Contrastive Learning of Temporal and Event-Based Behavioral Views for Universal User Embeddings

Yuuki Tachioka*
* Denso IT Laboratory, Japan
E-mail: tachioka.yuki@core.d-itlab.co.jp

***Abstract*—Modern recommender systems often tackle diverse downstream tasks such as product recommendation, churn prediction, and customer lifetime value estimation using independently trained task-specific models. However, this fragmented modeling approach incurs high development cost and struggles to address the cold start problem or data sparsity in long-tail user behaviors. To address these challenges, we propose a self-supervised framework for learning *universal user embeddings* from multi-event behavioral logs. Our approach models user behavior as views generated from a shared latent user vector via event-specific and time-window-specific transformations. We train the embeddings estimation model through a two-stage contrastive learning strategy: (1) intra-event contrast aligning different time slices of the same event, and (2) cross-event contrast aligning different event types. Additionally, we introduce auxiliary tasks, the prediction of the next item and the unseen term, to inject task-relevant semantics into the learned representations. Experimental results in the RecSys Challenge 2025 dataset demonstrate that the resulting universal embeddings achieve strong generalization across diverse downstream tasks, including cold start and long-tail scenarios.**

I. Introduction

The applications of recommender systems span a wide range of tasks, including purchase recommendation, churn prediction, and the estimation of customer lifetime value [1]. Although these tasks differ in their objectives, they generally depend on predicting user preferences or future actions based on behavioral histories [2], [3]. In practice, it is typical to develop and deploy task-specific models individually optimized for each use case. However, such task-specific modeling approaches entail several limitations. First, the design, training, and maintenance of separate models for each task impose high development and operational costs, resulting in inefficiencies [4], [5]. Second, the cold-start problem, where limited data is available for new users or items, remains a persistent challenge [6]. Moreover, collaborative filtering methods tend to learn item-dependent representations, making knowledge transfer to other tasks difficult [7]. In domains or events with extremely sparse data, model training becomes unstable [8], [9].

To address these challenges, recent efforts have focused on unified frameworks capable of handling multiple recommendation-related tasks. These works suggest that the capture of universal user preferences across domains or tasks can enhance cold start performance and facilitate transfer learning [10], [11], [12], [13], [14], [15]. Although some studies use auxiliary user attributes such as age and sex [16], the use of such data on a scale is limited due to privacy concerns and data collection costs.

Given these challenges, we propose a self-supervised framework for learning *universal user embeddings* solely from heterogeneous behavioral logs, without relying on explicit user attributes or task-specific supervision. This direction aligns with recent trends, as seen in RecSys Challenge 2025 [17], where general-purpose user representations are learned from event logs alone. A common baseline for such universal embeddings, which is also adopted in the challenge, involves segmenting user logs into multiple time windows and representing users by the IDs of top-N items appearing in each window [18], [19]. This approach is simple yet effective in practice, producing highly sparse and memory-efficient vectors. Unlike the USER framework [11], which integrates heterogeneous behaviors into a single sequential representation, our method explicitly models multi-event and multi-temporal views. While prior works [12] apply contrastive learning to user behavior sequences in single view, our two-stage contrastive learning strategy is designed to unify heterogeneous multi-event logs such as search, cart, and purchase behaviors. By explicitly modeling event-specific inverse mappings and aligning embeddings across event types and time windows, our approach captures richer representations of user intent. Our framework extracts task-agnostic user representations from multi-event, multi-temporal views, offering improved expressiveness while retaining scalability. The contributions are threefold:

- A two-stage contrastive learning framework that derives general-purpose user embeddings by aligning intra-event consistency and cross-event intent across heterogeneous behavioral histories.
- A joint optimization scheme that combines user-level identification within events and identity preservation across events.
- Auxiliary objectives (next-item and unseen-term prediction) that inject task-relevant semantics, ensuring adaptability to diverse downstream tasks and universality across diverse behavioral logs.

