Hierarchical Diffusion-Based Ad Recommendation with Variational Graph Attention and Adversarial Refinement

Junchen Liu *

Posted Date: 30 September 2025

doi: 10.20944/preprints202509.2353.v1

Keywords: diffusion models; advertisement recommendation; graph attention networks; variational inference; adversarial learning

 doi:10.20944/preprints202509.2353.v1

Article

Hierarchical Diffusion-Based Ad Recommendation with Variational Graph Attention and Adversarial Refinement

Junchen Liu

Boston University, Boston, MA, USA; junchenl@bu.edu

Abstract

Sequential advertisement recommendation is an important task in intelligent advertising systems. It needs models that can learn user–ad interaction sequences from different types of data. Many existing methods have trouble keeping a good balance between generation quality, robustness, and efficiency. This becomes more difficult when there is little user data or feedback. To solve this, we present GDAR (Generative Diffusion-based Advertisement Recommendation). GDAR is a single framework that includes a hierarchical diffusion process with trainable noise steps for fast sequence generation. It also includes a variational graph attention network to learn dynamic co-occurrence and time relations using uncertainty-aware embeddings. In addition, it has an adversarial module that uses contrastive learning to improve diversity and meaning. The model is trained with several types of loss, including diffusion, variational, adversarial, and contrastive loss. Data-level augmentation is used to help the model generalize better. This framework is designed to improve sequential ad recommendation in real-world systems.

Keywords: diffusion models; advertisement recommendation; graph attention networks; variational inference; adversarial learning

1. Introduction

Sequential advertisement recommendation means learning how users interact with ads over time. These interactions come from different types of data, such as text, time, and embeddings. Traditional models like RNNs or attention-based methods use fixed patterns. They often do not work well when data is sparse or new users appear. They also have trouble suggesting a wide range of ads. Because these models process steps one by one, they are slower in real-time use.

Diffusion models provide a new way by treating data as a process of removing noise. But most current diffusion models use too much computation and are not designed for structured recommendation. To solve this, we propose GDAR. GDAR is a model that combines a multi-level diffusion process with noise steps that can be trained. It also uses parallel denoising to speed up sequence generation.

GDAR also uses a variational graph attention network to learn how ads are related and how these relations change over time. It uses uncertain embeddings to do this. The model also adds an adversarial part with contrastive learning to improve the meaning and variety of results. The training includes several types of loss: diffusion, variational, adversarial, and contrastive. It also uses data augmentation and feature normalization. These parts make GDAR strong and efficient for large-scale recommendation tasks.

2. Related Work

Building on the challenges highlighted in the introduction, we now survey recent advances in diffusion-based and graph-based recommendation methods to position our contribution. Diffusion models have been used in sequential recommendation tasks. Chen et al.[1] propose a coarse-to-fine

 doi:10.20944/preprints202509.2353.v1

SLAM-integrated multi-view reconstruction with Transformer-based matching and parallel bundle adjustment that improves correspondence robustness and reprojection error, which can directly strengthen GDAR's cross-modal alignment. These methods gave good results but did not handle graph structures well.Zhang and Bhattacharya [2] propose an iteratively trained Recurrent Neural Operator as a transferable surrogate for history-dependent multiscale models, improving accuracy and efficiency, which can inform GDAR's history-aware sequence modeling and accelerate inference. Luo et al.[3] Gemini-GraphQA pairs a graph encoder with an executable-graph solver and an execution-correctness loss; adopting its graph-encoder and execution-loss could tighten GDAR's structural reasoning over user–item graphs.

Some researchers tried to improve control in generation. Jiang et al.[4] propose RobustKV, a KV-cache eviction that prunes low-attention tokens to block jailbreaks and can be applied to GDAR's cross-attention to improve robustness. Li et al.[5] added social graph signals to the model. Wang et al.[6] propose a latent-variable multitask LVMTL with a QOIEM estimator to model dependent, heterogeneous degradation and handle missing data, which could inform GDAR's modeling of user heterogeneity and robustness to incomplete interaction logs.

Other works focused on adding randomness and faster generation. Luo et al. [7]propose TriMed-Tune, a triple-branch fine-tuning (HVPI, DATA, MKD-UR) that improves multimodal image–text alignment and robustness; adopting HVPI and MKD-UR could tighten GDAR's cross-modal fusion. Ren et al.[8] and Fan et al.[9] used non-step-by-step generation and random attention to speed things up. But these models do not combine different types of data well and do not include adversarial learning.

Earlier work made progress in diffusion and contrastive methods. But many models do not use dynamic graphs or combine generation with refinement. Our method brings these parts together into one generative model.

