NTSFormer: A Self-Teaching Graph Transformer for Multimodal Isolated Cold-Start Node Classification

Jun Hu, Yufei He, Yuan Li, Bryan Hooi, Bingsheng He

National University of Singapore
jun.hu@nus.edu.sg, {yufei.he, li.yuan}@u.nus.edu, {dcsbhk, dcsheb}@nus.edu.sg

Abstract

Isolated cold-start node classification on multimodal graphs is challenging because such nodes have no edges and often have missing modalities (e.g., absent text or image features). Existing methods address structural isolation by degrading graph learning models to multilayer perceptrons (MLPs) for isolated cold-start inference, using a teacher model (with graph access) to guide the MLP. However, this results in limited model capacity in the student, which is further challenged when modalities are missing. In this paper, we propose **Neighbor-to-Self Graph Transformer (NTSFormer)**, a unified Graph Transformer framework that jointly tackles the isolation and missing-modality issues via a self-teaching paradigm. Specifically, NTSFormer uses a cold-start attention mask to simultaneously make two predictions for each node: a "student" prediction based only on self information (i.e., the node's own features), and a "teacher" prediction incorporating both self and neighbor information. This enables the model to supervise itself without degrading to an MLP, thereby fully leveraging the Transformer's capacity to handle missing modalities. To handle diverse graph information and missing modalities, NTSFormer performs a one-time multimodal graph pre-computation that converts structural and feature data into token sequences, which are then processed by Mixture-of-Experts (MoE) Input Projection and Transformer layers for effective fusion. Experiments on public datasets show that NTSFormer achieves superior performance for multimodal isolated cold-start node classification.

Code — https://github.com/CrawlScript/NTSFormer

Introduction

Multimodal graphs, whose nodes are associated with diverse data modalities, are prevalent in real-world scenarios, such as social networks and product co-purchase networks. **Node classification on multimodal graphs** is a critical task with applications in areas like fake news detection (Zhang et al. 2024), product tagging (Zhu et al. 2024), and more (Cai et al. 2024). Multimodal Graph Neural Networks (GNNs) emerge as a promising solution by jointly modeling multimodal content and graph structure (Wei et al. 2019; Tao et al. 2020).

Multimodal Isolated Cold-Start Node Classification. Real-world multimodal graphs frequently contain isolated cold-start nodes (Moscati 2024; Shen et al. 2024)—newly introduced isolated nodes with limited information, such as newly registered users in social networks. These isolated cold-start nodes typically pose two challenges for multimodal node classification models: (1) isolation, where the nodes have no connections (Zhang et al. 2022; He and Ma 2022; Wang et al. 2024b), and (2) missing modalities, where certain data (e.g., text or images) are absent (Wu et al. 2024; Ganhör et al. 2024). For example, as shown on the right side of Figure 1, a new user in a social network may have no connections (isolation) and only a profile image without providing a description (missing text). Such scenarios make node classification particularly difficult, as models cannot leverage the graph structure and must instead depend on multimodal features, which are sometimes missing.

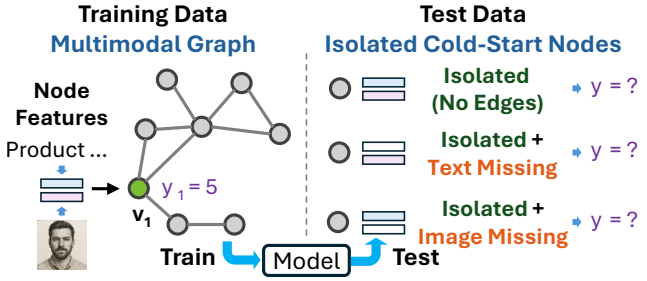


Figure 1: The **multimodal isolated cold-start node classification** task focuses on classifying isolated cold-start nodes that have no edges and may be missing certain modalities.

To illustrate this challenge, Figure 2 presents the performance of multilayer perceptrons (MLPs) and GNNs on this task across public datasets. Popular GNNs such as GraphSAGE (Hamilton, Ying, and Leskovec 2017), MMGCN (Wei et al. 2019), and MGAT (Tao et al. 2020) generally outperform MLPs on non-cold-start tasks. However, as illustrated in Figure 2, these GNNs exhibit poor performance in isolated cold-start scenarios and can even be outperformed by simple MLPs. The significantly degraded performance of GNN-based models on multimodal isolated cold-start node classification highlights the pressing need for robust strategies specifically designed for this task.

Teaching MLP-Students to Alleviate the Isolation Challenge. As shown in Figure 2, MLP often outperforms

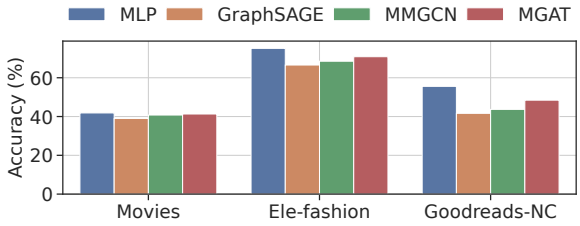
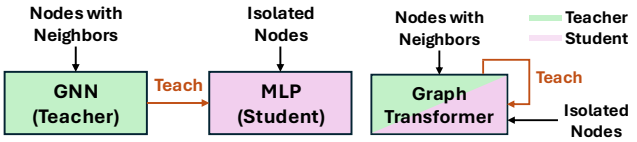


Figure 2: Performance on multimodal isolated cold-start node classification. General GNNs (GraphSAGE) and multimodal GNNs (MMGCN, MGAT) even underperform MLPs.



(a) GLNN (MLP-Student) (b) Ours (Self-Teaching)

Figure 3: NTSFormer uses a cold-start mask to make two predictions: a "student" prediction based on self-features only, and a "teacher" prediction with both self and neighbor information, enabling it to supervise itself without degrading to an MLP, thereby leveraging Transformers' capacity.

GNNs in isolated cold-start scenarios. One possible reason is that MLPs are trained and tested under consistent conditions, using only node-level features, whereas GNNs suffer from a train-test distribution shift due to the isolation of test-time nodes. Despite this advantage, a major limitation of MLP-students is their inability to exploit graph structure during training, missing out on valuable relational signals. To address this, recent methods such as GLNN (Zhang et al. 2022), SGKD (He and Ma 2022), and SimMLP (Wang et al. 2024b) adopt a teacher-student paradigm, as illustrated in Figure 3a, where a GNN teacher distills its structural knowledge into a structure-agnostic MLP-student. This approach effectively bridges the gap caused by isolation and improves generalization in isolated cold-start settings. However, such MLP-student methods often suffer from capacity bottlenecks in multimodal scenarios where they must handle more complex challenges, including missing modalities.

