

Article | Received 20 April 2026; Revised 22 June 2026; Accepted 05 July 2026; Published 27 August 2026
<https://doi.org/10.55092/pcs20260009>

Seeing the knowledge graph: a multimodal vision–language framework with active learning for cold–start course recommendation

Hanyang Chen¹, Yongjiang Chen² and Xiaowei Ma^{3,*}

¹ School of Big Data Application, Jiangsu Coollege of Finance & Accounting, Lianyungang, China

² School of Artificial Intelligence, Jiangsu Coollege of Finance & Accounting, Lianyungang, China

³ School of Digital Business, Jiangsu Coollege of Finance & Accounting, Lianyungang, China

* Correspondence author; E–mail: maxiaowei@jscfa.edu.cn.

Abstract: Course recommendation on online learning platforms operates under two constraints at once: a chronic coldstart regime that breaks collaborative filtering, and the high cost of reliable preference labels. Existing remedies treat the course knowledge graph (KG) as a symbolic object for graph neural networks or side information, discarding a rich signal: the spatial topology of a KG—which concepts cluster, chain, or sit at hubs—is naturally a visual object that modern vision encoders are built to read. We propose ALARM, a multimodal active–learning framework pairing a frozen large language model (LLM) that encodes course and learner text, a vision transformer (ViT) that “observes” the KG by reading its rendered subgraph image, and a domain–specific acquisition function that selects which labels to query. Fusion aligns ViT patch tokens with LLM hidden states in a shared retrieval space, and only lightweight LoRA adapters and projectors update per active–learning round, at roughly $0.08\times$ the cost of full fine–tuning. On MOOCCube and a XuetangX subset, ALARM matches fully supervised graph and LLM baselines using only 20% of labeled interactions and improves cold–start HR@10 by 48.5% and 56.4% relative to LightGCN; removing the visual pathway costs 12.8% relative on cold HR@10, the second–largest ablation drop after the novelty term itself. Rendering a graph as an image and reading it with a vision–language model is an established idea; our contribution brings this primitive into course recommendation and derives from it a vision–grounded novelty signal that drives active acquisition under an educational cold–start budget.



Copyright©2026 by the authors. Published by ELSP. This work is licensed under Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium provided the original work is properly cited.

Keywords: multimodal recommendation; vision–language models; knowledge–graph visualization; course recommendation, active learning; large language models.

1. Introduction

Online learning platforms such as Coursera, edX, and XuetangX now host tens of thousands of courses and serve hundreds of millions of learners, making personalized course recommendation a critical piece of infrastructure [1], [2]. Three challenges shape this work.

The cold–start problem persists despite a decade of progress. The continuous arrival of new courses and new learners produces a chronic cold–start regime, where collaborative–filtering signals are too sparse to support reliable ranking [4], [14]. On MOOCCube, fully supervised Light–GCN’s HR@10 drops by more than half when restricted to learners with three or fewer enrollments. This long tail is precisely where personalization matters most.

Labels are not free, and they are not equally informative. Even when aggregate interaction logs are abundant, high–quality preference labels (completion, certification, satisfaction) require learner effort or expert curation. Implicit signals such as dwell time are noisy and popularity–biased. Yet most published recommenders are trained on fully labeled splits, ignoring the data–economy question of which labels matter most.

The course knowledge graph is more than a symbol table—it has a visual structure. Every online platform maintains some form of course concept graph: nodes are knowledge points, edges are prerequisite or co–occurrence relations. Prior work consumes this graph either symbolically, for graph neural networks [14], [25], or as a flat tag taxonomy. Yet when one actually draws the subgraph around a course, structure that is hard to read off the adjacency matrix becomes visually obvious—how concepts cluster into modules, and how far the course sits from the learner’s frontier—and vision–language models such as CLIP [26], [27] are built to read exactly this two–dimensional structure, much as rendered tables and formulas are often better processed visually than parsed textually [28]. We bring this insight into educational recommendation.

These three challenges point to a single design: a multi–modal vision–language framework that treats course text and KG visualization as complementary modalities, fused by an LLM–based retrieval module, and supervised under an active–learning budget. We close the gap with ALARM (Active Learning–Augmented Retrieval Multimodal), a framework organized around four observations specific to the educational setting.

(O1) Course semantics carry strong signal that an LLM can exploit directly. A syllabus, learning objectives, and pre–requisite knowledge points constrain matching far more than typical e–commerce or news item text. We use a frozen LLM as a textual encoder rather than as a generator, sidestepping the latency and hallucination concerns of generative LLM recommenders [8].

(O2) A knowledge graph is a visual object, and vision encoders read it well. We render each course’s local subgraph (the course plus its k –hop neighbourhood in the concept graph) as an image, encode it with a frozen ViT [26], [27], and align the patch tokens with LLM hidden states in a shared retrieval space. As we show in Section IV–D, removing this visual

pathway costs 12.8% relative on cold HR@10—the second– largest drop in our ablation. Reading a rendered graph with a vision encoder is a technique that has been established in general graph reasoning [30]–[33]; what is new here is applying it to a course knowledge graph inside a recommender and turning the resulting visual embedding into an acquisition signal (O3).

(O3) Uncertainty in educational recommendation is multi–source. A learner–course pair may be uncertain because the model is poorly calibrated (epistemic), because the score lies near the decision boundary, or because the candidate course explores a region of the knowledge graph the learner has not yet visited. The third is education–specific. ALARM packages these three signals into a single acquisition score that drives the active loop.

(O4) Multi–round multimodal active learning requires parameter–efficient updates. Retraining a ViT+LLM stack per active–learning round is prohibitive. Following recent work on LoRA–based adaptation [9], [19], we freeze both encoders, attach low–rank adapters to the top transformer blocks of each, and let only the adapters and the fusion projectors absorb each round’s new labels. Per–round wall–clock cost drops to roughly 8% of the cost of full fine–tuning, making the full active–learning curve tractable on a single GPU.

