

RESEARCH ARTICLE

A Multimodal Deep Learning Framework for Personalized Event Recommendation with Hyperparameter Optimization via Genetic Algorithm

Serap Gulgen^{a,*}

^a Department of Computer Engineering, Artificial Intelligence Programme, Amasya University, 05100 Amasya, Türkiye

* Corresponding author: 258131511@ogrenci.amasya.edu.tr

ARTICLE INFO

Article history

Received 25 June, 2026
Revised 25 June, 2026
Accepted 30 June, 2026
Available online 30 June, 2026

Keywords

Event recommendation
Multimodal fusion
Heterogeneous graph transformer
Gated multimodal unit
Bayesian personalized ranking
Hyperparameter optimization
Genetic algorithm

Abstract

Personalized event recommendation poses a fusion challenge: textual descriptions, social graph connectivity, and user interaction history each carry information that no single-modality model can fully exploit. This paper presents a multimodal deep learning framework that combines DistilBERT text embeddings, Heterogeneous Graph Transformer (HGT) graph embeddings, and interaction features through a Gated Multimodal Unit (GMU) trained with Bayesian Personalized Ranking (BPR) loss. A Genetic Algorithm optimizes six hyperparameters including learning rate, batch size, embedding dimensions, and graph encoder depth. An order-preserving, leakage-free strategy is proposed for handling out-of-vocabulary tokens: rather than mapping unknown tokens to a shared index, they are removed from the input sequence while surviving tokens retain their original positional order, preventing representational collapse at small vocabulary sizes. Experiments are conducted on a controlled synthetic dataset comprising 10,000 users, 3,000 events, and 80,000 interactions, with random and popularity baselines included as reference points. Results confirm correct gradient flow and optimizer behavior; the experiments reported here establish architectural and pipeline feasibility rather than superior recommendation performance, with the framework positioned as a proof of concept intended for subsequent validation on datasets with genuine preference structure.

1. Introduction

Event discovery in urban environments has become increasingly difficult to navigate. Most platforms surface events chronologically or by category, leaving users to manually filter options that may be poorly matched to their interests, social connections, or prior behavior. The signals that could support better matching are available: users read event pages for varying durations, attend selectively, and maintain social ties with people whose tastes overlap their own. Combining these signals within a single trainable model remains an open problem [1].

Prior work on event recommendation has addressed different parts of this problem in isolation. Content-based methods encode textual descriptions and retrieve similar items, but treat each user independently of their social context [2], [3]. Collaborative filtering leverages the interaction matrix but discards what events are actually about [4], [5]. Graph-based methods have demonstrated that modeling user-item-social relationships as a heterogeneous network captures structural signals unavailable to text or interaction history alone [6], [7], though most implementations rely on hand-crafted node features. Combining text semantics, graph structure, and interaction

signals within a jointly trainable fusion architecture has received limited systematic attention in the event recommendation setting [8].

Gating mechanisms have emerged as an effective approach to multimodal fusion when modality informativeness varies across samples. Static concatenation assigns fixed importance to each source regardless of context [9], [10], whereas a Gated Multimodal Unit (GMU) [12] learns per-sample weights, suppressing sources that carry little signal for a given user-event pair. The Heterogeneous Graph Transformer (HGT) [13] provides a graph encoder suited to this setting, as its type-dependent attention distinguishes between the different edge semantics present in a user-event-social graph. Combining frozen DistilBERT embeddings as node features with HGT graph propagation and GMU fusion is a direct but underexplored design.

Hyperparameter selection for deep recommendation models involves the interaction of embedding dimensions, graph encoder depth, attention head count, learning rate, and batch size. Genetic Algorithms offer a population-based search strategy that has shown advantages over both random search and gradient-free hill-climbing for neural architecture and hyperparameter optimization [14], [15]. Applied to graph learning [17] through the DEAP library [16], GA-based search is a practical option when the configuration space is too large for exhaustive evaluation.