From MLP-Students to Self-Teaching Graph Transformers. Motivated by the capacity limitations of MLP-students, we propose **Neighbor-to-Self Graph Transformer (NTSFormer)**, a unified Graph Transformer with a *self-teaching* paradigm as shown in Figure 3b. Rather than resorting to a simple MLP, we harness the power of Transformers to handle both structural isolation and missing modalities. Specifically, it produces two predictions for each node: a *student* prediction based only on self-features, and a *teacher* prediction that also incorporates neighbor information. This design ensures that training on graphs aligns seamlessly with inference on isolated cold-start nodes—eliminating the need to degrade the model to an MLP for such cases. To handle diverse graph data and missing modalities, NTSFormer performs a one-time multimodal graph pre-computation, converting both structural and multimodal feature information into token sequences, which are then processed by Mixture-of-Experts (MoE) Input Projection and Transformer layers for effective fusion. Extensive experiments on public datasets demonstrate that NTSFormer consistently outperforms various baseline methods.

Our main contributions are as follows:

- We introduce **NTSFormer**, a Graph Transformer that unifies the modeling of structural isolation and missing modalities in multimodal isolated cold-start settings via self-teaching.
- To enable self-teaching within a unified Graph Transformer, we propose a *cold-start mask* that enables a Graph Transformer to produce two predictions per node: a student prediction based solely on self-features and a teacher prediction incorporating neighbor information.
- To handle diverse graph information and missing modalities, we design Multimodal Graph Pre-computation that captures various features from multimodal graphs, along with MoE Input Projection for effectively processing the various features before feeding them into Transformers.
- We conduct extensive experiments on public multimodal graph benchmarks, demonstrating the superior performance of NTSFormer over various baselines.

Related Work

Multimodal Graph Neural Networks

GNNs such as GCN (Kipf and Welling 2017), GAT (Velickovic et al. 2018), and GraphSAGE (Hamilton, Ying, and Leskovec 2017) have achieved strong results for node classification by leveraging both node features and graph structure via message passing. To handle multimodal graphs, MMGCN (Wei et al. 2019) uses modality-specific GNNs for intra- and inter-modal modeling. MGAT (Tao et al. 2020) applies attention to weight modalities. MIG (Hu et al. 2024) applies GNNs with modality-independent receptive fields.

Multimodal Isolated Cold-Start Node Classification

This task involves both structural isolation and missing modalities. Under structural isolation, traditional GNNs suffer from a train-test distribution shift, leading to degraded performance. To address this, GLNN (Zhang et al. 2022) distills knowledge from a GNN teacher to an MLP student. SA-MLP (Chen et al. 2024) employs structure-aware MLPs with structure-mixing distillation. SGKD (He and Ma 2022) transfers GNN logits to MLPs using feature propagation. SimMLP (Wang et al. 2024b) applies self-supervised alignment between GNN and MLP.

For missing modalities, neighbor-based imputation methods like NeighMean (Malitesta et al. 2024b) and FeatProp (Malitesta et al. 2024a) are inapplicable under isolation. MUSE (Wu et al. 2024) treats modalities as pseudo neighbors and applies modality dropping. GMD (Wang et al. 2024a) removes conflicting gradients to reduce modality co-dependence. SiBraR (Ganhör et al. 2024) enforces modality consistency between shared encoders.

Graph Transformers

Graph Transformers (GTs) adapt Transformers for graphs and exhibit strong performance for node classification. SGFormer (Wu et al. 2023) and Polynormer (Deng, Yue, and Zhang 2024) both use GNNs for local and Transformers for global encoding, but differ in integration. NAGphormer (Chen et al. 2023) precomputes neighbor features as tokens for Transformers on single-modality graphs, enabling classification without online message passing.

Different from existing work, we address multimodal isolated cold-start node classification by enabling self-teaching within Graph Transformers, allowing the model to simultaneously handle structural isolation and missing modalities.

Problem Definition

We study isolated cold-start node classification on multimodal graphs. Formally, let the training graph be $\mathcal{G} = (\mathcal{V}, X^{(\mathrm{t})}, X^{(\mathrm{v})}, A, Y)$, where $\mathcal{V}$ is the set of nodes, and $N = |\mathcal{V}|$. $X^{(\mathrm{t})} \in \mathbb{R}^{N \times d_t}$ and $X^{(\mathrm{v})} \in \mathbb{R}^{N \times d_v}$ are the text and visual features of nodes, respectively, and $A \in \{0,1\}^{N \times N}$ is the adjacency matrix. The label vector $Y \in \mathbb{Z}^N$ assigns each node a label in $\{1, \ldots, C\}$ or -1 if unlabeled.

At test time, we are given a set of isolated cold-start nodes $\mathcal{V}_{\mathrm{te}}$. For simplicity, we still use $X^{(\mathrm{t})}$, $X^{(\mathrm{v})}$, Y to denote the text and visual features, and labels for the test nodes, respectively. Our goal is to predict their labels Y. However, some test nodes may suffer from missing modalities, meaning that the corresponding text or visual feature vector is absent.

Method

In this section, we present our NTSFormer framework.

Overall Framework

We propose **Neighbor-to-Self Graph Transformer (NTSFormer)**, a unified Graph Transformer framework specifically designed to address isolated cold-start node classification on multimodal graphs, handling both isolation (absence of edges) and modality missing (e.g., missing text or image features). It comprises three key modules (see Figure 4):

- **Multimodal Graph Pre-computation**: To capture diverse graph information for each node, this module performs a one-time pre-computation on multimodal graphs, converting multi-hop neighbor information across different modalities into fixed-length token sequences, which are then used as input to the Graph Transformer.
- **Mixture-of-Experts (MoE) Input Projection**: To effectively fuse the diverse graph information, this module adopts multiple experts to dynamically project the input tokens into a shared embedding space, capturing different aspects of modality and neighbor information.
- **Neighbor-to-Self Teaching via Cold-Start Masking**: To enable self-teaching of GTs for isolated cold-start node classification, this module applies a cold-start attention mask within a Transformer to simultaneously produce a *student prediction* (from self-features only) and a *teacher prediction* (with neighbor information), allowing our GT to supervise itself without degrading to an MLP.