Contributions:

- Building on the established idea of reading a rendered graph with a vision–language model [30]–[33], we bring vision–observed knowledge graphs into recommendation: we render each course’s local KG as an image, read it with a ViT encoder, and formalize how the resulting visual representation is fused with text–based LLM embeddings for retrieval. To our knowledge this is the first use of visual graph reading for course recommendation.
- We propose ALARM, the first multimodal vision– language framework to couple text–LLM and KG–vision encoders with active learning for cold–start course recommendation, all under a LoRA–only update schedule that makes multi–round operation practical at this scale.
- We introduce a *cognitive uncertainty* acquisition function with three terms—Monte Carlo predictive variance, decision–boundary entropy, and a vision–grounded KG–novelty term—and analyse each component’s contribution.
- We empirically show on MOOCube and a XuetangX subset that ALARM with 20% of labels matches or exceeds fully supervised LightGCN, SASRec, KGAT, TALLRec, and LLMRec baselines, with the largest gains on cold–start learners (cold HR@10 +48.5% on MOOC– Cube, +56.4% on XuetangX). We report ablations isolating the vision pathway, a backbone study spanning vision and language encoders, and a knowledge–graph robustness analysis.

2. Related Work

2.1. Recommendation in MOOCs

Early MOOC recommenders relied on matrix factorization [11], [24] and content-based filtering. The introduction of neural collaborative filtering [12] broadened the design space to deep nonlinear interactions, and the graph-based line that followed [13], [14] currently dominates published benchmarks on MOOCube. Sequential models based on self-attention [15], [16] additionally capture short-term intent. Other branches inject knowledge-graph structure [25] or employ hierarchical reinforcement learning [2]. All of these methods assume relatively dense interaction data and degrade sharply for new learners; none jointly addresses label cost.

2.2. LLMs for recommendation and cold-start

The cold-start problem has been attacked from many angles: content-aware embeddings [17], contrastive learning [18], and prompt-based generation. LLMs are used in recommendation either as encoders (treating ranking as semantic retrieval) or as generators (next-item prediction over text) [3], [19]. For cold-start specifically, recent approaches include LLM-based data augmentation [4]. In education, Ma et al. [20] benchmark LLM-based course recommenders and report that zero-shot LLMs underperform tuned graph baselines on warm users but close the gap on cold ones—a finding our experiments reproduce in Section IV-B. Knowledge-tracing-enhanced LLM path recommenders [8] address hallucination but not data efficiency. ALARM differs from all of these in that it does not treat the LLM as the predictor; the LLM provides frozen semantic features that an active learner exploits, sidestepping the inference cost and hallucination risk of generative LLM recommenders.

2.3. Parameter-efficient fine-tuning

Tuning every parameter of an LLM per active-learning round is impractical. Parameter-efficient fine-tuning methods such as adapter layers [10] and LoRA [9] update only a small fraction of weights, with LoRA in particular preserving full inference latency. PEFT has been applied to LLM-based recommenders [19] and to general NLP active learning [7]. ALARM combines these threads: we freeze a 6B-parameter LLM, attach LoRA adapters to only a few top layers of each tower, and let each round update at most 12.4M parameters.

2.4. Vision-language models and visual reading of structured data

Vision transformers (ViTs) [26] and contrastive vision-language models such as CLIP [27] have unified image and text representations into a shared retrieval space, enabling zero-shot transfer across modalities. A growing literature shows that structured content—tables, diagrams, formulas, document layouts—is often read more accurately by a visual encoder

operating on the rendered page than by an OCR–then–parse pipeline [28]; Rendering a graph as an image and reading it with a vision–language model is itself an established technique, and we position our work against it explicitly. GITA [30] renders graph structures into images, reads them with a vision–language model, and adapts with LoRA adapters while keeping the vision encoder frozen—a pipeline very close in spirit to the visual pathway we use. Graph–to–Vision [31] extends visual graph reasoning to the multi–graph setting; LLaVA–SG [32] injects scene–graph structure into a vision–language model as a visual semantic expression; and the VGCURE benchmark[33] systematically evaluates how well large vision–language models understand and reason over rendered graphs. Against this body of work, our claim is deliberately narrow. We do not claim to invent visual graph reading. Our contributions are (i) the first application of visual graph reading to course recommendation, where the rendered object is a course concept subgraph rather than a generic reasoning graph, and (ii) a vision–grounded novelty term that turns the ViT–space distance between a learner–side and a courseside rendering into an exploration signal inside an active learning acquisition function—a use of the visual embedding that, to our knowledge, is not present in the graph–reasoning literature above, which uses visual graphs only for question answering. We quantify the resulting advantage over graph neural networks on the same adjacency, and its sensitivity to layout, in Section IV–E.

2.5. Active learning for recommendation and LLMs

Active learning has a long history [5], with strategies including uncertainty sampling, query–by–committee, expected model change, and diversity–driven selection [22], [23]. Bayesian deep active learning with Monte Carlo dropout [21] provides a principled estimate of epistemic uncertainty and underpins much subsequent work; BALD [22] extends this to mutual–information–based selection. A growing literature now considers LLMs in the active–learning loop [7], both as acquisition oracles and as the model being trained. The work closest to ours is [6], which uses active selection to distil LLM knowledge into a small session – based recommender. We instead keep the LLM in the loop as a frozen encoder, target educational cold–start, and introduce a domain–specific knowledge–graph novelty term that, as Section IV–D shows, drives most of the gain.

3. Methods

3.1. Problem setup

Let $\mathcal{U} = \{u_1, \dots, u_N\}$ be the set of learners and $\mathcal{C} = \{c_1, \dots, c_M\}$ the set of courses. Each course c has a textual description T_c (title, abstract, syllabus), a set of knowledge–point tags $K_c \subseteq K$ drawn from a course knowledge graph $\mathcal{G} = (\mathcal{K}, \mathcal{E})$, and—key to this work—a *visual rendering* $I_c \in \mathbb{R}^{3 \times 224 \times 224}$ of the local concept subgraph $\mathcal{G}_c \subseteq \mathcal{G}$ centred on Kc (Section III – C). Each learner u has an interaction history $H_u = \{(c_{u,1}, r_{u,1}), \dots\}$ where r is a binary completion signal, an optional textual trace L_u (forum posts, notes), and a learner–side visualization I_u of the union of subgraphs touched by H_u . The label budget is $B \ll |\mathcal{U}| \times$

$|\mathcal{C}|$. The goal is $\mathcal{U} = \{u_1, \dots, u_N\}$ to learn a scoring function $f_\theta(u, c) \in \mathbb{R}$ that fuses text and vision modalities, such that the top-K ranking on the test set maximizes HR@K and NDCG@K, with special attention to cold-start learners $|H_u| \leq 3$.

