

Sistem Rekomendasi Film Berbasis Konteks Menggunakan Graph Neural Network Dengan Contrastive Learning Pada Dataset MovieLens

Rey Muhamad Rifqi¹, Alva Hendi Muhammad²

^{1,2} Pascasarjana Teknik Informatika – Universitas Amikom Yogyakarta

Jl. Ring Road Utara, Ngringin, Condongcatur - Kec. Depok - Kabupaten Sleman - Daerah Istimewa Yogyakarta

reymuh72@gmail.com, alva@amikom.ac.id

Abstrak— Sistem rekomendasi berbasis *collaborative filtering* saat ini menghadapi permasalahan fundamental berupa *data sparsity* dan *over-smoothing* yang menghambat identifikasi hubungan struktural kompleks pada graf interaksi *user-item*. Untuk menghadapi kendala tersebut, penelitian ini secara spesifik bertujuan untuk merancang arsitektur sistem rekomendasi yang mengintegrasikan *Graph Neural Network* (GNN) berbasis LightGCN, mekanisme *Contrastive Learning*, serta konteks temporal untuk memitigasi anomali *overfitting* dan meningkatkan akurasi. Pada arsitektur usulan, LightGCN berfungsi untuk mempelajari representasi *node* melalui propagasi *embedding* secara efisien, sementara *Contrastive Learning* diterapkan sebagai regulasi tambahan melalui mekanisme *self-supervised learning* untuk menghasilkan representasi fitur laten yang lebih stabil saat menghadapi *data sparsity*. Konteks temporal mencakup tahun, bulan, hari, jam, musim, dan status akhir pekan dilibatkan secara dinamis tanpa merusak struktur graf utama. Eksperimen dilakukan menggunakan dataset publik MovieLens 1M dengan skenario evaluasi *offline*. Hasil pengujian membuktikan bahwa model usulan secara efektif mampu menahan laju konvergensi prematur atau *overfitting* yang dialami oleh model *baseline*. Dari segi performa, kombinasi tiga komponen dalam arsitektur ini mencatat *HitRatio* sebesar 70.74%, yang menandakan peningkatan sebesar 4.62% dibandingkan dengan model *baseline* murni. Selain itu, model usulan juga menghasilkan metrik evaluasi NDCG sebesar 0.1888, MRR sebesar 0.3383, *Recall@20* sebesar 0.0628, dan MAP sebesar 0.0959.

Kata Kunci— Sistem Rekomendasi, LightGCN, *Contrastive Learning*, Konteks Temporal.

Abstract— Current collaborative filtering-based recommendation systems face fundamental challenges such as *data sparsity* and *over-smoothing*, which hinder the identification of complex structural relationships in *user-item* interaction graphs. In response to these constraints, this study specifically aims to design a recommendation system model architecture that integrates a LightGCN-based Graph Neural Network (GNN) with a Contrastive Learning (CL) mechanism and the injection of temporal context to mitigate *overfitting* anomalies and improve accuracy. In the proposed architecture, LightGCN is utilized to learn node representations efficiently through embedding propagation, while CL is applied as an additional regularization via self-supervised learning to generate more stable latent feature representations when dealing with sparse data spaces. Furthermore, temporal context features—including year, month, day, hour, season, and weekend status—are incorporated dynamically without disrupting the primary graph structure.

Experiments were conducted using the public MovieLens 1M dataset under an offline evaluation scenario. The test results prove that the proposed model effectively withstands the premature convergence or *overfitting* experienced by the baseline model. In terms of performance, this integrated architecture successfully achieved a Hit Ratio of 70.74%, marking a 4.62% improvement compared to the pure baseline model. Additionally, the model generated competitive evaluation metrics, yielding an NDCG of 0.1888, an MRR of 0.3383, a *Recall@20* of 0.0628, and a MAP of 0.0959. In conclusion, the integration of these three components empirically creates an architecture that is robust against data bias and precise in modeling the dynamics of user preferences.

Keywords— Recommendation System, LightGCN, *Contrastive Learning*, Temporal Context.

I. PENDAHULUAN

A. Latar Belakang Masalah

Seiring dengan volume data interaksi digital pengguna yang meningkat, sistem rekomendasi menghadapi masalah fundamental seperti *data sparsity* [1] dan *over-smoothing* [2]. Pendekatan seperti *Collaborative Filtering* (CF) telah lama menjadi standar, namun teknik ini seringkali gagal mengidentifikasi pola hubungan struktural yang kompleks pada graf interaksi *user-item* [3]. Untuk mengatasi kendala tersebut serangkaian penelitian terhadap *deep learning* menggunakan *neural network* dilakukan untuk meningkatkan akurasi prediksi *rating* dibandingkan algoritma konvensional.

Arsitektur *Graph Neural Network* (GNN) khususnya LightGCN muncul sebagai solusi yang efektif untuk menangkap interaksi spesifik *user-item* dengan lebih adaptif. LightGCN lebih unggul dibandingkan metode tradisional dengan menyederhanakan mekanisme agregasi pesan tanpa membebani model melalui transformasi non-linear yang kompleks [4]. Penelitian yang dilakukan oleh Nesmaoui et al. [5], menyatakan LightGCN terbukti lebih cepat dan akurat untuk merepresentasikan graf secara kontinu pada dataset MovieLens 1M meskipun matriksnya tidak lengkap. Namun penelitian tersebut juga menyoroti bahwa manajemen redundansi informasi pada graf dengan skala besar tetap menjadi hambatan. Untuk variasi permasalahan lain seperti *cold-start*,

diusulkan model modifikasi LightGCN bernama LightGCN++ oleh Chanda [6] yang merekayasa agregasi *embedding* di *layer* terakhir, namun studi tersebut juga tetap merekomendasikan perlunya struktur konvolusi berbasis permutasi agar representasi berjalan normal.

Selain penguatan arsitektur secara struktural, melalui *Context-Aware Recommender System* (CARS) menjadi solusi karena memanfaatkan informasi tambahan tentang situasi pengguna seperti lokasi, waktu, perangkat, dan suasana hati untuk memberikan rekomendasi yang lebih akurat dan relevan dibanding metode tradisional serta lebih efektif untuk menangani masalah *data-sparsity* [7], [8], [9]. Konteks spesifik seperti waktu memiliki pengaruh secara signifikan terhadap pergeseran preferensi pengguna terhadap film, misalnya genre tertentu disukai pada masa atau musim tertentu. Lebih lanjut, kemajuan teknologi *mobile* dan sensor *smartphone* memudahkan untuk mengumpulkan data konteks [10].