Multimodal Graph Pre-computation

To enable NTSFormer to capture rich information in multimodal graphs, we perform a one-time multimodal graph pre-computation (Frasca et al. 2020) to convert neighbor information across different hops and modalities into token sequences suitable for Transformers. Given node features—text $X^{(\mathrm{t})} \in \mathbb{R}^{N \times d_t}$ and visual $X^{(\mathrm{v})} \in \mathbb{R}^{N \times d_v}$—we first align feature dimensions by zero-padding each modality to $d_{in} = \max(d_t, d_v)$. We then collect multimodal neighbor information within K hops by computing $\{\hat{A}^k X^{(\mathrm{t})} | k = 1, \ldots, K\}$ for text modality and $\{\hat{A}^k X^{(\mathrm{v})} | k = 1, \ldots, K\}$ for visual modality, where $\hat{A} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$ is the symmetrically normalized adjacency (Kipf and Welling 2017) with $\tilde{A} = A + I$ and $\tilde{D}_{ii} = \sum_{j=1}^{N} \tilde{A}_{ij}$. Here, $\hat{A}^k X^{(\mathrm{t})} \in \mathbb{R}^{N \times d_{in}}$ and $\hat{A}^k X^{(\mathrm{v})} \in \mathbb{R}^{N \times d_{in}}$ represent text and visual information collected from k-hop neighbors, respectively.

Input Tokens Construction. We arrange the collected multimodal neighbor information into an ordered input token sequence matrix for NTSFormer. We define the self-information tokens (self tokens) as:

$$\mathcal{X}_{\text{self}} = [X^{(\mathrm{t})} \text{ or } \langle \mathrm{MISS} \rangle,\ X^{(\mathrm{v})} \text{ or } \langle \mathrm{MISS} \rangle,\ \langle \mathrm{CLS}_S \rangle] \quad (1)$$

If a modality is missing at inference time, its position is replaced with a learned placeholder token $\langle \mathrm{MISS} \rangle$. During training, we simulate such cases by randomly replacing $X^{(\mathrm{t})}$ or $X^{(\mathrm{v})}$ with $\langle \mathrm{MISS} \rangle$ with probability p_{miss}. The $\langle \mathrm{CLS}_S \rangle$ token is a d-dimensional learnable vector used by the student for classification. The neighbor tokens are defined as:

$$\mathcal{X}_{\text{nbr}} = [\hat{A}X^{(\mathrm{t})},\ \hat{A}X^{(\mathrm{v})},\ \ldots,\ \hat{A}^K X^{(\mathrm{t})},\ \hat{A}^K X^{(\mathrm{v})},\ \langle \mathrm{CLS}_T \rangle] \quad (2)$$

where $\langle \mathrm{CLS}_T \rangle$ is another d-dimensional learnable token used by the teacher for classification.

The full input token sequence matrix is constructed as $S = \mathcal{X}_{\text{self}} \oplus \mathcal{X}_{\text{nbr}}$, where $\oplus$ denotes sequence-wise concatenation along the token dimension. The length of input token sequence per node is $L = 2K + 4$, comprising $2(K + 1)$ modality tokens and two classification tokens. In this paper, sequences are 1-indexed (e.g., $S[1]$ is the first item), and $S[-i]$ denotes the i-th item from the end.

This pre-computation is performed once on CPU, requires no gradient computation, and enables efficient Transformer training with fixed-length token sequences and standard mini-batching.

MoE Input Projection

Input projection layers are common in Transformers (Dosovitskiy et al. 2021), which map raw input feature tokens into a unified embedding space. The typical choice—using a shared MLP uniformly for all tokens—however, is limited in our setting, since the input tokens come from various sources (e.g., self/neighbor, modality, or special tokens).

To address this, we design an input projection module based on MoE techniques (Dai et al. 2024; Fedus, Zoph, and Shazeer 2022) that improves the model's capacity to handle diverse data by dynamically routing each token to specialized expert networks. A gating network is used to determine

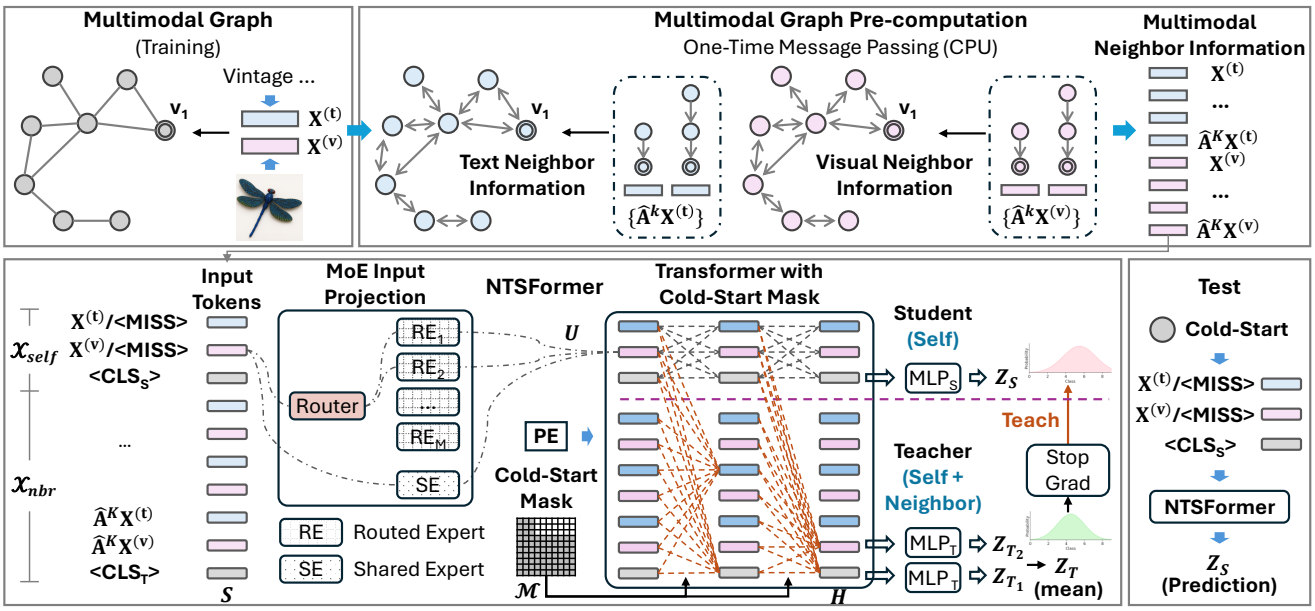


Figure 4: Overall framework of NTSFormer.

how tokens are assigned to different experts, including multiple routed experts and one shared expert.