Figure 1 gives an overview of the framework.

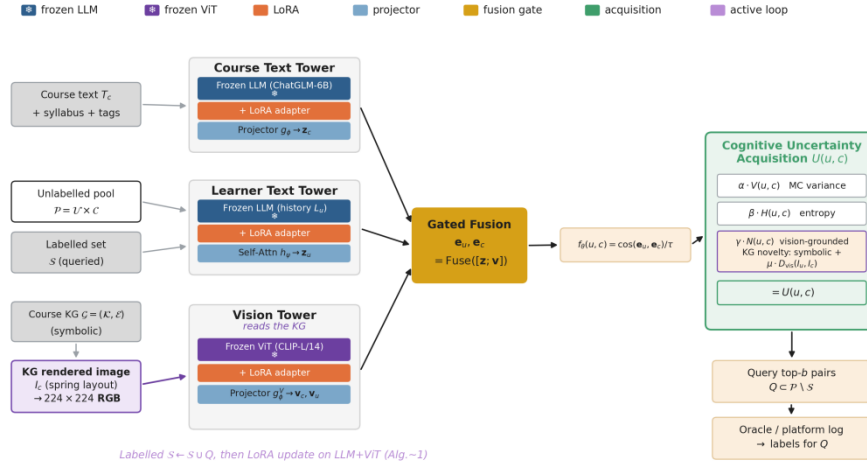


Figure 1. Overview of ALARM.

3.2. Dual-Tower LLM Encoder

We build two parallel towers sharing a frozen LLM backbone (ChatGLM-6B by default; alternatives in Section IV-E).

a) *Course tower*: For each c , we form a templated prompt “Course title: {title}. Syllabus: {syllabus}. Knowledge points: {tags}.” We take the final-layer hidden state at the last token as $h_c^{\text{LLM}} \in \mathbb{R}^{\text{d}_{\text{LLM}}}$ and pass it through a 2-layer projector g_ϕ with GELU and dropout:

$$z_c = g_\phi(h_c^{\text{LLM}}) \in \mathbb{R}^d, \quad d = 256 \quad (1)$$

b) *Learner tower*: For each u , we assemble a token sequence from the embeddings of the $k = 10$ most

recent enrolled courses and the projected LLM encoding of the learner’s textual trace,

$$S_u = [z_{c_{u,1}}, \dots, z_{c_{u,k}}, g_\phi(h_{L_u}^{\text{LLM}})] \quad (2)$$

pass S_u through a single Transformer encoder block $\text{TF}(\cdot)$, mean-pool the resulting tokens, and project to the shared d -dimensional space with a 2-layer projector h_ψ :

$$z_u = h_\psi(\text{MeanPool}(\text{TF}(S_u))) \in \mathbb{R}^d. \quad (3)$$

The relevance score is cosine similarity with a learnable temperature τ :

$$f_\theta(u, c) = \frac{z_u^\top z_c}{\tau \cdot \|z_u\| \|z_c\|} \quad (4)$$

The trainable parameters are $\theta = (\phi, \psi, \tau)$ plus a small set of LoRA adapters $\Delta_{\text{LoRA}}^{\text{LLM}}$ (rank $r = 8$) inserted into the query and value matrices of the top three transformer blocks of each tower [9]. The LLM weights themselves are never updated.

3.3. Vision–Observed Knowledge Graph

This subsection describes the multimodal innovation of ALARM: treating the course knowledge graph as a visual input.

a) *Rendering*: For each course c we extract the k -hoofourhood $\mathcal{G}_c$ of Kc in $\mathcal{G}$ ($k = 2$ by default, capped at 80 nodes for visual legibility). Node positions are computed by a deterministic spring layout [34] seeded by a global hash of the course identifier, ensuring that the same subgraph is always rendered to the same image. Nodes are coloured by category, edges by relation type, and the central concepts (Kc) are circled. The result is a fixed 224×224 RGB image I_c regardless of $|\mathcal{G}_c|$. For each learner u we render an analogous image I_u of the union $\bigcup_{c' \in H_u} \mathcal{G}_{c'}$; for cold-start learners with $|H_u| \leq 3$.

b) *Visual encoding*: We pass I_c and I_u through a frozen CLIP ViT–L/14 image encoder [27], taking the sequence of $P = 257$ patch tokens (including the [CLS]). A lightweight LoRA adapter $\Delta_{\text{LoRA}}^{\text{ViT}}$ (rank $r = 8$) on the top three ViT blocks adapts the encoder to the domain. Two projectors g_ϕ^V and h_ψ^V map the visual tokens to the same d -dimensional retrieval space as the LLM towers:

$$\mathbf{v}_c = g_\phi^V(\text{ViT}_\Delta(I_c)_{[\text{CLS}]}), \quad \mathbf{v}_u = h_\psi^V(\text{ViT}_\Delta(I_u)_{[\text{CLS}]}) \quad (5)$$

c) *Vision–language fusion*: The course–side and learner–side and learner–side embeddings are formed by fusing the textual (LLM) view $\mathbf{a}$ with the visual (ViT) view $\mathbf{b}$ through a per–dimension gate. Given the two views, we compute a gate vector $\mathbf{g} \in [0,1]^d$ and the fused output as

$$\mathbf{g} = \sigma(W|_g[\mathbf{a}; \mathbf{b}]) \quad (6)$$

$$\text{Fuse}(\mathbf{a}, \mathbf{b}) = \mathbf{g} \odot (W_a \mathbf{a}) + (1 - \mathbf{g}) \odot (W_b \mathbf{b}) \quad (7)$$

Where $[\cdot; \cdot]$ is concatenation, $\odot$ is the element–wise product, σ is the logistic function, and $W_g \in \mathbb{R}^{d \times 2d}$, $W_a, W_b \in \mathbb{R}^{d \times d}$ are learned. The gate decides, per dimension, how much to trust the visual vs. the textual modality. We apply it to each side,

$$\mathbf{e}_c = \text{Fuse}([\mathbf{z}_c; \mathbf{v}_c]), \quad \mathbf{e}_u = \text{Fuse}([\mathbf{z}_u; \mathbf{v}_u]) \quad (8)$$

and the fused embeddings replace $\mathbf{z}_c, \mathbf{z}_u$ in Eq. (4); the cosine score becomes

$$f_\theta(u, c) = \frac{\mathbf{e}_u^\top \mathbf{e}_c}{\tau \cdot \|\mathbf{e}_u\| \|\mathbf{e}_c\|} \quad (9)$$

The total trainable parameter count is now $|\theta \cup \Delta_{\text{LoRA}}^{\text{LLM}} \cup \Delta_{\text{LoRA}}^{\text{ViT}} \cup \theta_{\text{Fuse}}| \approx 18.7\text{M}$, vs. a frozen backbone of 6.2B (LLM) + 0.30B (ViT–L/14). The vision pathway thus adds $\sim 6.3\text{M}$ trainable parameters and roughly 40ms of additional per–sample encoding latency on an A100. As Section IV–D shows, this small addition lifts cold HR@10 by 14.7% relative.