Konteks yang digabungkan ini memiliki potensi yang besar, namun memiliki tantangan arsitektural tersendiri. Abdomaleki [11] mencoba dengan menggunakan algoritma genetika untuk pembobotan konteks genre dan kru, belum berhasil untuk mengungkap sinyal kolaboratif antar pengguna. Penelitian yang dilakukan oleh Gao & Ma [12] menggabungkan GNN dan CARS menemukan interaksi fitur yang tidak terstruktur pada *layer* dalam sangat membatasi fleksibilitas model untuk merekomendasikan interaksi spesifik. Arsitektur *ContextGNN* dari Yuan [13] mencoba mengatasi bias model global dengan memisahkan representasi lokal dan jarak jauh, namun sifat *pair-agnostic* pada komponen global beresiko melewatkan hubungan unik antar *node*. Bahkan saat Pham & Phan [14] menyuntikan fitur demografi pada dataset MovieLens 1M secara langsung pada inialisasi LightGCN, hal tersebut justru memunculkan *noise* yang mengacaukan performa rekomendasi, hal ini membuktikan bahwa metode agregasi sederhana tidak cukup untuk mengintegrasikan konteks statis dengan perilaku *user* yang dinamis.

Penambahan konteks apapun termasuk konteks temporal ke dalam ruang fitur berisiko tinggi untuk memicu *noise* dimensionalitas dan merusak struktur graf. Oleh karena itu, komponen *Contrastive Learning* (CL) diintegrasikan sebagai mekanisme regularisasi tambahan. CL mendorong model untuk membedakan secara efektif antara pasangan positif dan negatif pada proses pembelajaran (*learning / training*) [15]. Secara langsung, komponen ini memitigasi permasalahan *overfitting* saat model menghadapi *data-sparsity*, sekaligus mencegah kolapsnya kualitas *embedding* di *layer* graf yang lebih dalam [16]. Model rekomendasi yang menggunakan mekanisme *Contrastive Learning* terbukti efektif dimana penelitian yang dilakukan oleh Zhang et al. [15] model HMCL berhasil meraih peningkatan metrik hingga 20% melalui pembelajaran kontras *multi-view* tingkat tinggi. Selain itu, penelitian Li et al. [16] membuktikan bahwa penguatan *embedding* melalui *Deep Mutual Learning* sangat efektif untuk menyeimbangkan fitur dari berbagai modalitas.

Walaupun GNN dan konteks telah banyak dieksplorasi, penelitian yang mengintegrasikan fondasi LightGCN, *Contrastive Learning*, dan dinamisnya konteks temporal belum

dilakukan secara ekstensif [17]. Studi-studi sebelumnya hanya menerapkan GNN tanpa merancang strategi untuk menyeimbangkan representasi konteks yang berpotensi menjadi *noise*.

Penelitian ini mengisi ruang tersebut dengan mengimplementasikan optimasi *dual-objective* (BPR dan *InfoNCE*) dari *Contrastive Learning* untuk menyeimbangkan ruang *embedding*. Mekanisme ini dirancang agar ketika fitur temporal disuntikan, komponen ini tidak mengaburkan sinyal kolaboratif utama, melainkan memaksimalkan *mutual information* dari berbagai augmentasi pandangan graf (*view graph*). Hasilnya model usulan dapat menjadikan informasi temporal sebagai fitur yang kokoh dan tahan terhadap *overfitting* ketika model menghadapi *data sparsity*. Maka dari itu, dataset MovieLens 1M dengan kelengkapan *timestamp* dan metadatanya merupakan landasan yang cocok untuk menganalisis gabungan ketiga komponen untuk menangkap preferensi pengguna dari waktu ke waktu.

Agar lebih fokus dan terarah berikut merupakan tujuan dari penelitian ini adalah 1) Mengembangkan dan merancang arsitektur yang tepat dari LightGCN yang terintegrasi dengan *Contrastive Learning* untuk secara efektif merepresentasikan interaksi user-item (film) dan fitur konteks temporal untuk mengatasi data sparsity dan over-smoothing dalam dataset MovieLens. Tujuan kedua adalah mencari nilai *Mean Average Precision* (MAP), *Normalized Discounted Cumulative Gain* (NDCG), *Mean Reciprocal Rank* (MRR), dan *HitRatio* dari model usulan.

B. Perbandingan dengan penelitian terdahulu

Temuan pada penelitian penting untuk dibandingkan dengan penelitian terdahulu yang relevan agar mengetahui perbedaan dan gap yang bisa dilihat pada tabel 1.

TABEL I
PERBANDINGAN DENGAN PENELITIAN TERDAHULU

Penelitian	Metode Utama	Perbedaan
Pham & Phan (2026) [14]	<i>Contextual-enhanced</i> LightGCN (CF-LightGCN) yang menggunakan fitur konten dan demografi pengguna untuk inialisasi <i>embedding</i> di awal.	Model usulan menambahkan komponen <i>Contrastive Learning</i> dan konteks temporal untuk memitigasi <i>overfitting</i> dan menangani <i>data sparsity</i> .
Nesmaoui et al. (2023) [5]	Implementasi <i>baseline</i> murni arsitektur LightGCN untuk merekomendasikan film dengan mengeksplorasi informasi struktural.	Model usulan secara eksplisit memitigasi <i>overfitting</i> yang terjadi pada model LightGCN <i>baseline</i> .
Abdolmaleki & Rezvani (2022) [11]	<i>Content-based filtering</i> dengan algoritma genetika (GA) untuk memformulasikan pembobotan fitur genre, pemeran, dan kru, menggunakan <i>embedding</i> Word2Vec (CBOW)	Model usulan mampu mengekstraksi sinyal kolaboratif tingkat tinggi dari interaksi <i>user-item</i> melalui struktur graf. Tetapi model usulan tidak menggunakan fitur genre, pemeran, dan kru, yang dipakai adalah fitur temporal.
Yuan et al. (2024) [13]	<i>ContextGNN</i> yang menggabungkan	Model usulan memiliki kompleksitas arsitektur

