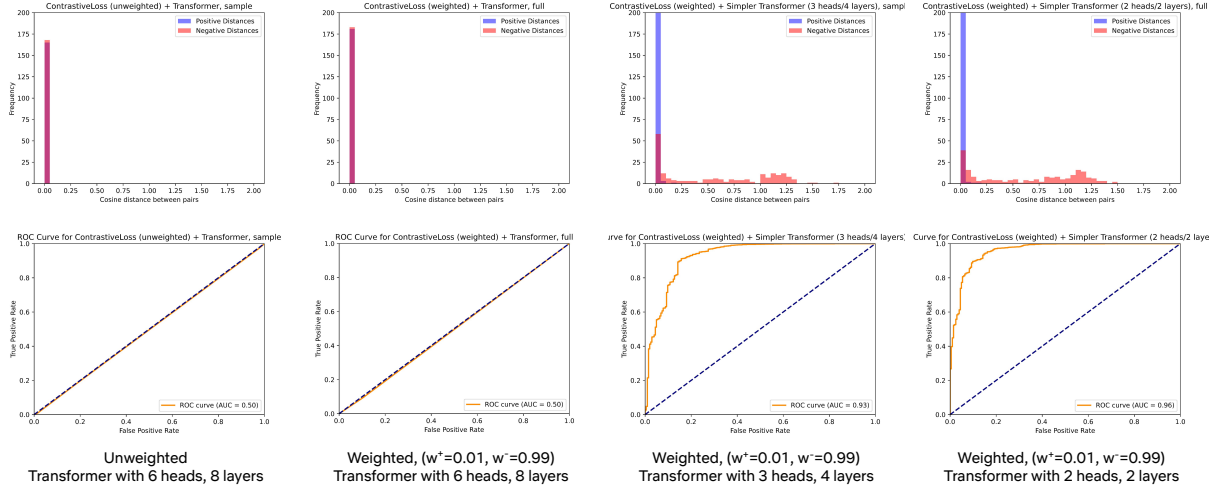


Figure 6: Distance distribution histogram and AUC curve for Transformer architectures

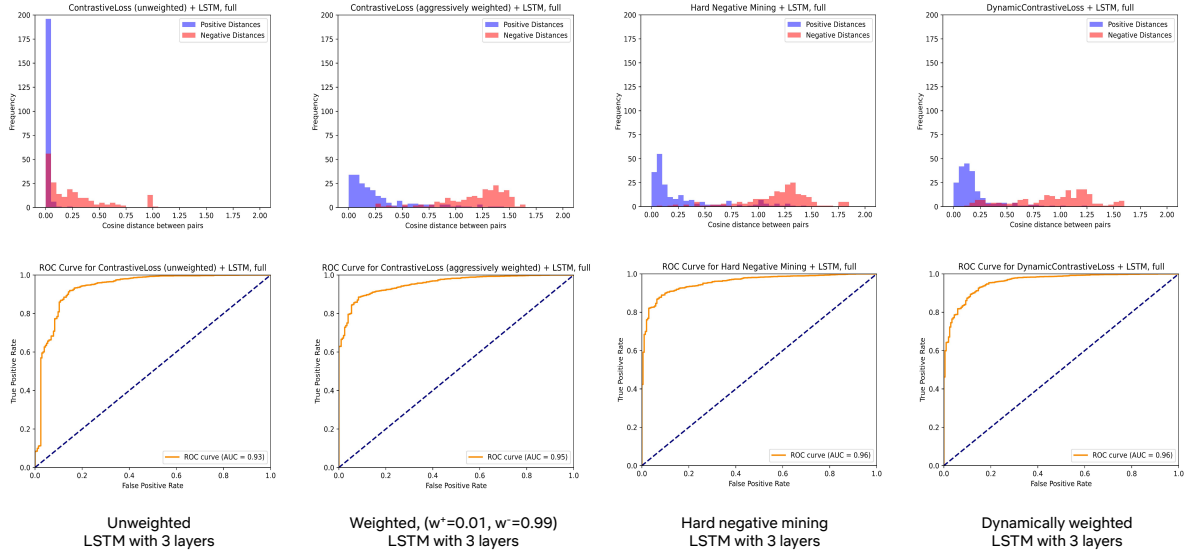


Figure 7: Distance distribution histogram and AUC curve for LSTM architectures

clusters. This natural alignment with known user patterns validates our contrastive learning approach. The fact that different profiles exhibit different dominant clustering patterns (some by geography, others by device) suggests the model captures genuine user behaviors rather than surface-level features.