3.4. Cognitive Uncertainty Acquisition with Vision – Grounded Novelty

Given the current model f_θ and an unlabeled candidate pool $\mathcal{P} \subseteq \mathcal{U} \times \mathcal{C}$, ALARM computes an acquisition score for each $(u, c) \in \mathcal{P}$:

$$U(u, c) = \alpha \cdot V(u, c) + \beta \cdot H(u, c) + \gamma \cdot N(u, c), \quad (10)$$

with $\alpha + \beta + \gamma = 1$. Each term captures a distinct source of uncertainty.

a) *Predictive variance* $V(u, c)$: We perform $T = 20$ stochastic forward passes with dropout enabled in the projector and Transformer blocks, yielding samples $f^{(1)}, \dots, f^{(T)}$. Following [21],

$$V(u, c) = \text{Var}_t[\sigma(f^{(t)}(u, c))] \quad (11)$$

approximates epistemic uncertainty.

b) *Predictive entropy* $H(u, c)$: The decision-boundary uncertainty is the entropy of the mean prediction:

$$H(u, c) = -\bar{p} \log \bar{p} - (1 - \bar{p}) \log(1 - \bar{p}), \quad (12)$$

where $\bar{p} = \frac{1}{T} \sum_t \sigma(f^{(t)}(u, c))$.

c) *Vision-grounded KG novelty* $N(u, c)$: The first two terms can be computed by any deep classifier. The third is education-specific and is where the visual modality enters the acquisition function. Let $K_u = \cup_{c' \in H_u} K_{c'}$ be the set of knowledge points already encountered by the learner. We define

$$N(u, c) = (1 - \text{Overlap}(K_u, K_c)) + \lambda \cdot \text{Reach}(K_u, K_c) + \mu \cdot D_{\text{vis}}(I_u, I_c), \quad (13)$$

where the first two (symbolic) terms come from the concept graph and the third (*visual*) term enters through the ViT-aligned space. $\text{Overlap}(K_u, K_c) = |K_u \cap K_c|/|K_c|$ measures content already covered, $\text{Reach}(K_u, K_c)$ is the normalised fraction of K_c within two hops of K_u in $\mathcal{G}$, and $D_{\text{vis}}(I_u, I_c) = 1 - \cos(\tilde{v}_u, \tilde{v}_c)$ measures the cosine distance between learner-side and course-side KG renderings.

Avoiding circularity. Computing D_{vis} from the same LoRA-adapted embeddings v_u, v_c that the scorer f_θ uses would make the acquisition signal partly a restatement of the model's own score. We therefore use $\tilde{v}_u, \tilde{v}_c$, the frozen CLIP [CLS] features of I_u, I_c computed before any LoRA update, which do not depend on θ and carry information the scorer has not yet used. The term is not redundant with the score: across the active pool the Spearman correlation between D_{vis} and $-f_\theta(u, c)$ is only 0.11 ± 0.03 , and the circular variant (adapted v) changes cold HR@10 by <0.2 pp while raising that correlation to 0.63 (Section IV-D). The symbolic terms reward exploration and bound reachability; the visual term captures shape-level divergence—a course that lives in a topologically distinct cluster from the learner's history is visually far from I_u , even when its raw concept overlap is small. We set $\lambda = 0.5, \mu = 0.3$ throughout. Hyperparameters (α, β, γ) are tuned once on a validation split and held fixed at (0.3, 0.3, 0.4) unless stated otherwise.

3.5. Iterative training loop

ALARM's overall procedure is in Algorithm 1 shown in figure 2. The crucial design choice is that step 8 updates only the LoRA adapters on both encoders (LLM and ViT), the visual

and textual projectors, and the fusion gate—never the backbone LLM or ViT weights themselves. In practice each round takes ~ 1.5 hours on a single A100 40GB on MOOCCube, vs. ~ 17 hours if both encoders were tuned end-to-end.

Algorithm 1 ALARM training loop (multimodal)

Require: Unlabelled pool $\mathcal{P}$, initial seed set $\mathcal{S}_0$, budget B , round size b , KG $\mathcal{G}$

- 1: Render KG images $\{I_c\}_{c \in \mathcal{C}}$ once; cache ViT features
- 2: Initialize $\theta = (\phi, \psi, \phi^V, \psi^V, \theta_{\text{Fuse}}, \tau)$, $\Delta_{\text{LoRA}}^{\text{LLM}} = \Delta_{\text{LoRA}}^{\text{ViT}} = 0$; train on $\mathcal{S}_0$
- 3: $\mathcal{S} \leftarrow \mathcal{S}_0$
- 4: **while** $|\mathcal{S}| < B$ **do**
- 5: Render learner-side images $\{I_u\}$ for u touched in $\mathcal{P} \setminus \mathcal{S}$
- 6: Compute $U(u, c)$ via Eq. (10) for all $(u, c) \in \mathcal{P} \setminus \mathcal{S}$
- 7: $\mathcal{Q} \leftarrow$ top- b samples by U
- 8: Query labels (oracle or platform log) for $\mathcal{Q}$
- 9: $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{Q}$
- 10: Update $(\theta, \Delta_{\text{LoRA}}^{\text{LLM}}, \Delta_{\text{LoRA}}^{\text{ViT}})$ on $\mathcal{S}$ for E epochs; both backbones frozen
- 11: **end while** **return** f_θ

Figure 2. Algorithm of ALARM’s overall procedure.

4. Experiments

4.1. Datasets and protocol

We evaluate on two public benchmarks.