Our gating network considers token positions, since they indicate tokens' semantic roles (e.g., self/neighbor, modality, or special token). For the i-th token in the input sequence across all N nodes, $S[i] \in \mathbb{R}^{N \times d_{in}}$, we concatenate a one-hot position vector $e_i \in \{0,1\}^L$ with only the i-th element equal to 1. The combined input is $\tilde{S}[i] = [S[i] \,\|\, \mathbf{1}_N e_i^\top] \in \mathbb{R}^{N \times (d_{\text{in}}+L)}$, and the gating scores are computed as:

$$\gamma = \text{softmax}\left(\tilde{S}[i] \cdot W_{\text{gate}}\right) \in \mathbb{R}^{N \times M}, \tag{3}$$

where $W_{\text{gate}} \in \mathbb{R}^{(d_{\text{in}}+L) \times M}$ is a learnable weight matrix. The result γ contains the normalized routing scores over the M experts for each node at token position i.

Each routed expert is implemented as a 2-layer MLP, denoted $\text{MLP}_{\text{RE}_m} : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^d$, where d is the hidden dimension of the Transformer. For each token position i, we select the top-$\hat{k}$ experts per node, indicated by a binary mask $\mathcal{T}(S[i]) = \text{TopK}_{\text{row-wise}}(\gamma, \hat{k}) \in \{0,1\}^{N \times M}$, where $\mathcal{T}(S[i])_{j,m} = 1$ indicates expert m is selected for node j. The routed expert output for token position i and node j is:

$$S'_{\text{RE}}[i]_j = \sum_{m=1}^{M} \mathcal{T}(S[i])_{j,m} \cdot \gamma_{j,m} \cdot \text{LN}\left(\text{MLP}_{\text{RE}_m}(S[i]_j)\right), \tag{4}$$

where $\gamma_{j,m} \in \mathbb{R}$ is the gating score, and LN is layer normalization. The full output matrix $S'_{\text{RE}}[i] \in \mathbb{R}^{N \times d}$ contains the routed representations for all N nodes at token position i.

We also include a shared expert $\text{MLP}_{\text{SE}} : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^d$, and the final projected output for token position i and node j is:

$$S'[i]_j = S'_{\text{RE}}[i]_j + \text{LN}\left(\text{MLP}_{\text{SE}}(S[i]_j)\right). \tag{5}$$

Applying this projection across all token positions yields the MoE-projected sequence: $U = \text{MoE}(S) \in \mathbb{R}^{N \times L \times d}$, where the projected tokens are combined into a 3D tensor over nodes, sequence length, and embedding dimension. This tensor serves as the input to the Transformer encoder.

MoE Regularization. To promote balanced expert usage in the MoE input projection layer, we include a load balancing loss (Fedus, Zoph, and Shazeer 2022) $\mathcal{L}_{\text{MoE}} = \sum_{m=1}^{M} P_m \cdot f_m$, where P_m is the average gate score and f_m is the fraction of tokens routed to expert m.

Neighbor-to-Self Teaching via Cold-Start Masking

To enable self-teaching within a unified Transformer, we design a mechanism that produces two distinct predictions: (1) The output token at $\langle \text{CLS}_S \rangle$ serves as the *student* prediction, based solely on self information, and is directly compatible with isolated cold-start inference scenarios. (2) The output token at $\langle \text{CLS}_T \rangle$ serves as the *teacher* prediction, incorporating both self and neighbor information for supervision during training. The teacher learns from full graph context, while the student mimics it using only self input.

Using standard self-attention (Vaswani et al. 2017) allows all tokens to attend to each other, causing neighbor information to leak into $\langle \text{CLS}_S \rangle$ and breaking the cold-start isolation assumption. To address this, we introduce a cold-start attention mask $\mathcal{M} \in \{0,1\}^{L \times L}$ as follows to separate student and teacher contexts during attention:

$$\mathcal{M} = \begin{pmatrix} \mathcal{M}^{(s\to s)} & \mathcal{M}^{(s\to n)} \\ \mathcal{M}^{(n\to s)} & \mathcal{M}^{(n\to n)} \end{pmatrix} = \begin{pmatrix} \mathbf{1}^{3\times 3} & \mathbf{0}^{3\times(L-3)} \\ \mathbf{1}^{(L-3)\times 3} & \mathbf{1}^{(L-3)\times(L-3)} \end{pmatrix} \tag{6}$$

where $\mathcal{M}_{ij} = 0/1$ disables/allows attention from token i to j. Here, 3 corresponds to the length of $\mathcal{X}_{\text{self}}$, which contains self information (including $\langle \text{CLS}_S \rangle$) and occupies the first three positions of the input token sequence. Specifically, the block $\mathcal{M}^{(s\to n)}$ gates attention from **s**elf tokens to **n**eighbor tokens, which are placed after position 3. $\mathcal{M}$ therefore ensures that self tokens can only attend to each other and are

strictly blocked from accessing neighbor tokens. As a result, the representation at $\langle \mathrm{CLS}_S \rangle$ remains purely self-based and suitable for isolated cold-start prediction, while $\langle \mathrm{CLS}_T \rangle$ incorporates full context and serves as the teacher for self-supervision during training.

With the cold-start mask, we apply $L^{(\mathrm{tf})}$ self-attention Transformer layers after the MoE input projection. Each layer updates the hidden states as follows:

$$H'^{(\ell)} = \mathrm{LN}\left(\mathrm{MHA}(H^{(\ell-1)}; \mathcal{M}) + H^{(\ell-1)}\right) \quad (7)$$

$$H^{(\ell)} = \mathrm{LN}\left(\mathrm{FFN}(H'^{(\ell)}) + H'^{(\ell)}\right) \quad (8)$$

where $H^{(1)} = U + \mathrm{PE}$ is the input to the first layer, and $\mathrm{PE} \in \mathbb{R}^{L \times d}$ is a learnable positional encoding. The Multi-Head Attention (MHA) module computes attention as:

$$\mathrm{MHA}(X; \mathcal{M}) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h) W^{(O)}, \quad (9)$$

$$\mathrm{head}_i = \mathrm{Att}\left(XW^{(Q)}_{(i)}, XW^{(K)}_{(i)}, XW^{(V)}_{(i)}\right), \quad (10)$$

$$\mathrm{Att}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}} + (1 - \mathcal{M})(-\infty)\right) V. \quad (11)$$

Here, $W^{(Q)}_{(i)}, W^{(K)}_{(i)}, W^{(V)}_{(i)} \in \mathbb{R}^{d \times d_h}$ and $W^{(O)} \in \mathbb{R}^{hd_h \times d}$ are parameters. The Feed-Forward Network (FFN) is a 2-layer MLP with GELU activation, applied independently to each token position with input/output dimension d.