In this way, the method is agnostic to the specific nature of behavioral events and can be applied not only to purchases or searches but also to other forms of interaction logs (e.g., clicks, media consumption, reviews). We formally define the

979-8-3315-7206-8/25/$31.00

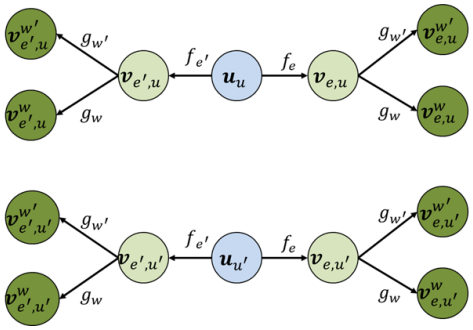

Fig. 1. Graphical model of the Universal user embedding framework. Each user vector $\boldsymbol{u}_u$ generates event-specific views via functions f_e, which are sliced into time windows via g_w to produce the final observations $\boldsymbol{v}_{e,u}^w$. The objective is to recover the user embedding across views and events.

Fig. 2. Estimated user embedding vector obtained by concatenating time window-wise and event-wise approximations $\boldsymbol{u}_u(\boldsymbol{v}_{e,u}^w)$ across events and time windows.

proposed Universal User Embedding Model in Sec. II and we propose a two-stage contrastive learning framework in Sec. III.

II. Universal User Embedding Model

Universal user embedding refers to a compact, behaviorally derived representation that supports zero-shot transferability across diverse recommendation tasks, regardless of event type or time window. To handle various types of user behavior histories in a unified manner, we assume a generative model centered around a user embedding vector $\boldsymbol{u}_u$. As illustrated in Fig. 1, this framework is modeled as a graphical model in which observable behaviors are generated from a latent user representation. For a given user u, the behavioral log $\boldsymbol{v}_{e,u}$ associated with event type e (e.g., product views, cart additions, purchases) is assumed to be generated from the user-specific vector $\boldsymbol{u}_u$ as follows:

$$\boldsymbol{v}_{e,u} = f_e(\boldsymbol{u}_u). \tag{1}$$

Furthermore, behavioral histories are segmented into different time windows w (e.g., the last 5 days, the last 30 days), with data for each period denoted as $\boldsymbol{v}_{e,u}^w$:

$$\boldsymbol{v}_{e,u}^w = g_w(\boldsymbol{v}_{e,u}), \tag{2}$$

where g_w is a slice operation that extracts a temporal segment of the behavior log, acting essentially as an indicator function on the timeline. The learning objective is to recover the original user embedding $\boldsymbol{u}_u$ from the observed behavior slices $\boldsymbol{v}_{e,u}^w$ by learning approximate inverse functions f_e^{-1} and g_w^{-1}:

$$\boldsymbol{u}_u \approx \boldsymbol{u}_u(\boldsymbol{v}_{e,u}^w) = f_e^{-1}(g_w^{-1}(\boldsymbol{v}_{e,u}^w)), \tag{3}$$

where we denote the approximate embedding derived from $\boldsymbol{v}_{e,u}^w$ as $\boldsymbol{u}_u(\boldsymbol{v}_{e,u}^w)$.

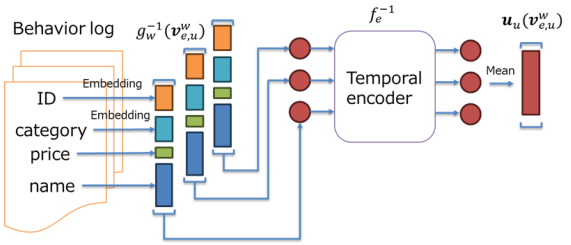

Fig. 3. Estimation model of the user embedding vector $\boldsymbol{u}_u(\boldsymbol{v}_{e,u}^w)$ from behavior log $\boldsymbol{v}_{e,u}^w$. For each event e, the model processes logs of user u from a time window w using event-specific temporal encoders and averages them into a fixed-dimensional representation.