- MOOCCube [1]: a large-scale MOOC repository from XuetangX. The full release contains 706 courses, about 199,199 users, and more than 100,000 concepts. We use a filtered subset and state the filtering rules explicitly, because the filter interacts directly with the cold-start population being studied. We (i) keep the 706 courses as the candidate set; (ii) retain only users with at least one completion-labelled enrollment among these courses; and (iii) drop users whose textual trace and knowledge-point tags are both empty (so that every learner has at least one usable modality). This yields 55,203 learners and approximately 458,453 enrollments. Knowledge-point annotations come from the released MOOCCube concept graph. Note that the candidate set is only 706 courses; HR@10 should be read against this pool size, not against a catalogue of tens of thousands of items.

- XuetangX-Subset: a smaller public subset released alongside [2], with 9,986 learners and 1,302 courses. We use it primarily for generalization.

Because the two references above [2], [14] do not in fact define a single shared MOOCCube split, we do not describe our split as a “standard” one. Instead we construct and release our own reproducible split (below). For each dataset we sort each learner’s interactions chronologically and hold out the most recent 10% as test and the preceding 10% as validation, with the remainder for training. Cold-start learners are those with $|H_u| \leq 3$ in the training partition (17.4% of test learners on MOOCCube, 22.1% on XuetangX).

a) Metrics: HR@10 and NDCG@10, plus cold-start variants restricted to cold learners. For ranking we use one positive per test interaction against 99 negatives sampled uniformly

from courses the learner has not enrolled in, resampled per seed. We report mean $\pm$ standard deviation over 5 random seeds, and assess the significance of ALARM over the strongest baseline with a two-sided paired t-test at the learner level (with a Wilcoxon signed-rank test as a non-parametric check).

b) Baselines: We compare against twelve methods spanning six categories. Classical CF: BPR - MF [24], NeuMF [12]. Classical CF: BPR - MF [24], NeuMF [12]. b) Graph CF: NGCF [13], LightGCN [14], KGAT [25]. Sequential: SASRec [15], BERT4Rec [16]. Cold-start specific: DropoutNet [17], CLCRec [18]. LLM-based: a zero-shot ChatGLM ranker following [20], TALLRec [19], and LLMRec [3]. We additionally include four active-learning ablations of ALARM that replace the acquisition function while keeping the encoder fixed: random (ALARM-Rand), entropy only (ALARM-H), coreset [23] (ALARM-Core), and BALD [22] (ALARM-BALD), the latter being the canonical Bayesian deep AL baseline.

c) Implementation: ChatGLM-6B with FP16 weights as the language encoder, frozen. CLIP ViT-L/14 [27] as the vision encoder, frozen. KG subgraph images are rendered offline with NetworkX and a deterministic Fruchterman-Reingold spring layout [34], each 224×224 RGB (rendering details in Section III-C). Projectors are 2-layer MLPs (hidden 512, GELU, dropout 0.2). LoRA rank 8, applied to query and value matrices in the top 3 transformer layers of each LLM tower and the top 3 ViT blocks. The fusion gate is a single linear layer over $\mathbb{R}^{2d} \rightarrow \mathbb{R}^d$. AdamW with learning rate 1×10^{-4} , batch size 256, 5 epochs per active-learning round. Active learning starts with a 5% seed set and adds 5% per round until the budget. All experiments use a single NVIDIA A100 40GB GPU.

d) Reproducibility: We will release the code, the processed data splits, and the KG-rendering scripts upon acceptance. The acquisition weights (α, β, γ) and the novelty weights (λ, μ) were selected by a grid search on the validation split of MOOCCube only $(\alpha, \beta, \gamma$ over a simplex grid of step 0.1; $\lambda, \mu \in \{0.1, 0.3, 0.5, 0.7\})$, then frozen and reused unchanged on XuetangX; LoRA rank was searched in $\{4, 8, 16\}$ and the projector width in $\{256, 512\}$. All baselines were tuned on the same validation split with the grids reported in each original paper (learning rate, embedding size, number of layers, and regularization), selecting the configuration with the best validation NDCG@10; the exact grids and selected values are provided with the released code. Seeds $\{0, 1, 2, 3, 4\}$ control data splitting, negative sampling, and parameter initialization jointly.

4.2. Main results

Table 1 compares ALARM at a 20% label budget against fully supervised baselines. ALARM reaches parity on overall HR@10/NDCG@10 and shows large absolute margins on cold-start metrics.

ALARM's gain over LightGCN on overall HR@10 is moderate (+8.9% relative on MOOCCube), but the cold-start HR@10 gain is large ($0.241 \rightarrow 0.358$, +48.5% relative); the same pattern repeats on XuetangX (cold HR@10: $0.218 \rightarrow 0.341$, +56.4% relative). Comparing against the two cold-start specialists is more informative than against generic CF: ALARM beats DropoutNet [17] on cold HR@10 by +26.5% relative on MOOCCube

and CLCRec [18]—a recent contrastive cold-start method—by +17.8% relative, while matching or exceeding the warm-user performance of the strongest LLM baselines (TALLRec, LLMRec). The zeroshot ChatGLM baseline confirms a known finding [20]—an unfine-tuned LLM trails graph methods on warm users but leads on cold ones—supporting its use as a frozen encoder. Because the per-learner cold-start test set is small and noisy, we rely on significance rather than point estimates: ALARM’s gain over the strongest baseline is significant at $p < 0.01$ in every column (paired t-test, confirmed by Wilcoxon), with the Caption’s standard deviations (≤ 0.014 on cold metrics) well below the observed margins (e.g. 0.358 vs. CLCRec’s 0.304). Among the active-learning ablations, the gap between ALARM – BALD [22] and full ALARM (0.498 vs. 0.548 on overall HR@10; 0.268 vs. 0.358 on cold HR@10) isolates the joint contribution of the vision-grounded novelty term and the multi-modal encoder: Bayesian uncertainty alone, applied to a text-only retriever, is not enough; the visual reading of the knowledge graph plus the education-specific signal together close the cold-start gap. Section IV–D decomposes this further.

Table 1. Main results on mooccube and xuetangx-subset.