Prediction and Self-Teaching. The output of the last Transformer, $H \in \mathbb{R}^{N \times L \times d}$, contains representations for all L tokens. The student prediction is produced from $\langle \mathrm{CLS}_S \rangle$ (token index: 3) via a student classifier $\mathrm{MLP}_S : \mathbb{R}^d \to \mathbb{R}^C$:

$$Z_S = \mathrm{softmax}(\mathrm{MLP}_S(H[:, 3])), \quad Z_S \in \mathbb{R}^{N \times C}. \quad (12)$$

Similarly, we use a teacher classifier $\mathrm{MLP}_T : \mathbb{R}^d \to \mathbb{R}^C$ for teacher predictions. By default, we use the last token $\langle \mathrm{CLS}_T \rangle$ (index: -1):

$$Z_{T_1} = \mathrm{softmax}(\mathrm{MLP}_T(H[:, -1])), \quad Z_{T_1} \in \mathbb{R}^{N \times C}. \quad (13)$$

To provide more stable supervision, we incorporate an additional teacher signal using the second-to-last token (index: -2), which also attends to neighbor information:

$$Z_{T_2} = \mathrm{softmax}(\mathrm{MLP}_T(H[:, -2])), \quad Z_{T_2} \in \mathbb{R}^{N \times C}. \quad (14)$$

The final teacher output is $Z_T = \frac{1}{2}(Z_{T_1} + Z_{T_2}) \in \mathbb{R}^{N \times C}$.

The student output Z_S is trained to match the gradient-stopped teacher output $\mathrm{stopgrad}(Z_T)$ using a self-teaching loss $\mathcal{L}_{\mathrm{ST}}$ based on the Kullback–Leibler (KL) divergence:

$$\mathcal{L}_{\mathrm{ST}} = \mathrm{KL}(\mathrm{stopgrad}(Z_T) \| Z_S). \quad (15)$$

Optimization and Inference

Training Objective. We use the standard cross-entropy loss for teacher prediction: $\mathcal{L}_{\mathrm{CE}} = \mathrm{CrossEntropy}(Y, Z_T)$. This is combined with the self-teaching loss and MoE regularization to form the final training objective:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \alpha \mathcal{L}_{\mathrm{ST}} + \beta \mathcal{L}_{\mathrm{MoE}}, \quad (16)$$

where α and β are hyperparameters. The entire model is optimized end-to-end using the AdamW optimizer.

Isolated Cold-Start Inference. At test time, NTSFormer takes in the self-only input sequence $\mathcal{X}_{\text{self}}$, where any missing modality is replaced with $\langle \mathrm{MISS} \rangle$. We take the student prediction from the output corresponding to $\langle \mathrm{CLS}_S \rangle$:

$$Z_S = \mathrm{softmax}(\mathrm{MLP}_S(\mathrm{NTSFormer}(\mathcal{X}_{\text{self}})[:, 3])). \quad (17)$$

Dataset	Nodes	Edges	Classes
Movies	16,672	218,390	20
Ele-fashion	97,766	199,602	12
Goodreads-NC	685,294	7,235,084	11

Table 1: Statistics of datasets.

Complexity Analysis

For pre-computation, which is only required for training, the time complexity is $O(KEd_{in})$, where E is the number of edges. For each node, the time complexities of MoE Input Projection, Multi-Head Attention, Feed-Forward Network, and prediction head are $O(KM(d_{in}d + d^2))$, $O(K^2d + Kd^2)$, $O(Kd^2)$, and $O(d^2 + dC)$, respectively.

Experiments

We conduct experiments on public datasets. All experiments are performed on a Linux system using a single NVIDIA RTX 3090 GPU (24GB), an Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz, and 376 GB of RAM.

Datasets

We use **Movies** from the MAGB benchmark (Yan et al. 2025), and **Ele-fashion** (abbreviated as **EF**) and **Goodreads-NC** (abbreviated as **GR-NC**) from the MM-Graph benchmark (Zhu et al. 2024), as shown in Table 1.

Movies is an Amazon movie-product e-commerce graph. Nodes are items with title/description text and cover images; edges indicate co-click or co-purchase interactions; labels are Amazon categories. Text features use RoBERTa (Liu et al. 2019); image features use CLIP-ViT (Radford et al. 2021). **Ele-fashion** is an Amazon fashion-product graph. Nodes contain item-title text and product images; edges capture co-purchase relations; labels are fashion product categories. Text and image features are extracted with T5-Base (Raffel et al. 2020) and ViT-Base (Dosovitskiy et al. 2021), respectively. **Goodreads-NC** is a large-scale graph derived from the Goodreads reading platform. Nodes are books with description text and cover images; edges link books with similar user preferences; labels are book genres (e.g., History, Children/Comics). Text and image features are extracted with T5-Base and ViT-Base, respectively.

Baselines

The baselines include MLP; a typical GNN, GraphSAGE (Hamilton, Ying, and Leskovec 2017); multimodal GNNs, including MMGCN (Wei et al. 2019), MGAT (Tao et al. 2020), and MIG (Hu et al. 2024); GTs, including Polynormer (Deng, Yue, and Zhang 2024), SGFormer (Wu et al. 2023), and NAGphormer (Chen et al. 2023); models for isolated node classification, including GLNN (Zhang et al. 2022), SGKD (He and Ma 2022), and SimMLP (Wang et al. 2024b); and models for modality-missing, including SiBraR (Ganhör et al. 2024) and MUSE (Wu et al. 2024).