	representasi <i>pair-wise</i> untuk item lokal dan arsitektur <i>two-tower</i> dengan <i>embedding</i> dangkal untuk item berjarak jauh pada graf temporal.	yang lebih sederhana dengan mempertahankan satu aliran propagasi LightGCN. <i>ContextGNN</i> unggul dalam menangani rekomendasi eksploratif di luar <i>subgraph</i> pengguna berkat mekanisme fusi skor personalisasi yang dinamis.
Zhang et al. (2024) [15]	HMCL menggunakan <i>High-order GNN</i> melalui <i>meta-paths</i> yang digabungkan dengan <i>multi-view contrastive learning</i> menggunakan ekstraksi SVD.	Model usulan mengintegrasikan bias waktu secara langsung untuk mempertahankan performa di tengah pergeseran preferensi historis. Pada model usulan sinyal <i>contrastive</i> masih bersifat <i>single-view</i> sedangkan HMCL membandingkan representasi lokal dan global secara <i>multi-view</i> untuk ketahanan maksimal terhadap bias popularitas.
Valera & Mehta (2025) [19]	Perbandingan ekstensif berbagai model <i>deep learning</i> (VAE, CNN, RNN) yang diintegrasikan dengan ekstraksi sentimen teks ulasan menggunakan DistilBERT	Model usulan menangkap ketergantungan relasional graf secara spasial yang sering kali gagal dipertahankan oleh model sekuensial murni. Arsitektur yang diusulkan tidak memanfaatkan data tidak terstruktur seperti teks meskipun mampu mempertajam prediksi skor. Tetapi kembali lagi model usulan berfokus pada <i>Contrastive Learning</i> dan Konteks temporal.

II. METODE PENELITIAN

Metode penelitian merupakan serangkaian kegiatan dalam mencari kebenaran suatu studi penelitian, yang diawali dengan suatu pemikiran yang membentuk rumusan masalah sehingga menimbulkan hipotesis awal, dengan dibantu dan persepsi penelitian terdahulu, sehingga bisa diolah dan dianalisis yang akhirnya membentuk kesimpulan [20].

A. Jenis, Sifat Penelitian

Jenis penelitian ini adalah penelitian eksperimental. Dimana penelitian ini bertujuan untuk menguji efektivitas integrasi algoritma LightGCN dengan pendekatan *Contrastive Learning* (CL) dalam sistem rekomendasi berbasis konteks. Pendekatan yang dilakukan adalah pendekatan eksperimen komparatif dimana kinerja algoritma yang diusulkan dibandingkan dengan algoritma LightGCN biasa tanpa *Contrastive Learning*.

Penelitian ini bersifat deskriptif, karena menggambarkan objek yang diteliti serta menjabatkan hasil pengujian yang dilakukan pada dataset yang dapat diketahui dengan *Recall@K*, *Mean Average Precision* (MAP), *Normalized Discounted*

Cumulative Gain (NDCG), *Mean Reciprocal Rank* (MRR), dan *Hit Ratio*.

B. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini diperoleh dari dataset MovieLens 1M yang merupakan dataset publik, mencakup informasi detail berupa rating pengguna terhadap film, genre film, timestamp rating diberikan, dan metadata film lainnya dengan jumlah data sebanyak satu juta data rating dari 6.000 pengguna dan 4.000 film agar model dapat memahami konteks yang terdapat dalam dataset. Proses pengumpulan data dilakukan melalui metode dokumentasi, yakni dengan mengunduh data yang telah tersedia secara publik dari platform MovieLens dalam format .csv. Terdapat empat file .csv pada dataset MovieLens ini antara lain *Links*, *Movies*, *Ratings*, dan *Tags*.

C. Data Preparation

Data yang sudah didapatkan selanjutnya diolah melalui tahapan *data preparation* atau *data preprocessing*. Pada proses ini, data akan diolah kembali untuk menghilangkan data yang tidak konsisten atau menghapus data yang kurang penting pada proses pengolahan nantinya. Pada file *ratings.csv* terdapat empat kolom antara lain *userId*, *movieId*, *rating*, *timestamp* dapat dilihat pada tabel II.

TABEL II
KONDISI AWAL FILE RATINGS.CSV

userId	movieId	rating	timestamp
1	1	4	964982703
1	3	4	964981247
1	6	4	964982224
1	47	5	964983815
1	50	5	964982931

Pertama melakukan kalkulasi dimensi global dengan mencari indeks pengguna (*user_idx*) dan item (*item_idx*) maksimum diseluruh dataset untuk menentukan kapasitas matrix *embedding*. Kedua proses yang dilakukan adalah melakukan ekstraksi fitur temporal dimana kolom *timestamp* pada file *ratings.csv* dipecah menjadi beberapa dimensi kategori yaitu *year*, *month*, *day*, *times of day* (*morning/afternoon/evening/night*), *season* (*spring/summer/autumn/winter*), dan *isWeekend* (indikator akhir pekan) penambahan kolomnya dapat dilihat pada tabel III.

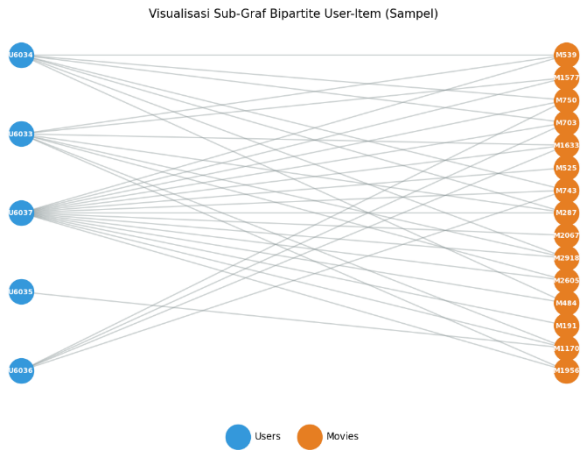
TABEL III
PENAMBAHAN KOLON TEMPORAL DARI KOLON TIMESTAMP

year	month	day	times_of_day	isWeekend	season
2000	4	Tuesday	night	FALSE	spring
2000	4	Tuesday	night	FALSE	spring
2000	4	Tuesday	night	FALSE	spring
2000	4	Tuesday	night	FALSE	spring
2000	4	Tuesday	night	FALSE	spring

Ketiga proses yang dilakukan adalah transformasi fitur temporal, dimana kolom *day*, *times of day*, *isWeekend*, dan *season* dikonversi menggunakan *LabelEncoder* menjadi indeks

angka yang dimulai dari nol. Selain itu dilakukan proses *Zero-Indexing* waktu dimana fitur tahun dinormalisasi menjadi *year_idx* dengan menghitung selisih terhadap tahun terkecil dalam data, sementara bulan disesuaikan menjadi rentang 0-11 agar sesuai dengan struktur *embedding*.