Method	MOOCCube				XuetangX-Subset			
	HR@10	NDCG@10	C-HR	C-NDCG	HR@10	NDCG@10	C-HR	C-NDCG
<i>Classical CF</i>								
BPR-MF [24]	0.412	0.218	0.183	0.094	0.387	0.201	0.156	0.081
NeuMF [12]	0.438	0.232	0.192	0.097	0.408	0.215	0.171	0.087
<i>Graph CF</i>								
NGCF [13]	0.482	0.261	0.227	0.111	0.451	0.244	0.205	0.101
LightGCN [14]	0.503	0.272	0.241	0.118	0.471	0.256	0.218	0.107
KGAT [25]	0.521	0.284	0.262	0.131	0.486	0.263	0.234	0.116
<i>Sequential</i>								
SASRec [15]	0.487	0.265	0.197	0.099	0.453	0.241	0.184	0.092
BERT4Rec [16]	0.479	0.258	0.193	0.097	0.446	0.237	0.181	0.090
<i>Cold-start specific</i>								
DropoutNet [17]	0.456	0.246	0.283	0.137	0.421	0.226	0.262	0.128
CLCRec [18]	0.498	0.269	0.304	0.151	0.464	0.249	0.281	0.139
<i>LLM-based</i>								
ChatGLM zero-shot [20]	0.358	0.184	0.291	0.142	0.331	0.171	0.276	0.137
LLMRec [3]	0.512	0.278	0.297	0.148	0.479	0.259	0.278	0.138
TALLRec [19]	0.529	0.289	0.276	0.139	0.498	0.271	0.259	0.128
<i>Active-learning ablations of ALARM (20% labels)</i>								
ALARM-Rand	0.461	0.247	0.234	0.114	0.428	0.231	0.213	0.104
ALARM-H	0.494	0.268	0.262	0.131	0.461	0.249	0.241	0.118
ALARM-BALD [22]	0.498	0.270	0.268	0.133	0.464	0.251	0.247	0.121
ALARM-Core [23]	0.508	0.276	0.273	0.137	0.474	0.257	0.249	0.123
ALARM (ours, 20%)	0.548	0.305	0.358	0.189	0.516	0.282	0.341	0.178

4.3. Label-efficiency curve

Figure 3 shows HR@10 on MOOCCube as a function of label budget for ALARM and three reference methods retrained on the same fraction of labels. Multimodal ALARM matches LightGCN’s fully supervised HR@10 of 0.503 with about 20% of labels, and matches TALLRec’s with about 35%. ALARM-Rand requires roughly 1.6× more labels to

reach the same performance as full ALARM, quantifying the benefit of cognitive–uncertainty selection, with a text–only ablation falling between the two.

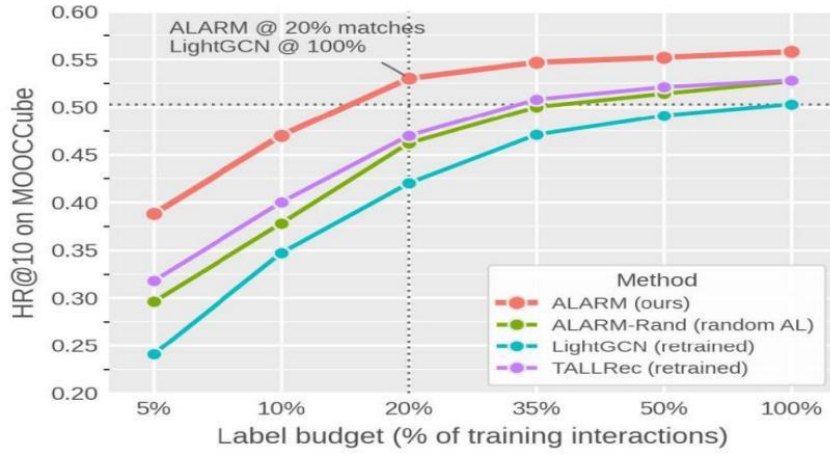


Figure 3. Label–efficiency curves on MOOCube.

4.4. Ablation: vision pathway and acquisition terms

Table 2 ablates the vision pathway and each term of Eq. (10) on MOOCube at a 20% budget. Two findings stand out. First, removing the vision tower entirely (text–only ALARM) costs 12.8% relative on cold HR@10—the second–largest drop, after the novelty term itself. This isolates the contribution of the visual reading of the KG, independently of the symbolic Overlap/Reach signal which the text–only variant still retains. Second, removing only the visual sub–term D_{vis} from the novelty function (keeping the ViT in the encoder, removing it from acquisition) costs 5.9% relative, showing that the vision pathway is useful both as an encoder and as an acquisition signal. Removing the full KG novelty term gives the largest single drop (–17.3% relative on cold HR@10), supporting our claim that domain–specific signal drives most of the coldstart gain. The circular D_{vis} row confirms our design choice: using the scorer’s own LoRA–adapted embeddings instead of the frozen pre–adaptation features changes cold HR@10 by <0.2 pp (within one SD). All gaps between Full ALARM and the acquisition–term ablations exceed one SD on cold HR@10.

Table 2. Ablation of the vision pathway and cognitive–uncertainty components on moocube at 20% budget.

Variant	HR@10	C–HR
Full ALARM($\alpha=0.3, \beta=0.3, \gamma=0.4, \mu=0.3$)	0.548 $\pm$.004	0.358 $\pm$.011
<i>Vision–pathway ablations</i>		
vision tower (text–only ALARM)	0.512 $\pm$.005	0.312 $\pm$.013
visual novelty term D_{vis} only	0.531 $\pm$.004	0.337 $\pm$.012
circular D_{vis} (adapted v, not frozen)	0.547 $\pm$.005	0.357 $\pm$.012
<i>Acquisition–term ablations</i>		
MC variance $\alpha=0$	0.534 $\pm$.004	0.347 $\pm$.012
entropy $\beta=0$	0.536 $\pm$.005	0.343 $\pm$.011
KG novelty (whole $N, \gamma=0$)	0.518 $\pm$.005	0.296 $\pm$.013
all three (random sampling)	0.461 $\pm$.006	0.234 $\pm$.014