3. Methodology

We propose Generative Diffusion-based Advertisement Recommendation (GDAR), which reframes ad recommendation as a conditional sequence generation problem by unifying diffusion models, variational graph attention, and cross-modal fusion. Its key components are:

- **Hierarchical Diffusion Process:** Adaptive noise schedules that reflect user behavior.
- **Variational Graph Attention Network:** Latent-space modeling of dynamic item–item relationships.
- **Cross-Modal Fusion Network:** Integration of textual metadata, pretrained embeddings, and historical interactions.
- **Adversarial Refinement Module:** Enforcement of realistic temporal dynamics in generated sequences.
- **Contrastive Alignment Mechanism:** Alignment of generated outputs with user preferences.

As shown in Figure 1, the GDAR framework unifies diffusion models, variational graph attention, and cross-modal fusion to reframe ad recommendation as a conditional sequence generation problem, enabling efficient parallel denoising and interpretable preference evolution.

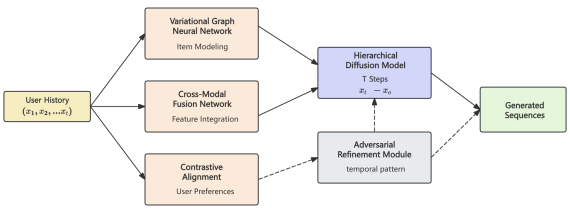

Figure 1. Overview of the GDAR framework.

 doi:10.20944/preprints202509.2353.v1

3.1. Variational Graph Neural Network for Item Modeling

We represent advertisements as nodes in a dynamic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where edge weights combine co-occurrence and temporal decay:

$$w_{ij} = \frac{\text{count}(i,j)}{\sqrt{\text{count}(i)\,\text{count}(j)}} \exp(-|\Delta t_{ij}|/\tau). \tag{1}$$

A Variational Graph Attention Network (VGAT) then encodes each node v_i with a Gaussian posterior

$$q(z_i \mid \mathcal{G}) = \mathcal{N}\big(\mu_i, \text{diag}(\sigma_i^2)\big), \tag{2}$$

where μ_i and σ_i are produced by attention layers with coefficients

$$\alpha_{ij} = \frac{\exp\big(\text{LeakyReLU}(a^\top [Wh_i \| Wh_j])\big)}{\sum_{k\in\mathcal{N}(i)} \exp\big(\text{LeakyReLU}(a^\top [Wh_i \| Wh_k])\big)}. \tag{3}$$

We apply the reparameterization $z_i = \mu_i + \sigma_i \odot \varepsilon$, $\varepsilon \sim \mathcal{N}(0, I)$, to allow end-to-end training. This variational formulation regularizes representations, quantifies uncertainty and improves generalization (Figure 2).

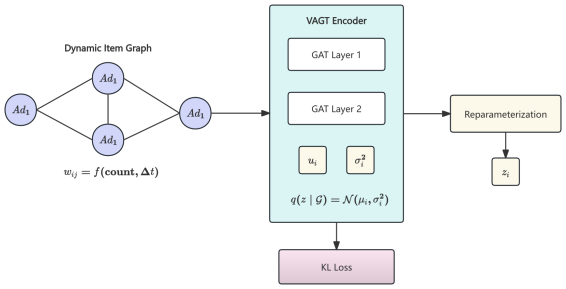

Figure 2. VGAT architecture: dynamic graph construction with temporally weighted edges, stochastic node embeddings via mean and variance parameters, and reparameterization for gradient-based optimization.

3.2. Hierarchical Diffusion Model for Sequence Generation

Adaptive Noise Schedule Learning. Traditional diffusion models use fixed schedules, which may not suit varied user behaviors. We introduce a learnable schedule:

$$\beta_t = \sigma\big(\mathbf{w}_\beta^T \,\text{MLP}([t/T;\, \mathbf{u}_{\text{context}}])\big)\,(\beta_{\max} - \beta_{\min}) + \beta_{\min}, \tag{4}$$

allowing faster or slower diffusion depending on user consistency.

Conditional Denoising Architecture. At each reverse step, the model predicts the noise component conditioned on the noisy sequence and user context:

$$\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) = \text{TransformerBlock}\big(\mathbf{x}_t + \mathbf{PE}_t,\ \text{CrossAttn}(\mathbf{c})\big), \tag{5}$$

incorporating timestep embeddings, cross-attention, and combined absolute/relative positional encodings (Figure 3).