The final user vector $\boldsymbol{u}'_u$ is obtained by concatenating approximated embeddings over events $e_1, e_2, \ldots$ and time windows $w_1, w_2, \ldots$ (Fig. 2),

$$\boldsymbol{u}'_u = \left[\boldsymbol{u}_u(\boldsymbol{v}_{e_1,u}^{w_1}), \boldsymbol{u}_u(\boldsymbol{v}_{e_1,u}^{w_2}), \ldots, \boldsymbol{u}_u(\boldsymbol{v}_{e_2,u}^{w_1}), \boldsymbol{u}_u(\boldsymbol{v}_{e_2,u}^{w_2}), \ldots\right], \tag{4}$$

or optionally averaged to reduce dimensionality (Eqs. (5)–(7)) because all components essentially approximate the same latent user vector. Averaging over time windows:

$$\boldsymbol{u}'_u = \frac{1}{W}\left[\sum_w \boldsymbol{u}_u(\boldsymbol{v}_{e_1,u}^w), \sum_w \boldsymbol{u}_u(\boldsymbol{v}_{e_2,u}^w), \ldots\right]. \tag{5}$$

Averaging over events:

$$\boldsymbol{u}'_u = \frac{1}{E}\left[\sum_e \boldsymbol{u}_u(\boldsymbol{v}_{e,u}^{w_1}), \sum_e \boldsymbol{u}_u(\boldsymbol{v}_{e,u}^{w_2}), \ldots\right]. \tag{6}$$

Averaging over both:

$$\boldsymbol{u}'_u = \frac{1}{WE}\sum_w \sum_e \boldsymbol{u}_u(\boldsymbol{v}_{e,u}^w), \tag{7}$$

where W is the number of time windows and E is the number of events. This composite embedding vector is then passed to downstream tasks. Inverse mappings f_e^{-1} and g_w^{-1} required to recover user embeddings from behavior logs $\boldsymbol{v}$ are trained using the contrastive learning framework described in the next section.

III. Two-stage Contrastive Learning

Fig. 3 illustrates the model used to estimate the approximate inverse mapping f_e^{-1}. For each event e, behavior logs within a given time window w are processed into vectors, which are then fed into a temporal encoder to derive intermediate representations. The encoder outputs are aggregated via mean pooling to ensure a fixed-size embedding across variable-length behavior sequences.

Categorical features such as item ID or category are embedded in fixed-length vectors, while prices are quantized. Textual features such as product names or search queries are assumed to be given in embedded language vector form. Each event type has its own independently trained temporal encoder. Based on this architecture, we implement a two-stage contrastive learning framework: First stage (intra-event):

Align user embeddings across time windows within the same event type. Second stage (cross-event): Align user embeddings across different event types. The following subsections describe each stage in detail.

A. First Stage: Intra-event Embedding Alignment

First, we align user embeddings within the same event type e such that embeddings from the same user are close, while embeddings from different users are distant. The time windows w and w' can differ. Specifically, for the same user u, the embeddings should be similar:

$$f_e^{-1}(g_w^{-1}(\boldsymbol{v}_{e,u}^{w})) \approx f_e^{-1}(g_{w'}^{-1}(\boldsymbol{v}_{e,u}^{w'})), \tag{8}$$

and for different users $u \neq u'$, the embeddings should be distinct:

$$f_e^{-1}(g_w^{-1}(\boldsymbol{v}_{e,u}^{w})) \neq f_e^{-1}(g_w^{-1}(\boldsymbol{v}_{e,u'}^{w'})). \tag{9}$$

This stage independently learns the inverse mapping f_e^{-1} for each event e, encouraging the encoder to extract consistent user-specific features over different time periods w.

B. Second Stage: Cross-event Embedding Alignment

Second, we further align embeddings across different event types e and e', encouraging their representations to be close even when derived from different behavior types:

$$f_e^{-1}(g_w^{-1}(\boldsymbol{v}_{e,u}^{w})) \approx f_{e'}^{-1}(g_{w'}^{-1}(\boldsymbol{v}_{e',u}^{w'})). \tag{10}$$

This stage serves to globally align event-specific embedding spaces at the user level. In our experiments, we consider events such as product views, cart additions/removals, and purchases, as summarized in Table I. Although these actions differ semantically, they all reflect varying degrees of user interest and thus can be mapped to a shared representation space.