The framework developed in this paper draws these threads into a single trainable pipeline. At its core is a three-branch multimodal design that fuses DistilBERT text embeddings, HGT graph embeddings, and interaction features through a GMU trained with BPR loss, while a Genetic Algorithm searches jointly over six hyperparameters spanning graph encoder configuration and embedding dimensions. The pipeline is complemented by an order-preserving, leakage-free OOV token filtering strategy, offered as an alternative to shared-index handling in vocabulary-limited sequence models.

1.1. Contributions

DistilBERT, HGT, GMU, BPR, and Genetic Algorithm-based search are each individually established techniques; the methodological contribution of this work is not any one of them but their specific integration into a single three-branch pipeline for the text-graph-interaction modality combination that event recommendation requires, a combination Section 1.3 notes has not been examined systematically in this setting, together with the novel filtering strategy in (iii) below. The contributions of this work are the following.

- (i) A three-branch multimodal framework for event recommendation that fuses frozen DistilBERT text embeddings, HGT graph embeddings, and interaction features through a Gated Multimodal Unit, with the fusion and scoring components trained end-to-end using Bayesian Personalized Ranking loss.
- (ii) A Genetic Algorithm that searches jointly over six hyperparameters spanning graph encoder configuration (HGT depth, attention heads), embedding dimensions (gmu_dim, int_dim), and optimization settings (learning rate, batch size).
- (iii) An order-preserving, leakage-free out-of-vocabulary token filtering strategy, proposed as an alternative to shared-index handling in vocabulary-limited sequence models.

1.2. Notation and organisation

Section 2 describes the dataset, architecture, and experimental protocol; Section 3 presents results and discussion; Section 4 concludes.

1.3. Related Work

Graph-based recommendation developed rapidly following PinSage [6], which demonstrated the scalability of graph convolutional networks to large-scale settings. Relational GCNs [7] and HGT [13] extended this to heterogeneous graphs, with HGT's type-dependent attention capturing the distinct roles of different edge types. In event-specific settings, graph neural networks have been applied to model social influence [18] and sequential attendance patterns [19]. The combination of HGT with pre-trained language model embeddings as node features has not been examined systematically in this context.

Pre-trained language models have become standard for text encoding in recommendation systems [2], [20]. DistilBERT [21] retains most of BERT's representational capacity at substantially lower parameter count, making

it practical for offline embedding precomputation. Freezing the language model and training only downstream components is a competitive strategy when interaction data is available [20], [22], and avoids fine-tuning instability in low-data regimes. Session-based recommendation research [24] has further established that simple sequential models trained on interaction histories can serve as strong baselines, motivating the inclusion of explicit behavioral interaction signals in the proposed framework.

For multimodal fusion, gating and attention mechanisms outperform concatenation when modality quality varies across samples [10], [11]. The GMU [12] learns soft weights over modality-specific projections, and its extension to a three-modality setting with social graph structure remains underexplored in event recommendation. GA-based hyperparameter search has been applied to convolutional architectures [14] and graph neural networks [23]; this work extends that line to a multimodal recommendation setting.

2. Method

2.1. Dataset

The experiments use a synthetic dataset designed to exercise the full heterogeneous graph structure and interaction signal diversity required by the proposed architecture. Publicly available collections such as Yelp, Foursquare, and Meetup each lack one or more of the following simultaneously: explicit waitlist interactions, friend-level attendance records, and rich free-text event descriptions at sufficient density. Because the study aims to validate individual architectural components under controlled conditions, a synthetic environment allows inter-component interactions to be examined without confounding from real-world data noise. Results are accordingly interpreted as evidence of pipeline correctness rather than deployment-ready recommendation performance.

The dataset contains 10,000 users, 3,000 events, 80,000 interactions across four action types (view, like, attend, waitlist), and 39,985 undirected social edges. Each user record carries age, city, interest tags, and an activity level label; each event carries category, city, price, and a free-text description. Interaction rates approximate platform-observed distributions (50% views, 25% likes, 15% attendances, 10% waitlist entries). The data is split 70/10/20 for training, validation, and testing, with random seed fixed at 42 throughout. Because attend actions were assigned without reference to user-event semantic similarity, the interaction matrix contains no learnable preference signal. This means NDCG@10 values reflect whether the pipeline is functioning correctly rather than whether the model has learned to recommend well, a distinction maintained throughout Section 3.