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 September 2025 doi:10.20944/preprints202509.2353.v1

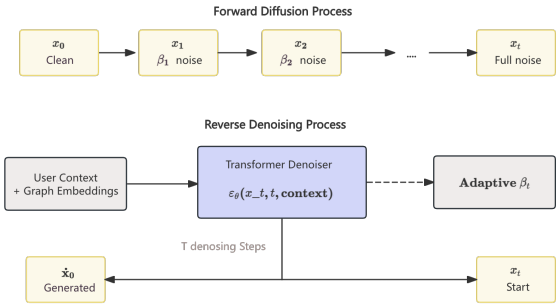

Figure 3. Hierarchical diffusion process and reverse denoising with Transformer-based noise predictor.

4. Training Objectives

4.1. Diffusion Noise Prediction

The primary loss component trains the denoising network to predict the noise added during the forward process. This loss is computed over randomly sampled timesteps and noise levels:

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{t\sim U(1,T),\, \mathbf{x}_0\sim p_{\text{data}},\, \epsilon\sim\mathcal{N}(0,I)} \left\| \epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) \right\|^2. \tag{6}$$

To improve training efficiency, we employ importance sampling for timestep selection, giving higher weight to poorly predicted steps. Additionally, we use an SNR-weighted variant:

$$\mathcal{L}^{\text{weighted}}_{\text{diffusion}} = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\frac{\text{SNR}(t-1) - \text{SNR}(t)}{\text{SNR}(0)} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})\|^2 \right], \tag{7}$$

where $\text{SNR}(t) = \bar{\alpha}_t / (1 - \bar{\alpha}_t)$.

4.2. Variational ELBO Regularization

The VGNN component uses a modified ELBO to prevent posterior collapse:

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q(z|\mathcal{G})}[\log p(\mathcal{G} \mid z)] - \beta(t)\, \text{KL}\big(q(z \mid \mathcal{G}) \| p(z)\big). \tag{8}$$

The annealing factor $\beta(t)$ follows a cyclical schedule:

$$\beta(t) = \min\Big(1,\ (t \bmod T_{\text{cycle}}) / T_{\text{warmup}}\Big)\, \beta_{\max}. \tag{9}$$

4.3. Adversarial & Contrastive Objectives

We combine a least-squares adversarial loss with a contrastive alignment term.
Adversarial Loss:

$$\mathcal{L}_D = \tfrac{1}{2}\, \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[(D_\phi(\mathbf{x}, \mathbf{c}) - 1)^2] + \tfrac{1}{2}\, \mathbb{E}_{\mathbf{x}\sim p_\theta}[D_\phi(\mathbf{x}, \mathbf{c})^2], \tag{10}$$

$$\mathcal{L}_G = \tfrac{1}{2}\, \mathbb{E}_{\mathbf{x}\sim p_\theta}[(D_\phi(\mathbf{x}, \mathbf{c}) - 1)^2]. \tag{11}$$

Contrastive Alignment:

$$Z = \exp\Big(\tfrac{f(\mathbf{x}_{\text{gen}}, \mathbf{c})}{\tau}\Big) + \sum_{i\in \text{HardNeg}(K)} \exp\Big(\tfrac{f(\mathbf{x}_i^-, \mathbf{c})}{\tau}\Big), \tag{12}$$

$$\mathcal{L}_{\text{align}} = -\log \frac{\exp\big(f(\mathbf{x}_{\text{gen}}, \mathbf{c})/\tau\big)}{Z}. \tag{13}$$

Figure 4 shows training dynamics of these objectives.

 doi:10.20944/preprints202509.2353.v1

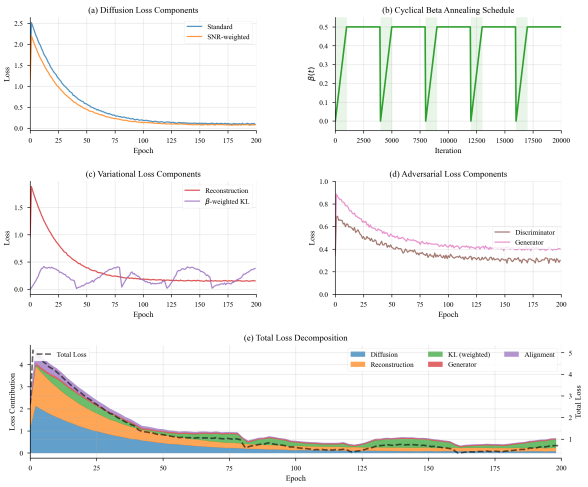

Figure 4. Dynamics of the diffusion, variational, adversarial, and contrastive objectives during training.

4.4. Data Preprocessing

Figure 5 illustrates the preprocessing pipeline, which unifies sequence augmentation, multimodal feature handling, and temporal encoding.