C. Loss Function with Auxiliary Tasks

To approximate inverse mappings f_e^{-1}, we adopt temporal representation learning methods that can extract stable user-level features from behavioral sequences, such as GRU [20] or Transformer Encoder [21]. For both stages of contrastive learning, we use InfoNCE loss [22], particularly in the SimCLR form, which contrasts each positive pair with all other samples in the batch as negatives.

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N}\sum_{i=1}^{N}\log\left[\frac{\exp\left(\text{sim}(\boldsymbol{h}_i, \boldsymbol{h}_i^{+})/\tau\right)}{\sum_{j=1}^{N}\exp\left(\text{sim}(\boldsymbol{h}_i, \boldsymbol{h}_j)/\tau\right)}\right]. \tag{11}$$

Here, $\boldsymbol{h}_i$ is an embedding vector, $\boldsymbol{h}_i^{+}$ is its positive counterpart, N is the batch size, and τ is a temperature parameter. We use cosine similarity for the similarity function sim.

To further improve the predictive power of the embeddings, we introduce the following auxiliary tasks: next-item prediction where for the next-item ID or URL, a classification loss (cross-entropy) is used and for queries, mean squared error loss is used, and unseen term prediction that predicts the number of weeks a user has not used the service as a classification task.

The final training loss combines these auxiliary losses with the contrastive loss using weighted averaging.

$$\mathcal{L} = \mathcal{L}_{\text{InfoNCE}} + \lambda_1\mathcal{L}_{\text{NextItem}} + \lambda_2\mathcal{L}_{\text{Unseen}}, \tag{12}$$

where λ_1 and λ_2 are weighting coefficients for the auxiliary tasks.

TABLE I
FIVE TYPES OF USER BEHAVIOR EVENTS.

Event	Features
product_buy (buy)	ID, category, price, name (16-dim)
add_to_cart (add)	ID, category, price, name (16-dim)
remove_from_cart (rm)	ID, category, price, name (16-dim)
page_visit (visit)	page ID
search_query (search)	query (16-dim)

TABLE II
DOWNSTREAM TASKS AND EVALUATION METRICS. NOTE THAT HIDDEN TASKS ARE DISCLOSED AFTER THE CHALLENGE ENDS.

Task	Evaluation Metric	# of Classes
Churn prediction	AUROC	2
Category propensity	Eq. (13)	100
Product propensity	Eq. (13)	100
hid1 (conversion prediction)	AUROC	2
hid2 (new product propensity)	Eq. (13)	20
hid3 (price propensity)	Eq. (13)	100

IV. EXPERIMENT

A. Experimental Setup

We validated the effectiveness of our proposed method using the dataset provided by RecSys Challenge 2025 [17]. This dataset consists of hundreds of millions of real user interaction logs, covering the five types of events shown in Table I. Each product is annotated with a 100-quantized price class and name. Names and search queries are embedded into 16-dimensional vectors using a large language model. The evaluation covers three public tasks: churn prediction, category propensity, and product propensity along with three hidden tasks, as summarized in Table II. Evaluation metrics include AUROC and a weighted score that combines AUROC with Novelty and Diversity.

$$\text{Score} = 0.8 \times \text{AUROC} + 0.1(\text{Novelty} + \text{Diversity}), \tag{13}$$

TABLE III
DATASET STATISTICS. VALUES IN PARENTHESES IS THE NUMBER AFTER ID MERGE.

Item	Value
# of users	1,000,000
# of product SKUs	1,178,412 (584,061)
# of product categories	6,774 (5,893)
# of URL references	12,650,786 (2,483,437)
# of logs (buy/add/rm)	1,775,394 / 5,674,064 / 1,937,170
# of logs (visit/search)	156,032,014 / 10,218,831

where novelty measures how many items recommended by the system were previously unseen by the user, and diversity captures the variety of item categories and properties in the recommendations. These metrics ensure that user experience factors such as surprise and exploration value are also considered, beyond simple accuracy. Due to the rules of the RecSys Challenge 2025, the evaluation server only returns average scores for each submission. Importantly, it does not provide per-sample correctness labels or variance across runs. Therefore, it is not possible to compute statistical significance tests or report confidence intervals.