Experiment Settings and Evaluation Metrics

Dataset Partitioning. To model the isolated cold-start scenario, we partition each dataset into four subsets: 20% la-

	Movies				Ele-fashion				Goodreads-NC			
	Text-Miss	Visual-Miss	No-Miss	All	Text-Miss	Visual-Miss	No-Miss	All	Text-Miss	Visual-Miss	No-Miss	All
MLP	41.15±1.33	37.63±1.23	46.76±2.60	41.85±1.34	69.72±3.61	71.67±1.51	84.04±0.51	75.15±0.71	40.25±0.83	58.16±1.01	68.27±0.54	55.56±0.22
GraphSAGE	38.60±3.04	37.05±1.41	41.40±2.46	39.02±2.02	74.68±1.23	46.57±0.47	78.42±0.41	66.56±0.23	36.25±2.45	34.47±3.35	54.23±0.92	41.65±1.01
MMGCN	40.68±2.66	34.82±2.46	46.80±1.64	40.77±0.41	69.01±5.20	56.50±4.06	80.05±2.27	68.52±0.87	22.55±7.97	49.59±1.02	58.92±0.91	43.69±3.14
MGAT	39.93±0.94	36.94±2.91	47.01±1.69	41.30±1.12	65.68±4.26	69.74±1.43	77.12±4.54	70.85±1.09	35.55±1.55	50.84±2.91	58.94±0.66	48.44±1.33
MIG	39.57±2.53	36.91±2.72	46.04±2.64	40.84±1.32	65.94±2.82	70.19±5.07	81.90±1.66	72.67±1.85	37.65±2.64	46.35±5.11	59.51±1.84	47.84±2.37
Polynormer	36.47±1.68	36.55±2.63	46.19±2.24	39.74±1.81	77.08±0.74	48.29±1.64	78.29±0.69	67.89±0.76	34.73±5.96	31.37±6.81	50.25±4.53	38.79±1.61
SGFormer	41.23±1.48	39.28±3.10	46.22±2.06	42.25±0.60	77.62±0.88	49.69±1.46	79.94±0.74	69.08±0.60	33.21±2.73	36.13±1.60	56.20±1.49	41.85±1.07
NAGphormer	39.03±2.25	39.14±1.39	44.50±2.52	40.89±1.48	71.08±0.99	70.90±2.58	82.41±0.36	74.79±0.87	35.74±1.00	46.66±0.97	56.95±0.57	46.45±0.53
GLNN	41.26±1.58	39.25±1.95	48.60±1.18	43.04±0.85	67.11±3.47	71.53±6.25	84.60±0.55	74.41±1.53	35.88±1.23	57.44±0.60	68.71±0.48	54.01±0.33
SGKD	42.81±1.68	37.59±1.65	46.76±1.27	42.39±0.77	72.73±2.61	73.23±2.76	84.33±0.51	76.76±1.09	40.19±0.89	58.26±0.63	68.50±0.54	55.65±0.38
SimMLP	42.88±1.04	39.10±2.17	48.38±2.23	43.45±0.88	67.91±5.39	72.76±2.28	84.40±0.79	75.02±0.89	37.26±0.80	58.75±0.49	68.74±0.71	54.92±0.28
SiBraR	36.55±1.90	40.51±1.94	44.17±1.73	40.41±0.79	70.51±3.90	47.19±5.54	73.27±3.78	63.65±2.97	28.59±5.17	31.55±2.84	34.70±4.16	31.62±0.92
MUSE	42.34±1.75	39.82±1.40	48.16±1.56	43.44±1.13	77.43±0.84	80.53±0.50	84.01±0.33	80.66±0.34	34.48±1.00	51.67±1.15	59.34±0.75	48.49±0.59
NTSFormer	**45.07±1.12**	**42.52±1.41**	**50.79±1.89**	**46.12±0.58**	**80.71±0.89**	**83.98±0.80**	**85.42±0.54**	**83.37±0.46**	**50.99±0.13**	**63.93±0.36**	**69.83±0.47**	**61.58±0.16**

Table 2: Performance Comparison on Multimodal Isolated Cold-Start Node Classification (Accuracy (%)).

beled training nodes, 60% unlabeled training nodes, 10% validation nodes (isolated cold-start), and 10% test nodes (isolated cold-start). The unlabeled training nodes are used in the training graph for message passing but receive no supervision. For validation and test nodes, all edges involving these nodes are removed to ensure they remain completely isolated, thereby creating an isolated cold-start environment.

Missing-Modality Setting. We further evaluate performance on isolated cold-start nodes under a modality-missing scenario. Specifically, while all training nodes have complete modality information, the validation/test set is divided into three disjoint subsets of equal size: one missing text, one missing visual features, and one with full multimodal input. That is, each subset contains approximately 33.3% of the validation/test nodes. We denote these three test conditions as **Text-Miss**, **Visual-Miss**, and **No-Miss**, and additionally report results on **All**, which combines all test nodes.

Parameter Setting. We apply unified hyperparameter settings across NTSFormer and all baselines to ensure fair comparison. In particular, the hidden dimension is fixed to 512, the maximum hop size is set to $K = 2$, and the input and hidden dropout rates are 0.2 and up to 0.5, respectively. All models are optimized using AdamW with learning rate 2×10^{-3} and weight decay 1×10^{-2}. Training is run for 300 epochs on *Movies* and *Ele-fashion*, and 50 epochs on *Goodreads-NC*. Early stopping is applied based on validation accuracy, and the best-performing checkpoint is used for final testing. Baselines are trained using their official implementations when available. For NTSFormer, we employ additional method-specific settings. Specifically, NTSFormer uses $L^{(\mathrm{tf})} = 2$ Transformer layers with 2 attention heads, and the MoE input projection module comprising $M = 6$ routed experts along with 1 shared expert. The self-teaching and MoE regularization weights are $\alpha = 1.0$ and $\beta = 0.1$, respectively. These hyperparameters were selected based on a light random search over the validation set.

Evaluation Metrics. We report classification accuracy as the evaluation metric. All results are averaged over 5 runs with random seeds from 1 to 5, controlling for variation in both dataset splitting and model initialization. Importantly, with a fixed seed, all methods are evaluated on identical train/validation/test splits and identical modality-missing settings, ensuring fair comparison. We report mean accuracy and standard deviation for each of the four settings: Text-Miss, Visual-Miss, No-Miss, and All.