Tahap keempat adalah mengubah data tabular menjadi graf karena model yang diusulkan menggunakan GNN lebih spesifiknya LightGCN. Secara spesifik data tabular dibentuk dan direpresentasikan sebagai Graf Bipartit. Interaksi historis antara pengguna dan film secara teori didefinisikan sebagai graf bipartit tak berarah ($G = (\mathcal{V}, \mathcal{E})$), dimana himpunan *node* $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ menaungi entitas pengguna (\mathcal{U}) dan item (\mathcal{I}). Agar dapat diproses oleh arsitektur GNN, graf G tersebut direpresentasikan dalam bentuk *matrix adjacency* $A \in \mathbb{R}^{(N+M) \times (N+M)}$. Untuk mencegah tumpang tindih indeks antara pengguna dan item di dalam matrix yang sama, dilakukan *re-indexing* dengan menggeser ID item sebanyak total jumlah pengguna ($ID_{item} + |\mathcal{U}|$). Graf yang telah dibuat harus dinormalisasi untuk menyeimbangkan pengaruh antara *node* yang sangat aktif (derajat koneksi tinggi) dengan *node* pasif, *matrix adjacency* A dinormalisasi secara simetris menggunakan formula $\tilde{D}^{-1/2} A \tilde{D}^{-1/2}$, dimana D adalah matriks derajat diagonal. Bentuk ternormalisasi inilah yang secara langsung digunakan dalam proses perambatan pada layer arsitektur model. Representasi graf bipartit dapat dilihat pada Gambar 1.



Gbr. 1 Visualisasi Sub-Graf Bipartite User-Item

Kelima dataset dibagi (*data splitting*) secara sekuensial berdasarkan urutan kronologis untuk mencerminkan skenario prediksi di dunia nyata. Rasio pembagian yang ditetapkan adalah 80% data latih (*training*), 10% data validasi (*validation*), 10% data uji (*testing*). Data validasi secara khusus digunakan untuk *setting hyperparameter* dan pencegahan *overfitting*, sementara data uji tetap terjaga sebagai *unseen data* untuk evaluasi akhir.

Keenam prosedur pelatihan dan optimasi dilakukan menggunakan fungsi kerugian gabungan

$$L_{total} = L_{BPR} + \lambda \mathcal{L}_{CL} \quad (1)$$

Dimana L_{BPR} adalah *Bayesian Personalized Ranking Loss*. Selama pelatihan, sistem menerapkan *Negative Sampling*, dimana setiap interaksi positif, satu item diambil secara acak sebagai sampel negatif guna melatih model dalam membedakan preferensi pengguna. Parameter optimasi meliputi *learning rate* 0.001 menggunakan *optimizer* Adam dengan ukuran *batch* 1024.

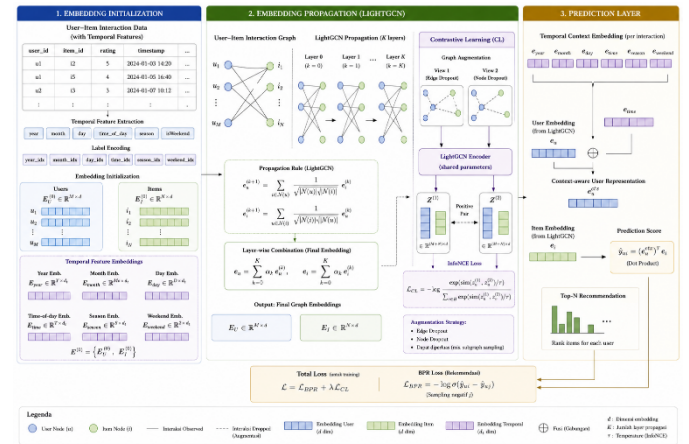
Kinerja model dievaluasi pada daftar rekomendasi *Top-20* ($K = 20$). Adapun lima metrik utama yang digunakan untuk mengukur efektivitas rekomendasi antara lain *Hit Ratio* (HR) untuk keberhasilan prediksi, serta NDCG, MRR, Recall, dan MAP untuk mengukur kualitas urutan dan presisi rekomendasi yang diberikan kepada pengguna.

D. Arsitektur Model

Model usulan menggunakan kombinasi LightGCN, *Contrastive Learning*, serta konteks temporal yang digunakan secara *embedding* untuk meningkatkan kualitas representasi *user-item*. Arsitektur terdiri dari tiga komponen utama yaitu

1. *Embedding Initialization*
2. *Embedding Propagation*.
3. *Prediction Layer*.

Pada model usulan konteks temporal tidak dimasukkan kedalam graf melainkan digunakan sebagai *contextual-feature embedding* yang digabungkan dengan *embedding user* setelah proses propagasi. Arsitektur dapat dilihat pada gambar 2.



Gbr. 2 Visualisasi Arsitektur Model Usulan

1) *Embedding Initialization*.

Pada MovieLens 1M *dataset* terdapat interaksi *user-item* yang mengandung informasi waktu, dimana setiap interaksi direpresentasikan sebagai :

$$(u, i, t)$$

Dimana $u \in \mathcal{U}$ adalah *user*, $i \in \mathcal{I}$ adalah *item* (film), dan t adalah *timestamp*. Setelah itu *user* dan *item embedding* awal diinisialisasi sebagai :

$$E_U^{(0)} \in \mathbb{R}^{M \times d}, E_I^{(0)} \in \mathbb{R}^{N \times d} \quad (2)$$

Dimana M dan N adalah jumlah *user* dan *item*, serta d adalah dimensi *embedding*. Selanjutnya adalah *temporal feature*

encoding dimana dari kolom *timestamp* informasi waktu diubah menjadi

- Tahun (*year*)
- Bulan (*month*)
- Hari (*day*)
- Waktu dalam hari (*time_of_day*)
- Musim (*season*)
- Akhir Pekan (*isWeekend*)

Setiap fitur dikodekan menggunakan *label encoding* menjadi indeks

$$(year_idx, month_idx, \dots)$$

Masing-masing informasi waktu direpresentasikan dalam *embedding*.

$$e_{year}, e_{month}, e_{day}, e_{time}, e_{season}, e_{weekend}$$

2) Embedding Propagation.