The data is chronologically partitioned to separate the training period from the evaluation period, thus preventing information leakage. The final evaluations are conducted by the challenge organizers in a closed setting. Due to submission limits, exhaustive experimentation is not feasible. Table III provides statistics on the dataset used for training. As shown, the dataset is large-scale and realistic. In our approach, user behavior logs are segmented into four time windows $w = \{7, 15, 30, 150\}$ [days], and embeddings are learned for each type of event within these time windows w. For each pair (e, w), we estimate a 100-dimensional embedding $\boldsymbol{v}_{e,u}^{w}$, which is then concatenated into a 2000-dimensional vector $\boldsymbol{u}_u'$. In the challenge regulation, the embedding dimensionality is limited to 2048.

Fig. 4 illustrates the histogram of stock keeping unit (SKU) frequencies in the `add` event. The left panel shows the original distribution that exhibits a long-tailed property, with many items appearing only 1–10 times. To reduce the dimensionality of the one-hot representations for items and URLs, we merge rare items into shared IDs in batches of five if the point per item is less than five, where the point per purchase is 2 and that per cart operation is 1. Similar aggregation is applied to categories (minimum 4 occurrences) and URLs (minimum 10 views). This ID merge strategy reduces the size of the embedding space while preserving discriminability for popular entities, thereby improving memory efficiency. The right panel shows the merged distribution after thresholding, which maintains the overall shape while reducing the total number of IDs.

We use a two-layer Transformer model for learning, optimized using Adam. For each user, up to 1000 most recent events are used per time window. The temperature parameter tau in (11) is set to 0.1. Training continues for 20 epochs in the first stage and 100 epochs in the second stage, with a batch size of 100. In the second stage, event pairs are uniformly sampled at random from the five event types (buy, add, rm, visit, search). This includes both homogeneous pairs (e.g., `buy–buy`) and heterogeneous pairs (e.g., `add–visit`). This random pairing strategy ensures broad cross-event alignment without bias toward specific event types. The loss weights are set as $\lambda_1 = \lambda_2 = 1.0$. The same embedding modules are shared across all event types for SKU IDs and categories. The embedding dimensionalities are set to 32 for SKUs and URLs, and 8 for categories.

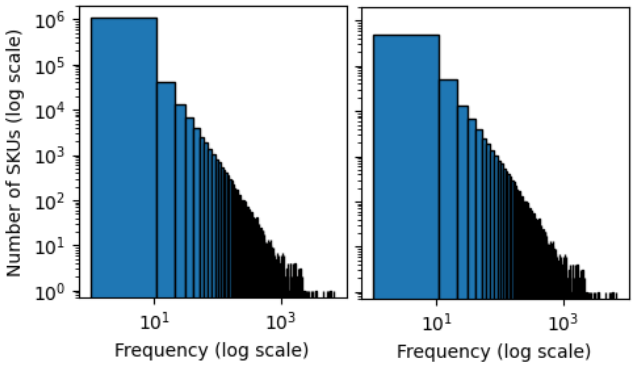

Fig. 4. Histogram of the SKU frequencies in the `add_to_cart` event. The left panel shows the original distribution, while the right panel shows the distribution after ID merge.

TABLE IV
PERFORMANCE COMPARISON BETWEEN BASELINE, GRU ENCODER, AND TRANSFORMER ENCODER (TF) WITH AND WITHOUT ID MERGE.