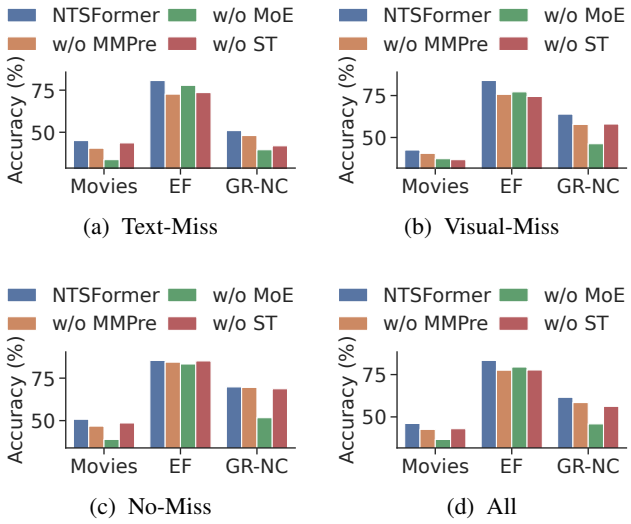
Performance Analysis

Table 2 compares the performance of baselines and NTSFormer, from which we draw the following observations:

- General (multimodal) GNNs—GraphSAGE, MMGCN, MGAT, and MIG—do not perform well on this task and even underperform MLPs, verifying that isolated cold-start is a severe issue for GNNs on multimodal graphs.
- Graph Transformers, including Polynormer, SGFormer, and NAGphormer, perform similarly to general GNNs and multimodal GNNs and do not perform well under the isolated cold-start setting.
- GLNN, SGKD, and SimMLP are GNNs considering structural isolation, and outperform General (multimodal) GNNs and GTs. Their MLP students taught by GNNs outperform the original MLP baseline.
- SiBraR and MUSE are GNNs considering modality missing. MUSE outperforms all general (multimodal) GNNs and GTs, and even GNNs for structural isolation on two datasets, showing that modality missing is also an important issue on this task. SiBraR does not perform well, potentially due to its shared encoder design, which cannot handle each modality differently.
- NTSFormer tackles isolation and modality-missing issues via a self-teaching paradigm, consistently outperforming all baselines across all datasets by a large margin, showing the advantage of self-teaching GTs on this task. Compared to teacher-student baselines, we can directly perform self-teaching via end-to-end training, without requiring complicated two-step operations.

Detailed Analysis

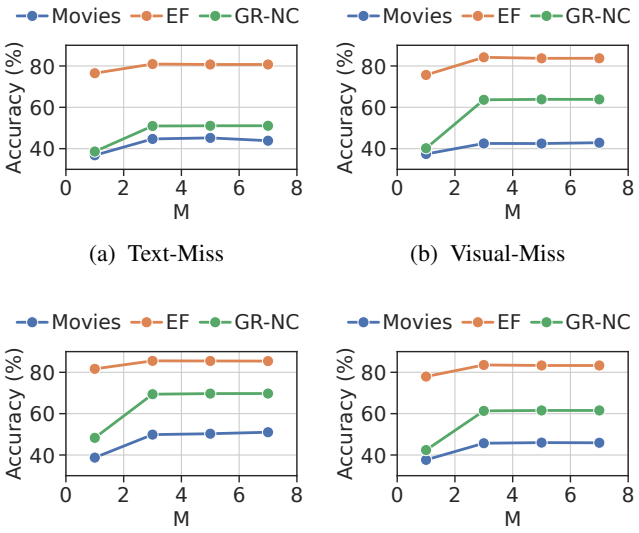
Impact of Multimodal Graph Pre-computation Our pre-computation collects text and visual neighbor information into separate tokens. The variant, NTSFormer w/o MMPre, replaces it with typical pre-computation. Specifically, it concatenates text and visual features as node input, and the resulting tokens do not distinguish modalities. Figure 5 shows that NTSFormer consistently outperforms NTSFormer w/o MMPre, verifying our method's effectiveness.



(a) Text-Miss (b) Visual-Miss

(c) No-Miss (d) All

Figure 5: Ablation results of **NTSFormer** and its ablated variants (w/o MMPre, w/o MoE, w/o ST (SelfTeach)). The first three subfigures report performance on different subsets of test nodes, each with a specific modality-missing setting: (a) Text-Miss, (b) Visual-Miss, and (c) No-Miss. The last subfigure (d) reports performance on all test nodes.



(a) Text-Miss (b) Visual-Miss

(c) No-Miss (d) All

Figure 6: Impact of number of routed experts (M).

$L^{(\text{tf})}$	Movies	Ele-fashion	Goodreads-NC
1	45.22	83.54	61.36
2	**46.12**	83.37	61.58
3	45.27	83.51	**61.68**
4	45.32	**83.67**	61.51

Table 3: Impact of number of Transformer layers ($L^{(\text{tf})}$).

Method	Movies	Ele-fashion	Goodreads-NC
MIG	82s	**96s**	387s
GLNN	65s	161s	778s
MUSE	154s	406s	1437s
NTSFormer	**46s**	226s	**260s**

Table 4: Training efficiency comparison (seconds).

Impact of MoE Input Projection We introduce a variant **NTSFormer w/o MoE**, where the MoE is replaced with a standard shared linear projection, following the design used in NAGphormer (Chen et al. 2023). As shown in Figure 5, the variant consistently underperforms NTSFormer, verifying MoE's effectiveness. Furthermore, Figure 6 illustrates the effect of varying the number of routed experts, M. The results indicate that performance degrades when $M < 3$, underscoring the importance of employing multiple experts for effectively encoding diverse multimodal graph information.

Impact of Self-Teaching Mechanism We replace the self-teaching mechanism with the MLP-student's two-branch setup, similar to GLNN (Zhang et al. 2022). In this variant, **NTSFormer w/o ST (SelfTeach)**, the student is implemented as a separate MLP trained independently to match the teacher branch output. As shown in Figure 5, this leads to a noticeable drop in performance, showing the superiority of self-teaching GTs over the MLP-student method.

Impact of Number of Transformer Layers As shown in Table 3, using multiple layers generally provides a slight performance gain compared to a single layer. Although the best-performing layer depth varies per dataset—2 layers for Movies, 4 layers for Ele-fashion, and 3 layers for Goodreads-NC—none of the highest results occur at a single-layer configuration, suggesting that deeper architectures can offer marginal improvements.

Efficiency of Training

We analyze the training efficiency across datasets of varying sizes. Table 4 shows NTSFormer maintains comparable efficiency on smaller datasets (Movies and Ele-fashion) but exhibits advantages on the larger Goodreads-NC dataset. This improvement stems from our one-time pre-computation that converts graph structures into regular-shaped tensors, avoiding costly repetitive message passing during training (Hu, Hooi, and He 2024). This enables better scalability on large graphs while achieving competitive performance.