2.2. System Architecture

Each user-event pair is processed through three encoding branches whose outputs are fused by a GMU before a scalar scoring head. At inference, all 3,000 events are scored for each user and ranked; the top-K items form the recommendation list.

2.2.1. Text Encoding (*DistilBERT*)

Event descriptions and user profiles are encoded offline using the distilbert-base-uncased checkpoint [21]. The final hidden state of the [CLS] token is extracted as a 768-dimensional embedding. User profiles are represented as short descriptive sentences combining age, city, interest tags, and activity level. Embeddings are precomputed in batches of 32 at a maximum sequence length of 128 tokens and saved to disk. DistilBERT weights are frozen throughout; only the fusion and scoring components are updated during training.

For vocabulary-limited input sequences, an order-preserving, leakage-free OOV filtering strategy is applied. The vocabulary of size K is constructed from training-set token frequencies only; no frequency information from the validation or test sets is used. At inference, tokens absent from the vocabulary are removed from the input sequence entirely. Surviving tokens retain their original positional order before entering the Embedding layer: removing a token at position i does not shift subsequent tokens forward. This differs from substituting a shared OOV index, which maps all unknown tokens to the same representation and causes samples to converge toward each other in embedding space as vocabulary shrinks. Padding uses index 0, passed to the Embedding layer as `padding_idx = 0`.

2.2.2. Graph Encoding (HGT)

A heterogeneous graph is constructed with two node types (user, event) and four edge types: user-clicked-event, user-attended-event, user-friends-user, and the reverse edge event-rev_clicked-user. Node features concatenate the precomputed DistilBERT embedding with a single normalized scalar (age for users, price for events), yielding 769-dimensional inputs. In the present implementation the HGT encoder [13] performs a single forward pass with fixed, randomly initialized weights, so the graph branch supplies structural projections of the heterogeneous topology rather than end-to-end learned features. This is a deliberate scoping choice: it isolates the fusion and scoring components so that their behaviour can be validated independently, and jointly training the encoder is taken up as a direct extension in Section 3.4. The encoder maps each node type from 769 dimensions through a 128-dimensional hidden space to 64-dimensional output embeddings:

(1)

$$h_t^{(l)} = \sigma \left(\sum_{s \in N(t)} \text{Attention}(s, e, t) * \text{MSG}(s, e, t) \right)$$

Type-specific weight matrices are used for both attention and message computation, maintaining distinct treatment of clicked, attended, and social edges. With fixed weights the graph branch encodes the structural position of each node within the heterogeneous topology; learning these relations end-to-end is a natural extension, discussed in Section 3.4.

2.2.3. Interaction Feature Encoding

For each user-event pair in training, the interaction signal consists of normalized dwell time and a one-hot encoded action type across four categories (view, like, attend, waitlist). These five values pass through a two-layer MLP with ReLU activations, producing an `int_dim`-dimensional representation. At inference for event-side encoding, where no interaction record is available, dwell time and action are set to zero and event price serves as a single scalar input with the remaining dimensions zeroed, providing a compact proxy for event-side context. A fuller design with dedicated inference-time event encoding is outlined in Section 3.4.

2.2.4. Gated Multimodal Fusion (GMU)

The three branch outputs are each projected to a shared `gmu_dim`-dimensional space with tanh activation. A gating network concatenates the three raw embeddings and produces a three-way softmax weight vector. The fused representation is:

$$\mathbf{z} = g_1 * \tanh(W_t h_{text}) + g_2 * \tanh(W_g h_{graph}) + g_3 * \tanh(W_i h_{int}) \quad (2)$$

The softmax constraint keeps gate weights summing to one. Because gates are computed from the full concatenation of all three raw embeddings, the weighting decision reflects inter-modality context rather than the magnitude of any individual branch.