Model	Churn	Prop. (Cat)	Prop. (SKU)	hid1	hid2	hid3
Baseline	0.6947	0.6985	0.6919	0.6579	0.7382	0.7207
GRU	0.7035	0.7747	0.7653	0.7163	0.7342	0.7749
GRU w/ ID merge	**0.7183**	0.7845	0.7798	0.7341	**0.7564**	0.7873
TF w/ ID merge	0.7174	**0.7876**	**0.7804**	**0.7375**	0.7390	**0.7885**

B. RQ: Does the choice of temporal encoder affect performance?

Table IV shows the comparison of our proposed method with the baseline. Our method outperforms the baseline in all tasks. The improvement is significant in propensity prediction tasks (category and SKU), where the embedded user knowledge better captures purchasing tendencies. These results demonstrate that our self-supervised embeddings, trained solely from behavior logs without any task-specific labels, are effective across a range of downstream tasks. In addition, ID merge, which clusters and merges long-tailed, low-frequency items, improves the performance for all tasks.

The performance difference between GRU and Transformer is marginal. This suggests that the core effectiveness of our approach is not primarily derived from the encoder architecture itself but rather from the overall contrastive learning design. The high quality of the learned representation allows even relatively simple encoders to perform competitively.

Lastly, the hidden tasks (hid1–hid3) represent unreleased evaluation scenarios provided by the challenge organizers. Even in these unseen tasks, our method outperforms the baseline. This is particularly significant, as it implies that the learned user embeddings possess a degree of generality and robustness that enables them to be applied in a zero-shot manner. Such versatility is highly valuable in real-world recommendation systems, where task diversity and evolving objectives are common. The results validate the proposed framework as a viable solution for learning reusable, task-independent user representations.

TABLE V
FINAL WEIGHTED LOSS (12) BETWEEN EVENT-SPECIFIC EMBEDDINGS (VALUES IN PARENTHESES SHOW LOSS WITHOUT FIRST STAGE). AS THE LOSS MATRIX IS SYMMETRIC, LOWER TRIANGLE VALUES ARE OMITTED.

	buy	add	rm	visit	search
buy	5.49 (5.64)	6.11 (6.27)	6.92 (6.83)	6.70 (6.78)	8.90 (8.79)
add		5.61 (5.86)	6.32 (6.43)	6.23 (6.19)	8.73 (8.77)
rm			5.53 (5.77)	6.44 (6.37)	8.75 (8.67)
visit				2.59 (2.88)	6.77 (6.69)
search					4.52 (4.54)

C. RQ: Does the two-stage training reduce contrastive loss?

Table V shows the final losses of our two-stage contrastive learning and contrastive learning without the first stage. The proposed two-stage contrastive learning framework consistently reduces intra-event loss (diagonal entries), indicating that stable and self-consistent user representations are being learned for each type of event. The `visit` event shows a substantial improvement in loss from 2.88 to 2.59, because `visit` is one of the most frequent and widely distributed behaviors, which suggests that the model can robustly extract user-specific embeddings even from high-entropy interactions. This confirms the effectiveness of the first-stage learning that anchors behavior patterns unique to each user. In contrast, cross-event losses (off-diagonal entries) show relatively minor changes overall. However, improvements are observed for pairs of semantically related events, for example, between `add` and `buy`, and between `add` and `rm`. These reductions reflect the model's ability to encode behavioral continuity and correlations across related actions. This is a desirable outcome of the second stage cross-event alignment, which successfully captures shared user preference structures across different types of events.

On the other hand, for semantically distinct events like `search` and `visit`, the loss values remain relatively high when paired with more outcome-oriented behaviors such as `buy`, i.e., pairings like `search-buy` or `visit-buy` are meaningfully separated, preserving the interpretability of the learned embeddings. This indicates that the model avoids overly blending unrelated behavioral patterns, maintaining a balance between generalization and discriminability. The comparison with the model trained without pretraining (shown in parentheses) further confirms that the proposed method consistently reduces intra-event loss without sacrificing meaningful separation across events. This structural coherence validates the framework's capacity to learn generalizable user embeddings that span heterogeneous event types.

Moreover, the method is scalable: training embeddings for one million users are completed in approximately five hours over 120 epochs on a single RTX8000 GPU. Inference of the embeddings post-training requires only about two hours without parallel computing. These characteristics highlight the practicality of the approach for large-scale industrial deployment.

