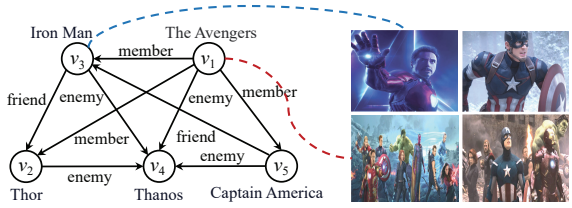
CrossETR: A Semantic-driven Framework for Entity Matching across Images and Graph

Qin Yuan[§] Zhenyu Wen[†] Jiaxu Qian[†] Ye Yuan[§] Guoren Wang[§]
[§]Beijing Institute of Technology [†]Zhejiang University of Technology
yuanq1020@gmail.com; wenluke427@gmail.com; q1anjiaxu001@gmail.com
yuan-ye@bit.edu.cn; wanggrbit@126.com

***Abstract*—Entity matching (EM) aims to identify whether two entities from different data sources refer to the same real-world entity. Most existing cross-modal EM assume that images have simple scenes containing few objects, or do not fully consider the cross-modal knowledge associated with entities. To support more practical application scenarios such as multi-modal knowledge graph integration and visual question answering in data lakes, we introduce our problem of semantic-driven EM across graph and images in this paper. Current semantically matching solutions over cross-modal data face the obstacle of low training efficiency, since their time complexity quadratically grows with the number of entities. To alleviate this issue, we present a novel framework (namely CrossETR) that follows an exploration-then-refinement paradigm. It firstly proposes a candidate exploration policy to boost the training efficiency, which explores candidate pairs according to entity correlations and captures structural semantics by adaptive sampling the most informative neighborhood subgraphs. Secondly, it refines the cross-modal entity representations to break modality heterogeneity to support unsupervised matching prediction. Extensive experimental evaluations on three publicly available benchmarks demonstrate the superiority of CrossETR over state-of-the-art approaches in terms of effectiveness and efficiency. Furthermore, a case study highlights that our proposed semantic-driven EM is promising to improve the performance of downstream tasks such as multi-modal knowledge graph integration.**

I. Introduction

Entity matching (EM) is a fundamental and critical task in data integration research, which aims to identify equivalent entities between different data sources [1], [2]. With the rapid increase of data variety in data lakes [3], [4], EM becomes increasingly important for practical applications. The structural information of entities in data lake is usually stored in a knowledge graph, while the visual features and scene information related to the entities are depicted in the form of images. To link these entities across images and graph, the structural and visual features of real-world entities can be completed to support downstream tasks such as multi-modal knowledge graph construction and visual question answering [5], [6]. For example, Figures 1(a) and 1(b) describe the structural information and visual scene about movie characters in a knowledge graph and a set of images, respectively. From the entities linked by blue dashed line, we can clearly know what the "Iron Man" looks like. The entities connected by red dashed line describe the visual characteristics and structural information of "The Avengers", i.e., its members and appearances. Entity matching across images and graph can enhance multi-modal knowledge graph construction by integrating data from graph and images [5], which motivates our work in this paper.



(a) Part of knowledge graph G (b) An image repository $\boldsymbol{I}$

Fig. 1. Example of entity matching across images and graph.

Most existing cross-modal matching solutions assume that visual image has a simple scene containing few objects [7], [8] such as the top images shown in Figure 1(b), or do not fully consider the cross-modal knowledge about entities [9], [10] such as the structural information of "The Avengers" shown in Figure 1(a). They explore the match described in blue dashed line well, but may not be able to identity the match depicted in red dashed line. This is because the image scene is complex and simply regarding it as one entity to perform cross-modal EM will result in missing or incomplete matches.

Following the above motivations, we address the problem of entity matching across images and graph by considering their structural semantics and scene semantics in this paper.

To efficiently perform semantic-driven EM over multi-source, existing methods usually properly encode the entity semantics as representations and then make predictions based on similarity calculation or matching probability [9], [11].They are typically classified into two main groups. Firstly, dual-based group is to first extract entity embeddings by training feature encoders for different data sources and then measure their similarity distances, as shown in Figure 2(a). For example, CLIP [12] and Sudowoodo [9] first encode the input pairs separately and then train the models contrastively to keep similar pairs close and dissimilar pairs apart. Due to modality heterogeneity, overly simple feature fusion may lead to insufficient representations for multi-modal entities and further affect the overall EM performance, as discussed in [13]. Secondly, fusion-based group leverages pre-trained models to initialize entity embeddings and then maps them into a common feature space based on Transformer architecture, as shown in Fig-