Graf interaksi *user-item* direpresentasikan sebagai graf bipartit $\mathcal{G} = (U, I, E)$. LightGCN digunakan untuk mempelajari representasi *node* melalui propagasi *embedding* tanpa transformasi non-linear. Selanjutnya adalah membuat *propagation rule* dimana *embedding* pada *layer* ke- $k + 1$ dihitung sebagai

$$e_u^{(k+1)} = \sum_{i \in N(u)} \frac{1}{\sqrt{|N(u)||N(i)|}} e_i^{(k)} \quad [21] (3)$$

$$e_i^{(k+1)} = \sum_{u \in N(i)} \frac{1}{\sqrt{|N(i)||N(u)|}} e_u^{(k)}$$

Dimana $N(u)$ dan $N(i)$ adalah himpunan tetangga dari *user-item*. Pada tahapan *layer aggregation* dimana *embedding* akhir diperoleh dengan menjumlahkan seluruh *layer* :

$$e_u = \sum_{k=0}^K \alpha_k e_u^{(k)}, \quad e_i = \sum_{k=0}^K \alpha_k e_i^{(k)} \quad [21] (4)$$

Dimana α_k adalah bobot *layer*

3) Modul Contrastive Learning.

Contrastive Learning digunakan untuk meningkatkan kualitas representasi pendekatan yang digunakan adalah augmentasi graf. Dua *view graph* dihasilkan melalui *edge dropout* dan *node dropout*. Kedua graf diproses menggunakan LightGCN *encoder* dengan parameter yang sama untuk menghasilkan.

$$z^{(1)}, z^{(2)}$$

Selain itu juga dihitung *contrastive loss* dimana model usulan akan dilatih menggunakan *InfoNCE loss* :

$$\mathcal{L}_{CL} = -\log \frac{\exp(\text{sim}(z_u^{(1)}, z_u^{(2)})/\tau)}{\sum_v \exp(\text{sim}(z_u^{(1)}, z_v^{(2)})/\tau)} \quad [22] (5)$$

Dimana :

- $\text{sim}(\cdot)$ adalah *cosine similarity*.
- τ adalah *temperature parameter*.

4) Integrasi Konteks Temporal.

Konteks temporal diintegrasikan pada tahap akhir sebagai *embedding* tambahan. *Embedding* konteks temporal dapat dihitung sebagai berikut :

$$e_{time} = e_{year} + e_{month} + e_{day} + e_{time_of_day} + e_{season} + e_{weekend} \quad (6)$$

Embedding konteks temporal ini selanjutnya digabungkan dengan *embedding user* dimana pendekatan ini memungkinkan model menangkap dinamika waktu tanpa mengubah struktur graf :

$$e_u^{ctx} = e_u + e_{time} \quad (7)$$

5) Prediction Layer.

Skor preferensi antara *user-item* (film) dihitung menggunakan fungsi *dot product*.

$$\hat{Y}_{ui} = (e_u^{ctx})^T e_i \quad (8)$$

Model akan dioptimasi menggunakan dua fungsi loss yaitu *Bayesian Personalized Ranking (BPR) loss* dan *Contrastive loss*.

$$\mathcal{L}_{BPR} = -\log \sigma(\hat{Y}_{ui} - \hat{Y}_{uj}) \quad [23] (9)$$

Sehingga fungsi *total loss* dapat didefinisikan sebagai berikut :

$$\mathcal{L} = \mathcal{L}_{BPR} + \lambda \mathcal{L}_{CL} \quad (10)$$

Dimana λ adalah parameter penyeimbang.

III. HASIL DAN PEMBAHASAN

Berikut merupakan hasil dan pembahasan dari eksperimen yang telah dilakukan.

A. Analisis Hasil Eksperimen Ablasi

Penelitian ini mengevaluasi performa model melalui studi ablas untuk memahami kontribusi dari setiap komponen yang diusulkan. Hasil evaluasi menggunakan metrik standar pada daftar rekomendasi *Top-20* disajikan pada Tabel IV.

TABEL IV
HASIL STUDI ABLASI PADA DATASET MOVIELENS 1M

Model	HitRatio	NDCG	MRR	Recall	MAP
LightGCN Base	66.12%	0.1807	0.3255	0.0615	0.0937
LightGCN + CL	68.93%	0.172	0.3314	0.0723	0.0833
Usulan (LightGCN + CL + Temporal)	70.74%	0.1888	0.3383	0.0628	0.0959

Berdasarkan data pada Tabel III terlihat peningkatan secara bertahap pada metrik *HitRatio*. Model *baseline* (LightGCN Base) menghasilkan metrik dasar dengan *HitRatio* 66.12% dan NDCG 0.1807. Terdapat peningkatan kemampuan penemuan

item dimana penambahan dari modul *Contrastive Learning* memberikan dorongan awal pada *HitRatio* menjadi 68.93% dan *Recall* 0.0723, namun memicu sedikit degradasi pada presisi peringkat NDCG turun menjadi 0.1720 dari model *baseline*. Degradasi ini merupakan efek yang lazim sesuai pada penelitian Zhang (2024) dimana fungsi objektif *Contrastive Learning* adalah untuk memaksakan penyebaran representasi secara merata [15], [18], sehingga model mengorbankan ketajaman prediksi pada *item-item* berinteraksi tinggi demi mengeksplorasi *item long tail*.

Namun dengan integrasi fitur temporal memberikan peningkatan yang paling signifikan hingga mencapai angka 70.74%. Ini membuktikan bahwa konteks temporal (*year, month, day, times_of_day, isWeekend, dan season*) cukup efektif dalam menangkap dinamika preferensi pengguna pada dataset MovieLens 1M. Selain itu, model yang diusulkan juga lebih sukses dalam memunculkan setidaknya satu item relevan dalam daftar rekomendasi pengguna. Model usulan memiliki peningkatan 4.62% pada metrik *HitRatio* saat dibandingkan dengan model *baseline* sekaligus memperbaiki pemerinkatan NDCG sebesar 0.1888, dan MRR 0.3383. Meskipun terdapat banyak peningkatan metrik pada model usulan bila dibandingkan dengan model *baseline* dan model kedua, pada model usulan terdapat penurunan metrik *Recall* 0.0628 menurun bila dibandingkan dengan model kedua dengan *Recall* sebesar 0.0723. Peningkatan metrik secara keseluruhan mengindikasikan bahwa model usulan menjadi arsitektur paling optimal.