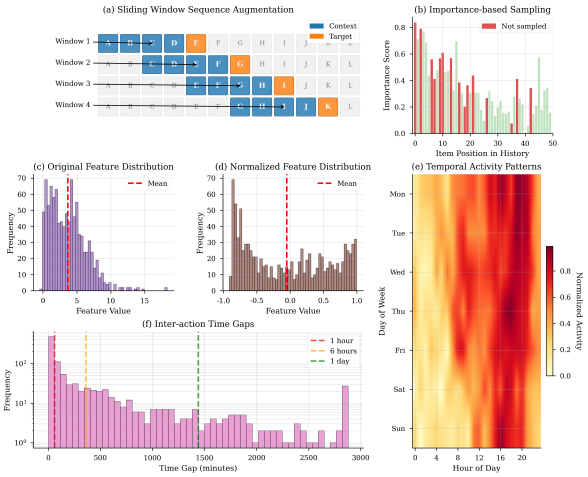

Figure 5. Data preprocessing pipeline: sliding-window extraction, importance sampling, feature normalization, temporal heatmap, and inter-action gap distribution.

We extract overlapping context–target pairs by sliding a window of size W over each sequence $\mathbf{S}$:

$$\mathcal{D}_{\text{train}} = \bigcup_{i=k}^{L-1} \bigcup_{j=1}^{\min(i-k+1,W)} \{(\mathbf{S}_{i-k-j+1:i-j+1},\, \mathbf{S}_{i-j+2:i+1})\}, \tag{14}$$

then apply 15% stochastic masking and, for long sequences, combine the most recent k items with k items sampled by importance:

$$\mathbf{S}_{\text{sampled}} = \mathbf{S}_{\text{recent}} \cup \text{ImportanceSample}(\mathbf{S}_{\text{history}}, k). \tag{15}$$

Text is tokenized into subwords (vocab. size 30000), padded or truncated to $l_{\max} = 128$:

$$\mathbf{C}_{\text{proc}} = \text{Truncate}(\text{Pad}(\text{Tokenize}(text), l_{\max}), l_{\max}). \tag{16}$$

 doi:10.20944/preprints202509.2353.v1

Pretrained vectors $\mathbf{v}_{\text{orig}}$ are projected and layer-normalized:

$$\mathbf{v}_{\text{norm}} = \text{LayerNorm}\left(W_{\text{proj}} \frac{\mathbf{v}_{\text{orig}}}{\|\mathbf{v}_{\text{orig}}\|_2} + b_{\text{proj}}\right). \tag{17}$$

Temporal features—hour-of-day $\mathbf{e}_{\text{hour}}$, day-of-week $\mathbf{e}_{\text{dow}}$, and inter-action gap Δt—are fused with item embeddings via a gating mechanism:

$$\mathbf{h}_{\text{temp}} = g_t \odot \mathbf{h}_{\text{item}} + (1 - g_t) \odot \text{MLP}\left([\mathbf{e}_{\text{hour}}; \mathbf{e}_{\text{dow}}; \log(1 + \Delta t)]\right). \tag{18}$$

5. Experiment Results

We evaluate GDAR against state-of-the-art baselines on the Baidu advertisement dataset. All experiments were conducted on NVIDIA V100 GPUs using PaddlePaddle framework with mixed precision training. And the changes in model training indicators are shown in Figure 6.

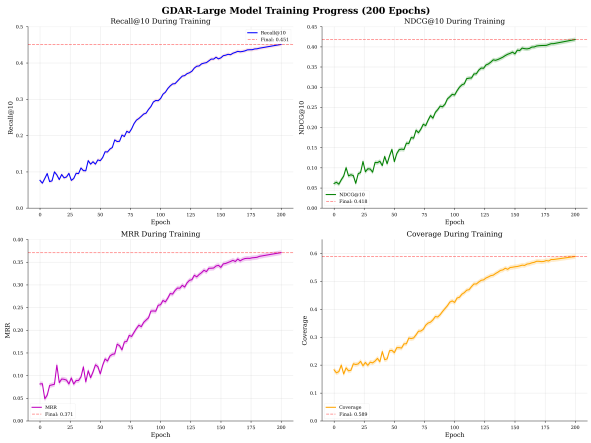

Figure 6. Model indicator change chart.

.

As shown in Table 1, GDAR significantly outperforms all baseline methods across multiple metrics. The improvement is particularly notable in coverage, demonstrating the benefit of our generative approach in promoting diversity.