2.2.5. Scoring and BPR Loss

User and event GMU representations are concatenated and passed through a two-layer MLP producing a scalar relevance score. Training uses Bayesian Personalized Ranking loss [25]:

$$\mathcal{L}_{BPR} = - \sum_{(u,i,j) \in D} \log \sigma(y_{ui} - y_{uj}) \quad (3)$$

where i is an attended event, j is a randomly sampled unobserved event, and σ is the sigmoid function. BPR optimizes pairwise ranking directly, aligning the training objective with the top-K evaluation criterion.

2.3. Hyperparameter Optimization via Genetic Algorithm

The search space covers six dimensions: `gmu_dim` in {64, 128, 256}, `int_dim` in {16, 32, 64}, learning rate in {1e-4, 5e-4, 1e-3, 5e-3}, batch size in {128, 256, 512}, HGT layers in {1, 2, 3}, and HGT attention heads in {2, 4, 8}. Each individual is encoded as a six-integer chromosome, with each gene indexing into the corresponding choice

list. The GA uses tournament selection (size 3), two-point crossover (probability 0.6), and uniform integer mutation (0.3 per gene), implemented with DEAP [16]. Each candidate is evaluated by training for three epochs; fitness is the resulting validation NDCG@10. A demonstration run uses population size 6 over three generations; this budget is scoped as a proof-of-concept demonstration of the joint architecture-hyperparameter search rather than an exhaustive search of the 1,296-configuration space, an interpretation Sections 3.3 and 3.4 return to when reading the resulting configurations.

Algorithm 1 GA-based joint NAS and hyperparameter search for EventRecModel (Section 2.3)

Input: search space $S = \{\text{gmu_dim, int_dim, lr, batch_size, hgt_layers, hgt_heads}\}$; population size $P = 6$; generations $G = 3$; tournament size = 3; crossover probability $CXPB = 0.6$; mutation probability $MUTPB = 0.3$ (per gene); fitness-training epochs = 3; seed = 42.

- 1: Initialize population $\text{pop} \leftarrow P$ individuals, each a uniformly random integer chromosome indexing into S .
- 2: for each individual ind in pop do
- 3: $\text{fitness}(\text{ind}) \leftarrow \text{Evaluate}(\text{ind})$ [decode ind ; train EventRecModel for 3 epochs; fitness = validation NDCG@10]
- 4: end for
- 5: $\text{best} \leftarrow$ individual in pop with the highest fitness
- 6: for generation $g = 1, \dots, G$ do
- 7: $\text{offspring} \leftarrow \text{TournamentSelect}(\text{pop}, \text{tournsize} = 3)$, cloned
- 8: for each consecutive pair $(c1, c2)$ in offspring do
- 9: with probability $CXPB$, apply two-point crossover to $(c1, c2)$ and invalidate their fitness
- 10: end for
- 11: for each ind in offspring do
- 12: with probability $MUTPB$ per gene, apply uniform-integer mutation to ind and invalidate its fitness
- 13: end for
- 14: for each ind in offspring with invalid fitness do
- 15: $\text{fitness}(\text{ind}) \leftarrow \text{Evaluate}(\text{ind})$
- 16: end for
- 17: $\text{elites} \leftarrow$ top-2 individuals of pop
- 18: $\text{pop} \leftarrow$ top- P individuals of $(\text{offspring} \cup \text{elites})$ $\square$ elitism
- 19: if $\text{best-of}(\text{pop}).\text{fitness} > \text{best}.\text{fitness}$ then $\text{best} \leftarrow \text{best-of}(\text{pop})$
- 20: end for

Output: best configuration; final population pop with per-individual fitness (Table 3); per-generation log (best/mean fitness).