B. Efisiensi Komputasi

Analisis waktu per *epoch* menunjukkan bahwa model yang diusulkan tetap efisien dari sisi komputasi yang ditunjukkan pada Tabel V.

TABEL V
WAKTU KOMPUTASI MODEL PER EPOCH

Model	Waktu per Epoch (detik)
LightGCN Base	50 detik
LightGCN + CL	132 detik
Usulan (LightGCN + CL + Temporal)	132 detik

LightGCN Base membutuhkan waktu komputasi sekitar 50 detik per *epoch*. Ketika modul *Contrastive Learning* ditambahkan maka menambahkan beban komputasi sehingga membutuhkan waktu sekitar 132 detik per *epoch*, dikarenakan adanya proses augmentasi graf dan perhitungan *InfoNCE Loss*. Model yang diusulkan dengan menambahkan fitur temporal tidak menambah waktu komputasi, tetap bertahan pada kisaran 132 detik per *epoch*. Hal ini menunjukkan bahwa arsitektur LightGCN, *Contrastive Learning* yang ditambahkan konteks temporal yang dirancang sangat ringan namun memberikan dampak akurasi yang cukup signifikan.

C. Evaluasi Tujuan Awal dan Temuan Utama

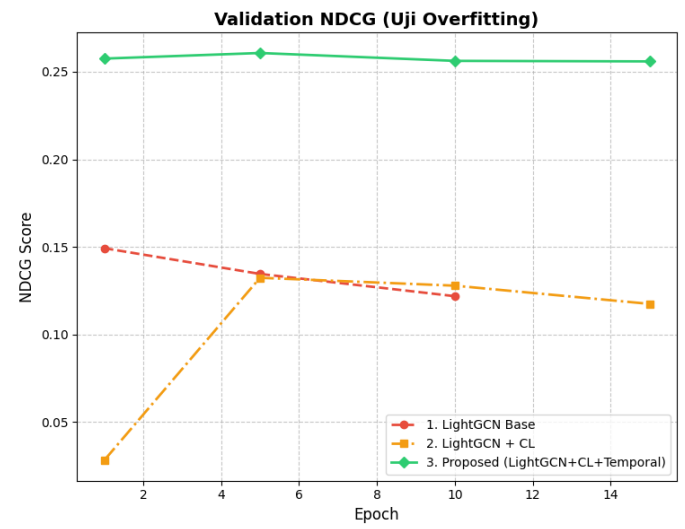
Penelitian ini awalnya dirancang untuk memitigasi anomali *over-smoothing* dan memecahkan masalah *data sparsity* pada arsitektur berbasis graf. Observasi empiris terhadap dinamika

log training ternyata mengungkapkan hal yang berbeda. Hambatan yang ditemukan pada dataset interaksi *user-item* (film) bukanlah homogenitas *embedding* akibat *over-smoothing*, melainkan kerentanan model *baseline* terhadap *overfitting* yang terjadi sangat agresif.

Pergeseran fokus temuan mencerminkan esensi dari evaluasi empiris dalam *machine learning* [24]. Arsitektur yang diusulkan tetap memiliki nilai kebaruan dan urgensi tinggi, bukan semata-mata menjawab tujuan diawal secara linier, melainkan secara tidak terduga menghadirkan solusi struktural terhadap masalah fundamental generalisasi model yang gagal ditangani oleh algoritma konvensional (model *baseline*).

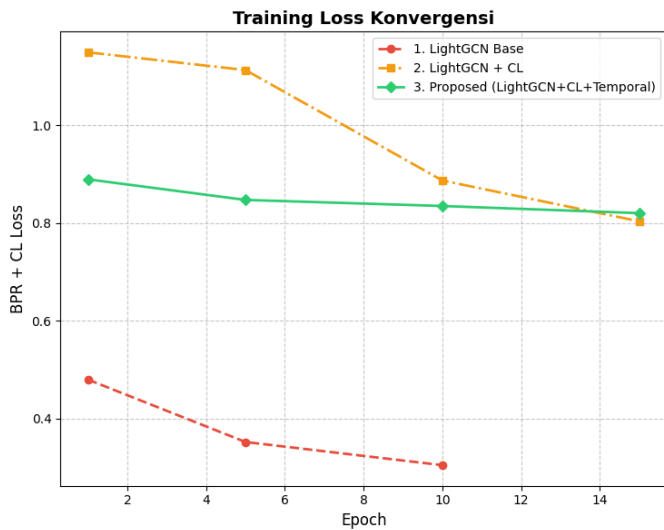
D. Overfitting pada Baseline

Analisis terhadap mekanisme *early stopping* dan kurva validasi mengungkap kekurangan pada arsitektur LightGCN tradisional. Model *baseline* mencapai kapasitas generalisasi maksimal nyaris secara instan dimana pada *epoch* pertama dengan nilai NDCG sebesar 0.1493. Pada *epoch* berikutnya, performa pada data validasi menurun secara signifikan menjadi 0.1346 (*Epoch* 5) dan 0.1219 (*Epoch* 10) pada gambar 3.



Gbr. 3 Grafik Uji Overfitting

Kontradiksi terjadi karena di saat yang bersamaan, nilai training loss terus menurun secara konsisten dari 0.4799 menjadi 0.3049 pada gambar 4. Pada model dasar hanya melakukan optimasi pada satu *function loss* tunggal, yaitu *Bayesian Personalized Ranking (L_{BPR})* yang berfokus hanya pada membedakan skor antar item positif dan negatif. Divergensi antara *training loss* yang mengecil dan validation score yang memburuk merupakan indikasi absolut dari proses penghafalan data (*memorization*). Model *baseline* mengoptimalkan *Loss Function* BPR dengan cara menghafal secara spesifik graf interaksi pelatihan, kehilangan kemampuannya untuk mengekstrapolasi pola tersebut ke dalam data interaksi yang belum pernah dilihat sebelumnya.