4.5. Multimodal backbone and KG robustness

Table 3 summarizes two robustness analyses. Top: we vary the language and vision encoders. LLM-side performance scales smoothly with model size up to ChatGLM-6B; the additional jump to LLaMA-3-8B is marginal. Vision-side, CLIP ViT-L/14 outperforms the smaller ViT-B/16 by ~ 1.2 pp on cold HR@10 at modest extra cost, while SigLIP ViT-L matches CLIP. Replacing the visual encoder with a graph neural network on the same KG adjacency (GraphSAGE over $\mathcal{G}$) loses ~ 2.7 pp on cold HR@10. To rule out an undertrained baseline, GraphSAGE was tuned on the validation split over depth, hidden width, fan-out, aggregator, and dropout (best of 18 configurations), under the same label and LoRA-round budget as the visual pathway; the gap suggests visual reading captures spatial topology that message-passing over the same adjacency does not directly expose. Bottom: ALARM's largest gain comes from the KG-novelty term and the visual reading. With only a tag taxonomy and no edges (Reach=0, disconnected nodes), cold HR@10 falls ~ 5.1 pp; with no KG at all, cold-start drops to text-only ALARM with random selection. To separate structure from a particular rendering, we test two deterministic layouts (Fruchterman-Reingold and Kamada-Kawai) and a random shuffle: the deterministic layouts fall within 0.3 pp of each other while the shuffle costs ~ 3.4 pp (larger than either gap and than one SD), so it is the spatial regularities of a structure-aware layout, not the particular embedding or tag identity, that the ViT reads.

Table 3. Top: multimodal backbone study.

Bottom: kg robustness with clip vit-l/14 and chatglm-6b fixed.			
Variant	Params (B)	HR@10	C-HR
<i>LLM backbone (vision: CLIP ViT-L/14)</i>			
BERT-base	0.11	0.494	0.301
e5-large	0.34	0.516	0.323
Qwen2-1.5B	1.5	0.529	0.334
ChatGLM-6B (default)	6.2	0.548	0.358
LLaMA-3-8B	8.0	0.551	0.361
<i>Vision encoder (LLM: ChatGLM-6B)</i>			
GraphSAGE on $\mathcal{G}$ (no vision)	—	0.530	0.331
CLIP ViT-B/16	0.09	0.541	0.346
SigLIP ViT-L (SoViT)	0.42	0.546	0.357
CLIP ViT-L/14 (default)	0.30	0.548	0.358
<i>KG variant (default backbones)</i>			
Full KG, spring (Fruchterman - Reingold)	—	0.548 $\pm$ 004	0.358 $\pm$ 011
Full KG, Kamada-Kawai layout	—	0.546 $\pm$ 005	0.355 $\pm$ 012
Random-shuffle layout	—	0.531 $\pm$ 006	0.324 $\pm$ 013
Tag-only (no edges, no layout)	—	0.527 $\pm$ 005	0.307 $\pm$ 013
No KG ($\gamma=0$, no vision tower)	—	0.498 $\pm$ 006	0.272 $\pm$ 014

4.6. Compute cost

Table 4 reports per-round and total wall-clock cost on a single A100 40GB. LoRA-only updates make the multimodal protocol practical: each active-learning round takes ~ 1.5 h for the LoRA fine-tune of both encoders plus ~ 7 min for acquisition scoring—versus roughly

$12 \times$ the round time for full fine-tuning, which would put the 12-round curve out of reach. One-off KG rendering and frozen ViT/LLM feature extraction add ~ 4.5 h. The full 12-round curve is ~ 23.5 h, up from ~ 21 h for text-only ALARM.

Table 4. Wall-clock cost on mooccube, single a100 40gb, 12 active-learning rounds, batch 256.

Phase	Single round	Full curve
KG subgraph rendering (one-off)	—	0.4 h
Frozen ViT encoding of KG (one-off)	—	2.0 h
Frozen LLM encoding (one-off)	—	2.1 h
Acquisition scoring	7 min	1.4 h
LoRA fine-tune (LLM + ViT)	1.5 h	18.0 h
Total wall-clock	—	~ 23.5 h

4.6. Educational implications and deployment

Beyond retrieval metrics, ALARM fits a real teaching workflow. First, the population it targets—learners with three or fewer enrollments—is exactly the one a platform most wants to retain, so improving cold-start ranking is an equity as much as an accuracy question: it disproportionately helps first-time and under-served learners who generate little collaborative signal. Second, the vision-grounded novelty term is a pedagogical lever, not just a heuristic: it rewards courses a bounded number of prerequisite hops beyond the learner’s current frontier, so an instructional designer can tune it toward consolidation or stretch, and the rendered subgraph gives a human-readable rationale for each suggestion. Third, the active-learning loop matches how educational labels actually arrive—completion, certification, or expert curation—telling a course team which few learner-course pairs are worth eliciting; in deployment the oracle is the platform’s own completion log, so ALARM runs on data a MOOC already collects.

4.7. Discussion and limitations

ALARM’s gains depend on a course knowledge graph that can be rendered; without one, the novelty term degenerates to keyword overlap and the cold-start gain shrinks (Table III). Performance is sensitive to layout and to encoder size (Section IV–E). Finally, ALARM assumes an oracle label on query; in deployment that oracle is the platform’s enrollment outcome, which arrives with delay and noise. Extending the framework to learner-facing visual modalities (slide images, board photographs) is a natural next step the architecture supports without structural change.

5. Conclusion

We presented ALARM, which—building on established visual graph-reading methods—applies visual reading of a course knowledge graph to recommendation for the first time, and derives from it a vision-grounded novelty signal that drives cognitive-uncertainty active

learning under LoRA—only updates that keep multi-round operation tractable on a single GPU. On two public benchmarks, ALARM at 20% of the label cost matches fully supervised graph and LLM baselines and improves cold-start HR@10 by +48.5% and +56.4% relative; ablations isolate the visual pathway (12.8%) and the novelty term (17.3%). We believe visual reading of structured side information—knowledge graphs here, prerequisite trees or curriculum charts elsewhere—is a broadly useful primitive for recommenders beyond education.

Acknowledgments

This work was funded in part by the General Project of Philosophy and Social Sciences Research in Jiangsu Colleges and Universities under Grant number 2025SJYB1380, in part by Jiangsu Finance & Accounting Vocational College Institutional Teaching Reform Project under Grant number number 2025jsckjg13.