2.4. Evaluation Protocol

Ranking quality is assessed at $K = 10$ using NDCG@10, Recall@10, and Precision@10. NDCG@10 is computed as:

$$NDCG@K = \frac{DCG@K}{IDCG@K}, \quad IDCG@K = \sum_{i=1}^{\min(|GT|, K)} \frac{1}{\log_2(i+1)} \quad (4)$$

Setting the ideal relevance vector to length $\min(|GT|, K)$ ensures the denominator reflects the full ground-truth set rather than only the retrieved relevant items. For a user with three attended events of which one appears at rank 1 in the top-10 list, $IDCG = 1/\log_2(2) + 1/\log_2(3) + 1/\log_2(4) = 1 + 0.631 + 0.5 = 2.131$, giving $NDCG = 1/2.131 = 0.47$. Scores are computed against all 3,000 events per user and averaged over a 500-user sample at each validation step. Random and popularity baselines are included as reference points; the popularity baseline ranks events by total attend-action count in the training set and returns the same list for every user. Library versions: PyTorch 2.11.0,

torch-geometric 2.8.0, transformers 5.3.0, scikit-learn 1.8.0, DEAP 1.4.4, NumPy 2.4.2, Python 3.14.0. Experiments run on standard CPU infrastructure.

2.5. Experimental Setting

All reported numbers come from a single run with a fixed random seed (seed = 42), covering dataset generation (Section 2.1), model initialization, training, and evaluation; no averaging or repetition across multiple seeds was performed, and reported values are a single-run point estimate rather than a mean or median over repeated trials. Library versions are listed in Section 2.4. All experiments — embedding extraction, training, baseline computation, and the genetic algorithm search — were run in a CPU-only environment with no GPU acceleration, on a 12-core machine.

3. Results and Discussion

3.1. Baseline Comparison

Table 1 reports NDCG@10, Recall@10, and Precision@10 for the random baseline, popularity baseline, and the trained model checkpoint at epoch 36, which achieved the best validation NDCG@10 across the full training run.

Table 1. Baseline comparison (corrected NDCG@10).

Model	NDCG@10	Recall@10	Precision@10
Random Baseline	0.0015	0.0027	0.0004
Popularity Baseline	0.0072	0.0150	0.0016
Proposed Model (epoch 36)	0.0063	0.0100	0.0010

The proposed model surpasses the random baseline across all three metrics, confirming that the optimizer acquires ranking preferences beyond the noise floor. It falls short of the popularity baseline, which is the expected outcome given the dataset's construction: because attend actions carry no preference signal, the only exploitable structure in the interaction matrix is marginal event frequency, and a user-personalized model cannot outperform a frequency ranker when no user-specific signal exists. As Table 2 shows, validation NDCG@10 varies across epochs, so the epoch-36 value of 0.0063 is best read as the strongest checkpoint of the run rather than a stable plateau—variation that is itself consistent with the absence of a learnable preference signal in this dataset. Together, the results confirm that the training pipeline is functioning correctly and that the performance ceiling on this dataset is a data property rather than an architectural limitation. A further consequence of the frozen HGT encoder (Section 2.2.2) is that the graph branch contributes fixed structural projections rather than end-to-end learned relational features; the results above should accordingly be read as validating the fusion and scoring components given that fixed graph representation, not as a measure of what a jointly trained graph encoder would achieve.

3.2. Training Dynamics

Table 2 reports training loss and validation NDCG@10 at selected epochs across the forty-epoch training run.

Table 2. Training loss and validation NDCG@10 at selected epochs.

Epoch	BPR Training Loss	Validation NDCG@10
1	0.6930	0.0019
2	0.6882	0.0000
5	0.6158	0.0000
10	0.6074	0.0000

15	0.6122	0.0000
20	0.6114	0.0032
30	0.6120	0.0000
40	0.6118	0.0000

The initial BPR loss of 0.6930 aligns with the theoretical value of $-\log(0.5) = 0.693$ expected under random weight initialization, where positive and negative items receive equal scores. Loss falls sharply over the first five epochs, from 0.6930 to 0.6158, then settles into a noisy band roughly between 0.60 and 0.62 for the remaining thirty-five epochs, drifting only slightly lower by the end of training (minimum 0.6001 at epoch 27) rather than flattening into a hard plateau. The pattern is consistent with the optimizer quickly exploiting the one axis of separation the dataset supports — marginal event popularity, since attend actions carry no user-specific signal by construction (Section 2.1) — after which further epochs mostly perturb the loss within that band rather than reducing it further. Validation NDCG@10 does not track the loss curve: it sits at or near zero for most epochs and spikes intermittently (0.0032 at epoch 20, 0.0063 at epoch 36) with no sustained trend in either direction, the expected signature of a ranking metric evaluated on a 200-user sample against events with no recoverable per-user preference structure. Together, the two curves show the optimizer doing substantial, real work while confirming that the work is bounded by what the dataset makes learnable, exactly as Section 2.1 anticipates.