Gbr. 4 Training Loss Konvergensi

E. Efektivitas Model Usulan terhadap Overfitting

Integrasi modul *Contrastive Learning* dan konteks temporal pada LightGCN secara empiris berhasil mengintervensi laju *overfitting* tersebut dan bertindak sebagai mekanisme regularisasi yang sangat presisi. Kurva pembelajaran pada model LightGCN+CL+Temporal (garis hijau) pada gambar 3 menunjukkan stabilitas konvergensi yang lebih stabil. Model usulan baru mencapai puncak pada *epoch* ke 5 dengan lonjakan metrik NDCG mencapai 0.2607.

Overfitting dapat ditahan pada model usulan karena pada modul *Contrastive Learning* menggunakan mekanisme augmentasi graf (melalui *node/edge dropout*). Injeksi *noise* terkontrol saat proses pelatihan yang memaksa ruang representasi agar tidak bergantung pada jalur koneksi spesifik yang terdapat pada data *training*. Ditambah dengan modul temporal yang menyuntikkan bias waktu, model dipaksa untuk mempelajari invariansi dan semantik yang lebih dalam dari preferensi pengguna, sehingga metrik pada fase pengujian akhir tetap solid dan tidak runtuh pasca *epoch* awal.

Nilai *loss* pada model usulan dan model kedua yang menggunakan *Contrastive Learning* harus meminimalkan fungsi objektif ganda secara simultan $L_{total} = L_{BPR} + \lambda_2 L_{reg} + \lambda_2 L_{CL}$. Komponen L_{CL} (*InfoNCE loss*) berperan untuk memaksa model terus menyeimbangkan antar dua tugas yang saling tarik-menarik yaitu menjaga susunan peringkat yang presisi (lewat BPR) dan menjaga ruang *embedding* tetap tersebar merata agar tidak bias terhadap popularitas (lewat CL). Hal ini menyebabkan total nilai *loss* konvergen pada titik keseimbangan secara normal lebih tinggi.

F. Hubungan Temuan Utama dengan Data Sparsity dan Over-smoothing

Temuan *overfitting* ini pada dasarnya memiliki keterikatan yang kuat dengan kondisi *data sparsity* [25]. Dalam struktur graf bipartit yang renggang (*sparse*) jumlah sinyal positif (interaksi valid) sangat terbatas dibandingkan dengan dengan jumlah kombinasi kemungkinan. Kondisi kelangkaan inilah

yang memicu *neural network* tanpa regularisator pada model *baseline* untuk mengambil jalan tercepat dengan menghafal data secara prematur. Model usulan dengan kombinasi LightGCN+CL+Konteks Temporal berhasil menunda dan memitigasi *overfitting* merupakan bukti tidak langsung bahwa arsitektur ini memiliki ketahanan yang lebih baik dalam memproses MovieLens 1M *dataset* dengan *data sparsity*.

Terkait dengan *over-smoothing*, kedalaman arsitektur dibatasi dalam tiga lapisan (*tiga layer*) dalam pengujian ini membuat dampak *over-smoothing* belum menjadi penyebab dari penurunan performa model usulan. Meski demikian, efek *uniformity* yang dihasilkan oleh *Contrastive Loss* secara teoritis mengamankan jarak semantik antar *node* di ruang laten. Fungsi ini menjamin bahwa representasi pengguna dan item tidak akan runtuh ke titik yang sama, memberikan fondasi arsitektural yang kokoh seandainya graf diekspansi menjadi lebih dalam di penelitian mendatang.

G. Perbandingan dengan penelitian terdahulu

Temuan pada penelitian penting untuk dibandingkan dengan penelitian terdahulu yang relevan agar mengetahui perbedaan dan *gap* yang bisa dilihat pada tabel 5.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian serta pembahasan yang dilakukan oleh penulis tentang Sistem Rekomendasi Film Berbasis Konteks Menggunakan Graph Neural Network Dengan *Contrastive Learning* Pada Dataset MovieLens dapat disimpulkan beberapa hal berikut :

- 1) Model LightGCN *baseline* terbukti rentan terhadap *overfitting* pada graf interaksi yang renggang (*sparse*), dimana model mengalami degradasi generalisasi secara instan sejak *epoch* pertama. Model usulan yang menambahkan *Contrastive Learning*, dan Konteks temporal secara empiris berhasil bertindak sebagai mekanisme regularisasi yang kuat. Pendekatan ini terbukti mampu menahan laju konvergensi prematur, menjaga stabilitas kurva pembelajaran pada fase validasi, dan mencegah representasi model kolaps akibat *noise*. Sehingga kombinasi tiga komponen ini dapat menangani *overfitting*, serta mengatasi *data sparsity* secara tidak langsung dengan mengekstrak pola pergeseran preferensi pengguna yang tervalidasi oleh waktu, sehingga tidak bergantung pada interaksi di masa lalu.
- 2) Kombinasi algoritma LightGCN dengan *Contrastive Learning* dan konteks temporal menghasilkan evaluasi performa yang sangat kompetitif, ditandai dengan pencapaian *Hit Ratio* yang melonjak hingga 70,74%, *Normalized Discounted Cumulative Gain* (NDCG) sebesar 0,1888, *Mean Reciprocal Rank* (MRR) sebesar 0.3383, *Mean Average Precision* (MAP) sebesar 0.0959, dan *Recall@20* sebesar 0.0628.

Pada penelitian kali ini mungkin terdapat banyak kekurangan dalam pelaporan maupun hasil pengujian pada penelitian kali ini. Maka dari itu untuk pengembangan lebih lanjut maka di perlukan beberapa saran sebagai berikut:

- 1) Meskipun fitur temporal (hari dan musim) telah terbukti efektif namun dapat dikembangkan kembali dengan mengintegrasikan *Knowledge Graph*. Konteks tambahan seperti aktor, sutradara, dan genre dapat digabungkan agar tidak hanya memperkuat representasi *semantic embedding*, tetapi juga mengurangi terjadinya *data sparsity*.
- 2) Arsitektur dapat diuji pada dataset yang lebih besar (misalnya MovieLens 25M) atau pada domain industri yang berbeda.