TABLE VI
EFFECT OF EMBEDDING DIMENSIONALITY AND AGGREGATION METHOD (TIME WINDOW-WISE (W) VS. EVENT-WISE (E)) ON DOWNSTREAM TASK PERFORMANCE.

Dim	Churn	Prop. (Cat)	Prop. (SKU)	hid1	hid2	hid3
100	0.7183	0.7845	**0.7798**	0.7341	0.7564	0.7873
2000 (w/e avg)	0.7178	0.7839	0.7742	0.7336	0.7484	0.7849
400 (w avg)	0.7173	0.7855	0.7732	**0.7358**	0.7348	**0.7892**
400 (e avg)	**0.7189**	**0.7863**	0.7789	0.7344	**0.7603**	0.7868

TABLE VII
PROPORTION OF ZEROS IN USER EMBEDDINGS (I.E., SPARSITY).

Dimensionality	Zero ratio
100	77.6%
2000 (w/e avg)	0.0%
400 (w avg)	49.2%
400 (e avg)	52.0%

D. RQ: Is it more effective to increase embedding dimensionality or to utilize multiple events / time windows?

Our proposed method segments user behavioral history along two axes: time windows and event type (e.g., purchase, search) and generates corresponding embeddings for each segment. As a result, one key design decision is how to aggregate these segment-level embeddings into a single-user representation that satisfies the challenge constraint (dimensionality $\leq$ 2048).

Table VI compares different configurations of dimensionality per event and time window. The first row is the same result as in Table IV and the second row is the result of the 2000-dimensional embedding. In this setting, the 2000-dim user embeddings are generated by averaging all slices (7) from 40,000-dim full concatenation of all 5 events $\times$ 4 time windows $\times$ 2000-dim vectors. The third and fourth rows are the results of the 400-dim variants, where 8000-dim embeddings are averaged over the time window (5) and event(6) to generate 2000-dim and 1600-dim user embeddings, respectively. Interestingly, all configurations yield comparable performance in downstream tasks. This suggests that the embeddings derived from multiple time-event slices contain semantically similar information, which aligns with our design objective: to build universal user embeddings that robustly encode user traits from any behavioral segment, regardless of time or interaction type. We observe a slight advantage in event-wise averaging (Eq. (6)) compared to time window-wise averaging (Eq. (5)). This aligns with the intuition that behaviors grouped by event types are semantically more coherent than those grouped by time alone.

From a sparsity perspective, this becomes even more salient. As shown in Table VII, lower-dimensional representations (100- or 400-dim) exhibit a significant proportion of zeros, naturally arising from user inactivity in certain time windows or event types. These zero or null vectors translate into partial sparsity in the final user representation, which is advantageous for both interpretability and memory efficiency. In contrast, the

2000-dim representation averaged from a dense 40,000-dim vector loses this sparsity due to the averaging process, resulting in a dense vector with values in nearly all dimensionalities. Despite this, its performance does not outperform the more sparse 400-dim or even 100-dim variants. These findings highlight the practical benefits of compact, sparse embeddings: not only do they maintain competitive accuracy, but they are also more suitable for scalable deployment and downstream model compression.

V. Conclusion

We proposed a two-stage contrastive learning framework for universal user embeddings. By capturing user consistency across time (intra-event) and aligning intent across event types (cross-event), the method provides task-agnostic representations. Experiments on the large-scale RecSys Challenge 2025 dataset show clear improvements over baselines, particularly in propensity prediction tasks, and demonstrate strong zero-shot generalization to unseen tasks. The approach is also computationally efficient, making it suitable for large-scale and privacy-sensitive applications because it relies solely on behavioral logs without requiring explicit personal attributes such as age or gender. Looking ahead, extending the framework to incorporate new types of behavioral signals (e.g., multimedia consumption, cross-platform activities) and exploring its integration with privacy-preserving learning paradigms represent promising directions for further enhancing universality and practical deployments.