3.3. Hyperparameter Search Results

Table 3 reports all six individuals in the GA's final-generation population, with fitness values computed at three training epochs; rows are auto-labeled by trainable parameter count (Smaller / Larger Model) rather than by a fixed set of names, since the frozen HGT branch contributes no trainable weights and `gmu_dim/int_dim` are what actually set model capacity here (Section 2.2.2).

Table 3. GA configurations and three-epoch validation NDCG@10.

<code>gmu_dim</code>	<code>int_dim</code>	<code>lr</code>	<code>batch</code>	HGT layers	HGT heads	NDCG@10	Label
64	16	5e-3	512	3	8	0.0053	Smaller Model (GA Best)
64	16	5e-3	512	3	8	0.0053	Smaller Model (GA Best)
64	16	1e-4	512	3	8	0.0032	Smaller Model
64	16	1e-4	512	3	8	0.0032	Smaller Model
64	16	5e-3	128	3	8	0.0032	Smaller Model
64	64	5e-3	128	3	8	0.0032	Larger Model

The GA converged quickly: the best three-epoch NDCG@10 (0.0053) was first reached in generation 2 and did not improve in generation 3, and the winning individual survives into the final population twice through elitism — Table 3 reports the population as returned, duplicates included, rather than a curated subset. Five of the six final individuals share an identical trainable-parameter count (65,508), since `gmu_dim`=64 and `int_dim`=16 — not HGT depth or head count, which stayed fixed at three layers and eight heads across the whole final population — are what set trainable capacity while the HGT branch itself remains frozen (Section 2.2.2); the sixth individual differs only by a larger `int_dim` (64), raising its parameter count to 72,900. Four individuals also return bit-identical three-epoch fitness values despite differing in learning rate or batch size, indicating that the three-epoch proxy is too coarse to separate these configurations on this dataset. The best configuration pairs the highest learning rate in the search space (5e-3) with the deepest, widest HGT setting tested (three layers, eight heads), while keeping `gmu_dim`

and `int_dim` at their smallest values. At a population of six and three generations, the search visits only 24 of the 1,296 possible configurations — small enough that which few candidates happen to get sampled, and how they interact with the fixed training seed, can dominate the outcome as much as any genuine hyperparameter effect. Accordingly, this run's learning-rate and HGT-depth pattern is reported descriptively rather than as a settled sensitivity finding, and should be treated as provisional until replicated at a larger population and generation count, as Section 3.4 discusses.

3.4. Scope and Future Work

This study was scoped as an architectural proof of concept, and its results are best read at that level. The controlled synthetic dataset was chosen so that gradient flow, optimizer configuration, and pipeline correctness could be established in isolation, before the additional variability of real preference data is introduced. Recommendation-quality claims are accordingly reserved for that next stage, since the interaction matrix carries no preference signal by design (Section 2.1); the model's position relative to the popularity baseline follows directly from this choice and matches what the design anticipates. With the pipeline now validated end to end, the framework is well placed for the extensions set out below.

Several extensions follow naturally from the current design and define the next phase of the work. The most immediate is to train the HGT encoder jointly with the fusion and scoring components, so that the graph branch contributes learned relational features rather than the fixed structural projections used here; because the architecture already isolates this branch, the upgrade is a localized change rather than a redesign. A second refinement concerns the interaction branch, where event-side encoding at inference reuses event price as a compact stand-in for behavioural context. Routing the training-time and inference-time event representations through dedicated encoding paths would replace this proxy with a cleaner separation of the two information sources. The order-preserving OOV filtering strategy is similarly ready for a sharper test: its theoretical advantage—preserving positional order so that vocabulary-limited sequences avoid the representational collapse of shared-index mapping—does not surface at $K = 10$ over 3,000 candidates, where out-of-vocabulary events fall outside the top-10 list under any strategy. Evaluating it at larger cutoffs, or on corpora with higher OOV rates, is precisely the setting in which its benefit is expected to become measurable.