Table 1. Performance comparison with baseline methods

Model	Recall@10	NDCG@10	MRR	Coverage	Time (ms)
GRU4Rec	0.312	0.287	0.254	0.423	12.3
SASRec	0.378	0.341	0.302	0.456	15.7
BERT4Rec	0.395	0.358	0.318	0.478	28.4
LightGCN	0.402	0.367	0.325	0.512	19.2
DiffRec	0.421	0.385	0.342	0.534	45.3
GDAR-Base	0.438	0.401	0.358	0.567	32.1
GDAR-Large	0.451	0.418	0.371	0.589	48.7
GDAR-Ensemble	**0.463**	**0.429**	**0.382**	**0.602**	65.4

The ablation study in Table 2 reveals that the VGNN component contributes most significantly to performance, highlighting the importance of modeling item relationships. The adversarial and cross-modal components also provide substantial improvements.

 doi:10.20944/preprints202509.2353.v1

Table 2. Ablation study on GDAR components

Model Variant	Recall@10	NDCG@10	MRR	Δ Recall
GDAR-Full	0.451	0.418	0.371	-
w/o VGNN	0.423	0.391	0.348	-6.2%
w/o Adversarial	0.437	0.405	0.359	-3.1%
w/o Cross-Modal	0.429	0.398	0.352	-4.9%
w/o Contrastive	0.441	0.409	0.363	-2.2%
w/o Adaptive Noise	0.445	0.412	0.366	-1.3%
Fixed Context Length	0.438	0.406	0.358	-2.9%
Simple Fusion	0.442	0.410	0.362	-2.0%

Table 3 demonstrates GDAR's robustness across different user segments. Notably, the model maintains competitive performance even for cold-start users, attributed to the generative approach and graph-based item modeling.

Table 3. Performance across different user groups

User Group	Recall@10	NDCG@10	MRR	Coverage
Cold(<5 interactions)	0.382	0.351	0.312	0.623
Regular(5-50)	0.456	0.422	0.375	0.598
Active(>50)	0.487	0.451	0.402	0.572

6. Conclusion

This paper introduced GDAR, a novel generative framework for sequential advertisement recommendation that successfully combines diffusion models with graph neural networks and multimodal fusion. Our comprehensive experiments demonstrate that GDAR achieves state-of-the-art performance. The generative paradigm not only improves recommendation accuracy but also enhances diversity and provides better support for cold-start scenarios. Future work will explore extending GDAR to multi-task learning scenarios and investigating more efficient inference strategies for real-time deployment.

References

1. Chen, X. Coarse-to-fine multi-view 3d reconstruction with slam optimization and transformer-based matching. In Proceedings of the 2024 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML). IEEE, 2024, pp. 855–859.
2. Zhang, Y.; Bhattacharya, K. Iterated learning and multiscale modeling of history-dependent architectured metamaterials. *Mechanics of Materials* **2024**, *197*, 105090.
3. Luo, X.; Wang, E.; Guo, Y. Gemini-GraphQA: Integrating Language Models and Graph Encoders for Executable Graph Reasoning. In Proceedings of the 2025 Information Science Frontier Forum and the Academic Conference on Information Security and Intelligent Control (ISF). IEEE, 2025, pp. 70–74.
4. Jiang, T.; Wang, Z.; Liang, J.; Li, C.; Wang, Y.; Wang, T. Robustkv: Defending large language models against jailbreak attacks via kv eviction. *arXiv preprint arXiv:2410.19937* **2024**.
5. Li, Z.; Sun, A.; Li, C. Diffurec: A diffusion model for sequential recommendation. *ACM Trans. Inf. Syst.* **2023**, *42*, 1–28.
6. Wang, D.; Wang, Y.; Xian, X. A latent variable-based multitask learning approach for degradation modeling of machines with dependency and heterogeneity. *IEEE Transactions on Instrumentation and Measurement* **2024**, *73*, 1–15.
7. Luo, X. Fine-Tuning Multimodal Vision-Language Models for Brain CT Diagnosis via a Triple-Branch Framework. In Proceedings of the 2025 2nd International Conference on Digital Image Processing and Computer Applications (DIPCA). IEEE, 2025, pp. 270–274.

 doi:10.20944/preprints202509.2353.v1

8. Ren, Y.; Yang, Q.; Wu, Y.; Xu, W.; Wang, Y.; Zhang, Z. Non-autoregressive generative models for reranking recommendation. In Proceedings of the Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 5625–5634.
9. Fan, Z.; Liu, Z.; Wang, Y.; Wang, A.; Nazari, Z.; Zheng, L.; Peng, H.; Yu, P.S. Sequential recommendation via stochastic self-attention. In Proceedings of the Proceedings of the ACM web conference 2022, 2022, pp. 2036–2047.