REFERENSI

- [1] R. Lavanya and B. Bharathi, "Movie Recommendation System to Solve Data Sparsity Using Collaborative Filtering Approach," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 20, no. 5, p. 88:1-88:14, Jul. 2021, doi: 10.1145/3459091.
- [2] T. Fang, Z. Xiao, C. Wang, J. Xu, X. Yang, and Y. Yang, "DropMessage: Unifying Random Dropping for Graph Neural Networks," *AAAI*, vol. 37, no. 4, pp. 4267–4275, Jun. 2023, doi: 10.1609/aaai.v37i4.25545.
- [3] M. Hasan and F. Roy, "An Item–Item Collaborative Filtering Recommender System Using Trust and Genre to Address the Cold-Start Problem," *Big Data and Cognitive Computing*, vol. 3, no. 3, p. 39, Sep. 2019, doi: 10.3390/bdcc3030039.
- [4] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph Neural Networks in Recommender Systems: A Survey," Apr. 02, 2022, *arXiv: arXiv:2011.02260*. doi: 10.48550/arXiv.2011.02260.
- [5] R. Nesmaoui, M. Louhichi, and M. Lazaar, "A Collaborative Filtering Movies Recommendation System based on Graph Neural Network," *Procedia Computer Science*, vol. 220, pp. 456–461, 2023, doi: 10.1016/j.procs.2023.03.058.
- [6] P. Chanda, S. Yadav, J. B. Pal, and G. Nagakalyani, "Solving the Cold Start problem in Recommendation Systems - Case Study on MovieLens Dataset," Sep. 2022.
- [7] U. Javed, K. Shaikat, I. A. Hameed, F. Iqbal, T. Mahboob Alam, and S. Luo, "A Review of Content-Based and Context-Based Recommendation Systems," *Int. J. Emerg. Technol. Learn.*, vol. 16, no. 03, p. 274, Feb. 2021, doi: 10.3991/ijet.v16i03.18851.
- [8] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin, "Context-Aware Recommender Systems".
- [9] K. Verbert *et al.*, "Context-Aware Recommender Systems for Learning: A Survey and Future Challenges," *IEEE Transactions on Learning Technologies*, vol. 5, no. 4, pp. 318–335, Oct. 2012, doi: 10.1109/TLT.2012.11.
- [10] M. Del Carmen Rodríguez-Hernández and S. Ilarri, "AI-based mobile context-aware recommender systems from an information management perspective: Progress and directions," *Knowledge-Based Systems*, vol. 215, p. 106740, Mar. 2021, doi: 10.1016/j.knosys.2021.106740.
- [11] A. Abdolmaleki and M. Rezvani, "An optimal context-aware content-based movie recommender system using genetic algorithm: a case study on MovieLens dataset," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 36, pp. 1–27, Dec. 2022, doi: 10.1080/0952813X.2022.2153279.
- [12] Q. Gao and P. Ma, "Graph Neural Network and Context-Aware Based User Behavior Prediction and Recommendation System Research," *Computational Intelligence and Neuroscience*, vol. 2020, no. 1, p. 8812370, 2020, doi: 10.1155/2020/8812370.
- [13] Y. Yuan *et al.*, "ContextGNN: Beyond Two-Tower Recommendation Systems," Nov. 29, 2024, *arXiv: arXiv:2411.19513*. doi: 10.48550/arXiv.2411.19513.
- [14] D.-Q.-H. Pham and H.-T. Phan, "A Contextual-Enhanced LightGCN for Movie Recommendation Systems," *Journal of Technical Education Science*, vol. 21, no. 01, pp. 71–80, Feb. 2026, doi: 10.54644/jte.2026.2069.
- [15] A. Zhang, B. Li, K. Wang, Y. Yang, X. Zhang, and M. Lei, "HMCL: High-order Graph Neural Networks with Multi-view Contrastive Learning for Recommendation," *Procedia Computer Science*, vol. 242, pp. 1147–1154, 2024, doi: 10.1016/j.procs.2024.08.182.
- [16] J. Li, C. Yang, G. Ye, and Q. V. H. Nguyen, "Graph neural networks with deep mutual learning for designing multi-modal recommendation systems," *Information Sciences*, vol. 654, p. 119815, Jan. 2024, doi: 10.1016/j.ins.2023.119815.
- [17] W. Cao, Y. Liu, G. Cao, and Z. He, "Implicit user relationships across sessions enhanced graph for session-based recommendation," *Information Sciences*, vol. 609, pp. 1–14, Sep. 2022, doi: 10.1016/j.ins.2022.07.062.
- [18] P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive Representation Learning: A Framework and Review," *IEEE Access*, vol. 8, pp. 193907–193934, 2020, doi: 10.1109/ACCESS.2020.3031549.
- [19] M. Valera and Dr. R. Mehta, "Advanced Deep Learning Models for Improving Movie Rating Predictions: A Benchmarking Study," *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, vol. 4, no. 4, p. 100200, Dec. 2024, doi: 10.1016/j.tbench.2025.100200.
- [20] S. H. Sahir, *E-Book Metodologi Penelitian Syafrida*. Yogyakarta: Penerbit KBM Indonesia, 2026. Accessed: Apr. 15, 2026. [Online]. Available: <https://repositori.uma.ac.id/bitstream/123456789/16455/1/E-Book%20Metodologi%20Penelitian%20Syafrida.pdf>
- [21] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," Jul. 07, 2020, *arXiv: arXiv:2002.02126*. doi: 10.48550/arXiv.2002.02126.
- [22] A. van den Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," Jan. 22, 2019, *arXiv: arXiv:1807.03748*. doi: 10.48550/arXiv.1807.03748.
- [23] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian Personalized Ranking from Implicit Feedback," May 09, 2012, *arXiv: arXiv:1205.2618*. doi: 10.48550/arXiv.1205.2618.

- [24] M. Herrmann *et al.*, “Position: Why We Must Rethink Empirical Research in Machine Learning,” May 25, 2024, *arXiv*: arXiv:2405.02200. doi: 10.48550/arXiv.2405.02200.
- [25] L. Petrini, F. Cagnetta, E. Vanden-Eijnden, and M. Wyart, “Learning sparse features can lead to overfitting in neural networks*,” *J. Stat. Mech.*, vol. 2023, no. 11, p. 114003, Nov. 2023, doi: 10.1088/1742-5468/ad01b9.