The genetic-algorithm search was run at a small population and generation count, in keeping with the proof-of-concept budget. At six individuals over three generations, the search visits only a small fraction of the 1,296 possible configurations (Section 3.3), which is on its own reason enough to treat any single run's learning-rate or HGT-depth pattern as a description of that run rather than a reliable hyperparameter-sensitivity signal. Replicating the search at a larger population and generation count, and on a dataset with genuine preference structure, is accordingly not an optional refinement but a precondition for treating any pattern the GA surfaces as design guidance rather than an artifact of which few configurations happened to be sampled. Closing the loop between the GA-selected configuration and the full training run belongs to that same next step: the forty-epoch run reported here used the default configuration, and carrying the GA's best configuration through to final training is where its contribution will be quantified.

4. Conclusion

This paper described a multimodal event recommendation framework that combines frozen DistilBERT text embeddings, HGT-based graph projections, and interaction features through a Gated Multimodal Unit trained with BPR loss. A Genetic Algorithm was applied to search over six hyperparameters including graph encoder depth and embedding dimensions. An order-preserving OOV token filtering strategy was proposed as an alternative to shared-index handling, removing unknown tokens from input sequences while preserving the positional order of surviving tokens.

Experiments on a controlled synthetic dataset confirmed that the training pipeline functions correctly: the initial BPR loss matches its theoretical expectation, loss falls sharply over the first several epochs before settling into a noisy low band, and the trained model surpasses the random baseline at $NDCG@10 = 0.0063$ at epoch 36. The model does not surpass the popularity baseline, attributable to the absence of learnable preference structure in the

synthetic interaction matrix rather than any architectural deficiency. The GA search produced measurable variation across configurations; at a population of six over three generations the search samples only a small fraction of the configuration space, so conclusions from a single such run should be treated as provisional rather than as settled hyperparameter guidance. More broadly, every number reported in this paper comes from a single run with a fixed random seed (Section 2.5): no averaging or repetition across seeds was performed, so results should be read as a point estimate of pipeline behavior rather than as evidence of robustness, and this limits what can be concluded about generalizability beyond the specific run reported here.

The most direct path forward is replication on a dataset where attend actions reflect genuine user preferences, repeated across multiple random seeds rather than the single run reported here, where the GMU gating behavior, OOV filtering effects, and GA-derived configuration patterns can be evaluated against meaningful baselines. Training the HGT encoder jointly with the rest of the model, rather than using fixed random projections, would also be a substantive architectural upgrade. The modular design, with independently replaceable text, graph, and interaction branches, is intended to accommodate these extensions without changes to the fusion or scoring components.

CRedit authorship contribution statement. Serap Gulgen: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. Funding. This research received no external funding.

Declaration of competing interest. The author declares no conflict of interest.

Data availability. The synthetic dataset generator, model implementation, and scripts required to reproduce every table in this paper are available at <https://github.com/SerapG/multimodal-event-recommendation>, under the MIT licence.