Our approach effectively addresses the two key challenges identified earlier. First, regarding the broad spectrum of identity needs, the session-level embeddings provide a flexible foundation that can be aggregated for different applications. Second, concerning the lack of high-quality direct signals, our results demonstrate that carefully chosen product-generated signals, combined with appropriate model architecture and training strategies, can effectively learn user patterns despite noisy and imbalanced supervision.

One limitation of our current approach is the potential over-reliance on device and location signals. While these provide strong clustering patterns, they might not capture more subtle behavioral signatures that could help distinguish users sharing the same devices or locations. Additionally, while reflecting user behavior, the varying clustering patterns across profiles could present challenges for downstream applications requiring consistent feature interpretation. Finally, the lack of ground truth data associating each session with a virtual user makes it challenging to perform validation. Currently, the learned session-level embeddings have been integrated into production workflows and are being used in live user-based A/B testing for downstream applications, particularly in audience measurement and user counting tasks.

6 CONCLUSION

We presented a contrastive learning approach for virtual user identification that effectively learns from session-level behavioral patterns without requiring explicit user labels. Through careful handling of class imbalance and architectural simplification, our LSTM-based model successfully distinguishes between same-user and different-user sessions. The learned embeddings demonstrate meaningful clustering by geographic location, device types, and household patterns, validating that our model captures genuine user characteristics while only training on weak supervision signals from product features. Our work provides a foundation for improving personalization in streaming services by focusing on behavioral patterns rather than explicit identity. Future work could explore incorporating additional behavioral signals such as content and subtitle preferences, handling explicit co-viewing scenarios, and evaluating the impact of improved user identification on downstream personalization tasks.