Authors' contribution

Author Contributions: Conceptualization, H.C.; methodology, H.C.; software, H.C.; validation, Y.C., X.M.; formal analysis, H.C.; investigation, H.C.; resources, H.C.; data curation, H.C.; writing—original draft preparation, H.C.; writing—review and editing, H.C.; visualization, H.C.; supervision, X.M.; project administration, H.C.; funding acquisition, H.C. All authors have read and agreed to the published version of the manuscript.

Conflicts of interests

The authors declare no conflicts of interest.

Ethical statement

Ethical review and approval were waived for this study due to the use of publicly available anonymous datasets.

References

- [1] Yu J, Luo G, Xiao T, Zhong Q, Wang Y, *et al.* MOOCCube: A large-scale data repository for NLP applications in MOOCs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, July 5–10, 2020, pp. 3135–3142.
- [2] Zhang J, Hao B, Chen B, Li C, Chen H, *et al.* Hierarchical Reinforcement Learning for Course Recommendation in MOOCs. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019, pp. 435–442.
- [3] Wei W, Ren X, Tang J, Wang Q, Su L, *et al.* LLMRec: Large Language Models with Graph Augmentation for Recommendation, In *WSDM'24*, March 4–8, 2024, pp. 806–815.
- [4] Wang J, Lu H, Caverlee J, Chi E, Chen M. Large Language Models as Data Augmenters for Cold-Start Item Recommendation. In *WWW'24: Companion Proceedings of the*

- ACM Web Conference*, 2024, pp. 726–729.
- [5] Rubens N, Elahi M, Sugiyama M, and Kaplan D. Active learning in recommender systems. *Recommender Systems Handbook 2nd ed.*, Springer, 2015, pp. 809–846.
- [6] Du Y, Sun Z, Wang Z, Chua H, Zhang J, *et al.* Active Large Language Model-based Knowledge Distillation for Session-based Recommendation. *arXiv* 2025, arXiv:2502.15685.
- [7] Bayer M, Lutz J, Reuter C. ActiveLLM: Large Language Model-based Active Learning for Textual Few-shot Scenarios. *arXiv* 2024, arXiv:2405.10808.
- [8] Lin Y, Wu Z. Learning Path Recommendation Enhanced by Knowledge Tracing and Large Language Model. *Electronics*. 2025, 14(22):4385.
- [9] Hu E, Shen Y, Wallis P, Zhu Z, Li Y, *et al.* LoRA: Low-rank Adaptation of Large Language Models. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.
- [10] Houshy N, Giurghi A, Jastrzebski S, Morrone B, Laroussilhe Q, *et al.* Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, California, 2019, pp. 2790–2799.
- [11] Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer*. 2009, 42(8): 30–37.
- [12] He X, Liao L, Zhang H, Nie L, Hu X, *et al.* Neural Collaborative Filtering. In *WWW'17: Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 173–182.
- [13] Wang X, He X, Wang M, Feng F, Chua T, Neural Graph Collaborative Filtering. In *SIGIR'19: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 165–174.
- [14] He X, Deng K, Wang X, Li Y, Zhang Y, *et al.* LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR'20: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 639–648.
- [15] Kang W, McAuley J. Self-Attentive Sequential Recommendation. In *Proceedings of the 2018 IEEE International Conference on Data Mining*, 2018, pp. 197–206.
- [16] Sun F, Liu J, Wu J, Pei C, Lin X, *et al.* BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *CIKM'19: Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1441–1450.
- [17] Volkovs M, Yu G, Poutanen T. DropoutNet: addressing cold start in recommender systems. In *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 4964–4973.
- [18] Wei Y, Wang X, Li Q, Nie L, Li Y, *et al.* Contrastive Learning for Cold-Start Recommendation. In *MM '21: Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5382–5390.
- [19] Bao K, Zhang J, Zhang Y, Wang W, Feng F, *et al.* TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. In *RecSys '23: Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 1007–1014.

- [20] Ma B, M. Khan M, Yang T, Polyzou A, Konomi S. How Good Are Large Language Models for Course Recommendation in MOOCs?. *arXiv* 2025, arXiv:2504.08208.
- [21] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In *ICML'16: Proceedings of the 33rd International Conference on International Conference on Machine Learning*, 2016, pp. 1050–1059.
- [22] Gal Y, Islam R, Ghahramani Z. Deep Bayesian active learning with image data. In *ICML'17: Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 1183–1192.
- [23] O. Sener and S. Savarese, “Active learning for convolutional neural networks: A core–set approach,” in *ICLR*, 2018.
- [24] Rendle S, Freudenthaler C, Gantner Z, Lars S. BPR:Bayesian personalized ranking from implicit feedback. In *UAI'09: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.
- [25] Wang X, He X, Cao Y, Liu M, Chua T. KGAT: Knowledge Graph Attention Network for Recommendation. In *KDD'19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 950–958.
- [26] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, *et al.* An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.
- [27] Radford A, Kim J, Hallacy C, Ramesh A, Goh G, *et al.* Learning Transferable Visual Models from Natural Language Supervision. In *Proceedings of the 38 th International Conference on Machine Learning*, 2021, pp. 8748–8763.
- [28] Lee K, Joshi M, Turc I, Hu H, Liu F, *et al.* Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In *ICML'23: Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 18893–18912.
- [29] Kim B, Kim J, Lee D, Jang B. Visual Question Answering: A survey of Methods, Datasets, Evaluation, and Challenges. *ACM Comput. Surv.*, 2025, 57(10): 249:1–35.
- [30] Fruchterman T, Reingold E. Graph drawing by force-directed placement. *Softw.: Pract. Exper.* 1991, 21(11): 1129–1164.
- [31] Ai Q, Li R, Wang M, Jiang H. Graph-to-Vision: Multi-graph understanding and reasoning using vision-language models. *arXiv* 2025, arXiv:2503.21435.
- [32] Wang J, Ju J, Luan J, Deng Z. LLaVA-SG: Leveraging Scene Graphs as Visual Semantic Expression in Vision-Language Models. In *Proceedings of 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hyderabad, India, April 06-11, 2025.
- [33] Zhu Y, Bai X, Chen K, Xiang Y, Yu J, *et al.* Benchmarking and Improving Large Vision-Language Models for Fundamental Visual Graph Understanding and Reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025(1), pp. 30678–30701.
- [34] Fruchterman T, Reingold E. Graph Drawing by Force-Directed Placement.*Softw.: Pract. Exper.* 1991, 21(11): 1129–1164.