References

- [1] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in Proc. 26th International World Wide Web Conference, pp. 173-182, 2017, <https://doi.org/10.1145/3038912.3052569>.
- [2] Y. Hou, S. Mu, W. X. Zhao, Y. Li, B. Ding, and J.-R. Wen, "Towards universal sequence representation learning for recommender systems," in Proc. 28th ACM SIGKDD Conference, pp. 585-593, 2022, <https://doi.org/10.1145/3534678.3539381>.
- [3] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in Proc. ACM CIKM, pp. 1419-1428, 2017, <https://doi.org/10.1145/3132847.3132926>.
- [4] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in Proc. 42nd ACM SIGIR, pp. 165-174, 2019, <https://doi.org/10.1145/3331184.3331267>.
- [5] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," arXiv preprint arXiv:1706.02263, 2017, <https://arxiv.org/abs/1706.02263>.
- [6] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in Proc. 24th ACM SIGKDD, pp. 974-983, 2018, <https://doi.org/10.1145/3219819.3219890>.
- [7] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in Proc. 15th ESWC, pp. 593-607, 2018, https://doi.org/10.1007/978-3-319-93417-4_38.
- [8] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer," in Proc. 28th ACM CIKM, pp. 1441-1450, 2019, <https://doi.org/10.1145/3357384.3357895>.
- [9] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in Proc. IEEE ICDM, pp. 197-206, 2018, <https://doi.org/10.1109/ICDM.2018.00035>.
- [10] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, and T.-S. Chua, "MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video," in Proc. 27th ACM Multimedia, pp. 1437-1445, 2019, <https://doi.org/10.1145/3343031.3351034>.
- [11] Q. Liu, J. Hu, Y. Xiao, X. Zhao, J. Gao, W. Wang, Q. Li, and J. Tang, "Multimodal recommender systems: A survey," ACM Computing Surveys, 2024, <https://arxiv.org/abs/2302.03883>.
- [12] J. Arevalo, T. Solorio, M. Montes-y-Gomez, and F. A. Gonzalez, "Gated multimodal units for information fusion," in Proc. ICLR Workshop on Representation Learning, 2017, <https://arxiv.org/abs/1702.01992>.
- [13] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in Proc. The Web Conference 2020, pp. 2704-2710, 2020, <https://doi.org/10.1145/3366423.3380027>.
- [14] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," Nature Machine Intelligence, vol. 1, no. 1, pp. 24-35, 2019, <https://doi.org/10.1038/s42256-018-0006-z>.
- [15] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in Proc. 33rd AAAI, pp. 4780-4789, 2019, <https://doi.org/10.1609/aaai.v33i01.33014780>.

- [16] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagne, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171-2175, 2012, <http://jmlr.org/papers/v13/fortin12a.html>.
- [17] K. Zhou, X. Huang, Q. Song, R. Chen, and X. Hu, "Auto-GNN: Neural architecture search of graph neural networks," *Frontiers in Big Data*, vol. 5, 2022, <https://doi.org/10.3389/fdata.2022.1029307>.
- [18] J. Zhang, X. Shi, S. Zhao, and I. King, "STAR-GCN: Stacked and reconstructed graph convolutional networks for recommender systems," in *Proc. 28th IJCAI*, pp. 4264-4270, 2019, <https://doi.org/10.24963/ijcai.2019/592>.
- [19] Y. Liu, L. Chen, X. He, J. Peng, Z. Zheng, and J. Tang, "Modelling high-order social relations for item recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 9, pp. 4385-4397, 2022, <https://doi.org/10.1109/TKDE.2020.3039463>.
- [20] Y. Hou, B. Hu, Z. Zhang, and W. X. Zhao, "CORE: Simple and effective session-based recommendation within consistent representation space," in *Proc. 45th ACM SIGIR*, pp. 1796-1801, 2022, <https://doi.org/10.1145/3477495.3531955>.
- [21] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019, <https://arxiv.org/abs/1910.01108>.
- [22] W. Hua, Y. Ge, S. Xu, J. Ji, Z. Li, and Y. Zhang, "UP5: Unbiased foundation model for fairness-aware recommendation," in *Proc. 18th EACL*, pp. 1899-1912, 2024, <https://doi.org/10.18653/v1/2024.eacl-long.114>.
- [23] J. Xu et al., "AutoHEnsGNN: Winning solution to AutoGraph challenge for KDD Cup 2020," in *Proc. 38th IEEE ICDE*, pp. 352-366, 2022, <https://doi.org/10.1109/ICDE53745.2022.00031>.
- [24] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proc. ICLR*, 2016, <https://arxiv.org/abs/1511.06939>.
- [25] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th UAI*, pp. 452-461, 2009, <https://arxiv.org/abs/1205.2618>.