References

- [1] Michal Aharon, Eshcar Hillel, Amit Kagian, Ronny Lempel, Hayim Makabee, and Raz Nissim. 2015. Watch-it-next: a contextual TV recommendation system. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7–11, 2015, Proceedings, Part III* 15. Springer, 180–195.
- [2] Suzanna Becker and Geoffrey E Hinton. 1992. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature* 355, 6356 (1992), 161–163.
- [3] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/621461af90cadfdaf0e8d4cc25129f91-Paper.pdf
- [4] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.), Association for Computational Linguistics, Doha, Qatar, 1724–1734. <https://doi.org/10.3115/v1/D14-1179>
- [5] S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. 539–546 vol. 1. <https://doi.org/10.1109/CVPR.2005.202>
- [6] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-Balanced Loss Based on Effective Number of Samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9260–9269. <https://doi.org/10.1109/CVPR.2019.00949>
- [7] Lei Guo, Jinyu Zhang, Tong Chen, Xinhua Wang, and Hongzhi Yin. 2022. Reinforcement learning-enhanced shared-account cross-domain sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 7 (2022), 7397–7411.
- [8] Lei Guo, Jinyu Zhang, Li Tang, Tong Chen, Lei Zhu, and Hongzhi Yin. 2022. Time interval-enhanced graph neural network for shared-account cross-domain sequential recommendation. *IEEE Transactions on Neural Networks and Learning Systems* 35, 3 (2022), 4002–4016.
- [9] Wei Guo, Can Zhang, Zhicheng He, Jiarui Qin, Huifeng Guo, Bo Chen, Ruiming Tang, Xiquiang He, and Rui Zhang. 2022. Miss: Multi-interest self-supervised learning framework for click-through rate prediction. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 727–740.
- [10] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings*, 297–304.
- [11] R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. 1735–1742. <https://doi.org/10.1109/CVPR.2006.100>
- [12] B Hidasi. 2015. Session-based Recommendations with Recurrent Neural Networks. *ICLR* (2015).
- [13] S Hochreiter. 1997. Long Short-term Memory. *Neural Computation* MIT-Press (1997).
- [14] Jyun-Yu Jiang, Cheng-Te Li, Yian Chen, and Wei Wang. 2018. Identifying users behind shared accounts in online streaming services. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 65–74.
- [15] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 447–456.
- [16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2020. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 2 (2020), 318–327. <https://doi.org/10.1109/TPAMI.2018.2858826>
- [17] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. π -net: A parallel information-sharing network for shared-account cross-domain sequential recommendations. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 685–694.
- [18] Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. [arXiv:1802.03426 \[stat.ML\]](https://arxiv.org/abs/1802.03426) <https://arxiv.org/abs/1802.03426>
- [19] Carolina Nery, Renata Galante, and Weverton Cordeiro. 2021. FIP-SHA-Finding Individual Profiles Through SHared Accounts. In *International Conference on Database and Expert Systems Applications*. Springer, 115–126.
- [20] Lyndon Nixon, Jeremy Foss, Konstantinos Apostolidis, and Vasileios Mezaris. 2022. Data-driven personalisation of television content: a survey. *Multimedia Systems* 28, 6 (2022), 2193–2225.
- [21] Jiarui Qin, Jiachen Zhu, Yankai Liu, Junchao Gao, Jianjie Ying, Chaoxiong Liu, Ding Wang, Junlan Feng, Chao Deng, Xiaozheng Wang, et al. 2023. Learning to distinguish multi-user coupling behaviors for tv recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 204–212.
- [22] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive Learning with Hard Negative Samples. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=CR1XOQ0UTh>
- [23] Wenchao Sun, Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2021. Parallel split-join networks for shared account cross-domain sequential recommendations. *IEEE Transactions on Knowledge and Data Engineering* 35, 4 (2021), 4106–4123.
- [24] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f2747c2d894ec1c3c743d2-Paper.pdf
- [25] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).
- [26] Zhijin Wang, Yan Yang, Liang He, and Junzhong Gu. 2014. User identification within a shared account: Improving IP-TV recommender performance. In *Advances in Databases and Information Systems: 18th East European Conference, ADBIS 2014, Ohrid, Macedonia, September 7–10, 2014. Proceedings* 18. Springer, 219–233.
- [27] Xinyu Wen, Zhaohui Peng, Shanshan Huang, Senzhang Wang, and Philip S Yu. 2021. MISS: A multi-user identification network for shared-account session-aware recommendation. In *Database Systems for Advanced Applications: 26th International Conference, DASFAA 2021, Taipei, Taiwan, April 11–14, 2021, Proceedings, Part III* 26. Springer, 228–243.
- [28] Yan Yang, Qinmin Hu, Liang He, Minjie Ni, and Zhijin Wang. 2015. Adaptive temporal model for IPTV recommendation. In *Web-Age Information Management: 16th International Conference, WAIM 2015, Qingdao, China, June 8–10, 2015. Proceedings* 16. Springer, 260–271.
- [29] Amy Zhang, Nadia Fawaz, Stratis Ioannidis, and Andrea Montanari. 2012. Guess who rated this movie: Identifying users through subspace clustering. *arXiv preprint arXiv:1208.1544* (2012).
- [30] Wei Zhang and Chris Challis. 2020. Automatic Identification of Account Sharing for Video Streaming Services. In *Trends in Artificial Intelligence Theory and Applications. Artificial Intelligence Practices: 33rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2020, Kitakyushu, Japan, September 22–25, 2020, Proceedings* 33. Springer, 162–173.
- [31] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (London, United Kingdom) (KDD '18)*. Association for Computing Machinery, New York, NY, USA, 1059–1068. <https://doi.org/10.1145/3219819.3219823>