Conclusion

We study isolated cold-start node classification on multimodal graphs, where such nodes have no edges and often have missing modalities. Existing methods degrade graph models to MLPs for cold-start inference, limiting capacity. We propose the **Neighbor-to-Self Graph Transformer (NTSFormer)**, which addresses isolation and missing-modality issues via self-teaching without degrading to an MLP. Experiments on public benchmarks show the superiority of NTSFormer over various baselines, underscoring the effectiveness of self-teaching Graph Transformers on this task. Future work includes adapting it to dynamic graphs.

Acknowledgments

This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative, and the Ministry of Education, Singapore, under the Academic Research Fund Tier 2 (FY2025) (Grant MOE-T2EP20124-0009). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority.

References

Cai, J.; Wang, X.; Li, H.; Zhang, Z.; and Zhu, W. 2024. Multimodal Graph Neural Architecture Search under Distribution Shifts. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 8227–8235. AAAI Press.

Chen, J.; Bai, M.; Chen, S.; Gao, J.; Zhang, J.; and Pu, J. 2024. SA-MLP: Distilling Graph Knowledge from GNNs into Structure-Aware MLP. *Transactions on Machine Learning Research*.

Chen, J.; Gao, K.; Li, G.; and He, K. 2023. NAGphormer: A Tokenized Graph Transformer for Node Classification in Large Graphs. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Dai, D.; Deng, C.; Zhao, C.; Xu, R. X.; Gao, H.; Chen, D.; Li, J.; Zeng, W.; Yu, X.; Wu, Y.; Xie, Z.; Li, Y. K.; Huang, P.; Luo, F.; Ruan, C.; Sui, Z.; and Liang, W. 2024. DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models. arXiv:2401.06066.

Deng, C.; Yue, Z.; and Zhang, Z. 2024. Polynormer: Polynomial-Expressive Graph Transformer in Linear Time. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Fedus, W.; Zoph, B.; and Shazeer, N. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120): 1–39.

Frasca, F.; Rossi, E.; Eynard, D.; Chamberlain, B.; Bronstein, M.; and Monti, F. 2020. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198*.

Ganhör, C.; Moscati, M.; Hausberger, A.; Nawaz, S.; and Schedl, M. 2024. A Multimodal Single-Branch Embedding Network for Recommendation in Cold-Start and Missing Modality Scenarios. In *Proceedings of the 18th ACM Conference on Recommender Systems, RecSys 2024, Bari, Italy, October 14-18, 2024*, 380–390. ACM.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 1024–1034.

He, Y.; and Ma, Y. 2022. SGKD: A Scalable and Effective Knowledge Distillation Framework for Graph Representation Learning. In *IEEE International Conference on Data Mining Workshops, ICDM 2022 - Workshops, Orlando, FL, USA, November 28 - Dec. 1, 2022*, 666–673. IEEE.

Hu, J.; Hooi, B.; and He, B. 2024. Efficient Heterogeneous Graph Learning via Random Projection. *IEEE Trans. Knowl. Data Eng.*, 36(12): 8093–8107.

Hu, J.; Hooi, B.; He, B.; and Wei, Y. 2024. Modality-Independent Graph Neural Networks with Global Transformers for Multimodal Recommendation. arXiv:2412.13994.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.

Malitesta, D.; Rossi, E.; Pomo, C.; Malliaros, F. D.; and Noia, T. D. 2024a. Dealing with Missing Modalities in Multimodal Recommendation: a Feature Propagation-based Approach. arXiv:2403.19841.

Malitesta, D.; Rossi, E.; Pomo, C.; Noia, T. D.; and Malliaros, F. D. 2024b. Do We Really Need to Drop Items with Missing Modalities in Multimodal Recommendation? In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, 3943–3948. ACM.

Moscati, M. 2024. Multimodal Representation Learning for High-Quality Recommendations in Cold-Start and Beyond-Accuracy. In *Proceedings of the 18th ACM Conference on Recommender Systems*, 1290–1295.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67.

Shen, M.; Wei, Y.; Yin, J.; Rajan, D.; Hu, D.; and See, S. 2024. Enhancing Modality Representation and Alignment

for Multimodal Cold-start Active Learning. In *Proceedings of the 6th ACM International Conference on Multimedia in Asia*, 1–8.

Tao, Z.; Wei, Y.; Wang, X.; He, X.; Huang, X.; and Chua, T. 2020. MGAT: Multimodal Graph Attention Network for Recommendation. *Inf. Process. Manag.*, 57(5): 102277.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Wang, H.; Luo, S.; Hu, G.; and Zhang, J. 2024a. Gradient-Guided Modality Decoupling for Missing-Modality Robustness. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 15483–15491. AAAI Press.

Wang, Z.; Zhang, Z.; Zhang, C.; and Ye, Y. 2024b. SimMLP: Training MLPs on Graphs without Supervision. arXiv:2402.08918.

Wei, Y.; Wang, X.; Nie, L.; He, X.; Hong, R.; and Chua, T. 2019. MMGCN: Multi-modal Graph Convolution Network for Personalized Recommendation of Micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, 1437–1445. ACM.

Wu, Q.; Zhao, W.; Yang, C.; Zhang, H.; Nie, F.; Jiang, H.; Bian, Y.; and Yan, J. 2023. Simplifying and Empowering Transformers for Large-Graph Representations. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Wu, Z.; Dadu, A.; Tustison, N. J.; Avants, B. B.; Nalls, M. A.; Sun, J.; and Faghri, F. 2024. Multimodal Patient Representation Learning with Missing Modalities and Labels. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yan, H.; Li, C.; Yin, J.; Yu, Z.; Han, W.; Li, M.; Zeng, Z.; Sun, H.; and Wang, S. 2025. When Graph meets Multimodal: Benchmarking and Meditating on Multimodal Attributed Graphs Learning. arXiv:2410.09132.

Zhang, L.; Zhang, X.; Zhou, Z.; Huang, F.; and Li, C. 2024. Reinforced adaptive knowledge learning for multimodal fake news detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 16777–16785.

Zhang, S.; Liu, Y.; Sun, Y.; and Shah, N. 2022. Graph-less Neural Networks: Teaching Old MLPs New Tricks Via Distillation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Zhu, J.; Zhou, Y.; Qian, S.; He, Z.; Zhao, T.; Shah, N.; and Koutra, D. 2024. Multimodal Graph Benchmark. arXiv:2406.16321.