References

[1] Y. Li, K. Liu, R. Satapathy, S. Wang, and E. Cambria, "Recent developments in recommender systems: A survey [review article]," *IEEE Computational Intelligence Magazine*, vol. 19, no. 2, pp. 78–95, May 2024. [Online]. Available: https://doi.org/10.1109/MCI.2024.3363984

[2] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems*, ser. RecSys, 2016, pp. 191–198. [Online]. Available: https://doi.org/10.1145/2959100.2959190

[3] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," arXiv preprint arXiv:1511.06939, 2016. [Online]. Available: https://arxiv.org/abs/1511.06939

[4] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD, 2018, pp. 1059–1068. [Online]. Available: https://doi.org/10.1145/3219819.3219823

[5] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD, 2018, pp. 1930–1939. [Online]. Available: https://doi.org/10.1145/3219819.3220007

[6] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR, 2002, pp. 253–260. [Online]. Available: https://doi.org/10.1145/564376.564421

[7] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD, 2011, pp. 448–456. [Online]. Available: https://doi.org/10.1145/2020408.2020480

[8] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, ser. UAI, 2009, pp. 452–461.

[9] C. Wang, J. Zhu, A. Li, Z. Li, and Y. Wang, "Dual-channel representation consistent recommender for session-based new item recommendation," *Expert Systems with Applications*, vol. 249, p. 123681, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424005475

[10] M. Jangid and R. Kumar, "Mitigating cold start problem in recommendation systems via transfer learning approach," in *2023 International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC)*, 2023, pp. 1–6.

[11] J. Yao, Z. Dou, R. Xie, Y. Lu, Z. Wang, and J.-R. Wen, "USER: A unified information search and recommendation model based on integrated behavior sequence," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, ser. CIKM, 2021, pp. 2373–2382. [Online]. Available: https://doi.org/10.1145/3459637.3482489

[12] C. Li, Y. Xie, C. Yu, B. Hu, Z. Li, G. Shu, X. Qie, and D. Niu, "One for all, all for one: Learning and transferring user embeddings for cross-domain recommendation," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, ser. WSDM, 2023, pp. 366–374. [Online]. Available: https://doi.org/10.1145/3539597.3570379

[13] Y. Ni, D. Ou, S. Liu, X. Li, W. Ou, A. Zeng, and L. Si, "Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD, 2018, pp. 596–605. [Online]. Available: https://doi.org/10.1145/3219819.3219828

[14] J. Gu, F. Wang, Q. Sun, Z. Ye, X. Xu, J. Chen, and J. Zhang, "Exploiting behavioral consistence for universal user representation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 05 2021, pp. 4063–4071.

[15] X. Li, J. Sheng, J. Cao, W. Zhang, Q. Li, and T. Liu, "CDRNP: Cross-domain recommendation to cold-start users via neural process," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, ser. WSDM, 2024, pp. 378–386. [Online]. Available: https://doi.org/10.1145/3616855.3635794

[16] R. Seth and A. Sharaff, *A Comparative Overview of Hybrid Recommender Systems: Review, Challenges, and Prospects*. Wiley, 2022, pp. 57–98.

[17] J. Dabrowski, M. Janicka, L. Sienkiewicz, G. Stomfai, D. Jannach, M. Polignano, C. Pomo, A. Srivastava, and F. Barile, "Recsyschallenge2025," 2025. [Online]. Available: https://www.recsyschallenge.com/2025/

[18] Q. Tan, J. Zhang, J. Yao, N. Liu, J. Zhou, H. Yang, and X. Hu, "Sparse-interest network for sequential recommendation," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, ser. WSDM, 2021, pp. 598–606. [Online]. Available: https://doi.org/10.1145/3437963.3441811

[19] Y. Ji, A. Sun, J. Zhang, and C. Li, "A re-visit of the popularity baseline in recommender systems," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR, 2020, pp. 1749–1752. [Online]. Available: https://doi.org/10.1145/3397271.3401233

[20] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "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)*, 2014, pp. 1724–1734. [Online]. Available: https://aclanthology.org/D14-1179/

[21] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, ser. KDD, 2021, pp. 2114–2124. [Online]. Available: https://doi.org/10.1145/3447548.3467401

[22] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML, 2020.