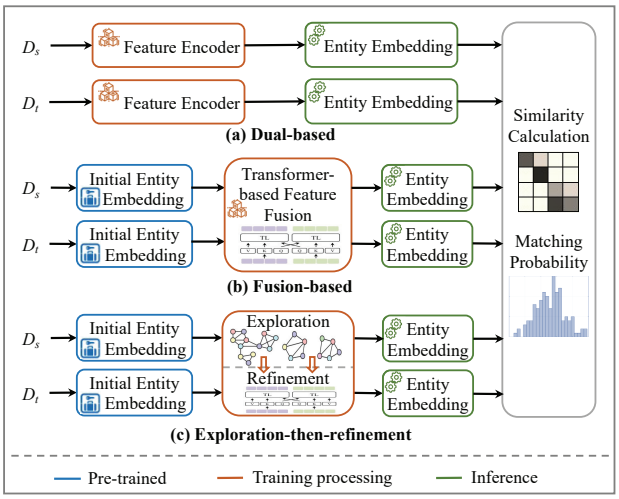


Fig. 2. Semantic-driven EM: (a) Dual-based approaches, which measure the distances of entity embeddings across different sources; (b) Fusion-based approaches, which encode entities from different sources into a common feature space; and (c) our exploration-then-refinement, which explores candidates and then performs cross-modal refinement, proposed in our CrossETR.

ure 2(b). For example, IMRAM [14] encodes texts and images into a common feature space using an iterative matching mechanism with recurrent attention. PromptEM [10] unifies heterogeneous data as textual sequences and transforms the generalized EM as a masked language task to predict target words in a low-resource setting. These methods fuse entity features from different modalities as much as possible, but are still empirically limited by memory usage and time complexity that grows quadratically with the number of entity pairs [13].

From the above investigations and comparisons, we aim to develop a semantic-driven solution to boost the training efficiency of cross-modal EM. This remains a challenging endeavor. Firstly (C1), *how to efficiently explore candidate vertices and effectively capture their structural semantics related to images?* The existing solutions discussed above with quadratic time complexity in entity number is expensive for large-scale knowledge graph and image repository. To improve the training efficiency, we intuitively expect to prioritize exploring those candidate vertices that are most likely to match the images. Furthermore, the most popular graph representation methods [15], [16] usually capture the structural semantics of vertices by expanding on their adjacent neighborhood substructures. However, vertices have too many neighbors in real-world large graphs such as Freebase [17] and WordNet [18], which causes scalability issues and also introduces noise into the matching models. Secondly (C2), *how to refine entity representations to support cross-modal EM in an unsupervised manner?* Most existing semantic-driven methods depend on vast label annotations to train the matching models, which is labor-intensive even unavailable in large data lake [9], [10]. It remains a challenge to extract entity features through structure and scene interactions and then unsupervised back-propagate them to train the matching model.

To tackle these challenges, we propose a novel semantic-driven framework to address EM across images and graph, namely CrossETR. It follows an *exploration-then-refinement* paradigm that first explores candidates based on entity correlations and then performs cross-modal refinement, as shown in Figure 2(c). To improve the training efficiency of cross-modal EM (for C1), we propose a *candidate exploration policy* to reduce redundant objects of images and then samples the most important neighbors for candidate vertices to capture their structural semantics relevant to images. Subsequently, we introduce a *cross-modal refinement* method (for C2) to refine entity representations by performing cross-modal feature fusion between vertices and images, and guide the training of the matching model in an unsupervised manner. Our contributions are summarized as follows.

(1) Semantic-driven matching framework. To the best of our knowledge, this is the first work on semantic-driven entity matching across images and graph. As data diversity and volume increase, our proposed exploration-then-refinement paradigm will become popular in different practical scenarios. Details will be shown in Section III.

(2) Candidate exploration policy. We propose a candidate exploration policy consisting of instance selection and adaptive subgraph sampling to discover relevant candidate pairs and boost training efficiency, as described in Section IV.

(3) Cross-modal refinement. We present a cross-modal feature fusion to break modal heterogeneity, and an unsupervised training mechanism to obtain entity associations to address our semantic-driven EM, as detailed in Section V.

(4) Extensive Experiments. We conduct comprehensive evaluations on semantic-driven EM task compared with some state-of-the-art approaches. Extensive experimental results verify the superiority of our CrossETR in terms of effectiveness and efficiency. A case study indicates that semantic-driven EM can significantly improve the performance of multi-modal knowledge graph integration, illustrated in Section VI.

II. Problem Definition

In this section, we formally present the problem definition of entity matching across graph and images. The notions that are frequently used in this paper are summarized in Table I.

A. Preliminary

Data Graph. We consider a directed graph defined as $G = (V, E, L)$, where (a) V is a vertex set; (b) $E \subseteq V \times V$ is a set of edges; (c) L is the set of all unique words contained in the labels of edges and vertices; and (d) $L(v)$ and $L(e)$ represent the labels of vertex $v \in V$ and edge $e \in E$, respectively.

Structured and semi-structured data in data lakes can be converted as graph by encoding tuples or keys into vertices and foreign keys or references as relationships of graph [3], [4]. By using some sentence parsing models based on language structures [19], [20], unstructured text documents can be constructed as a graph where named entities are represented as vertices and their syntactic relations are edges.

Images. An image I captures the scene that includes various objects such as people, landscapes, backgrounds contributing to the overall understanding or interpretation of the image. Multimedia data can be cut into images based on frames [21], [22], and a video is collected as a set of images. This will be discussed in our future works.

Instance Segmentation. Instance segmentation is a typically instance-level image understanding task [23], [24] that focuses on delineating an object with a segmentation mask in the given image. It is different from the object detection tasks to recognize individual objects in the given scene with a bounding box, which may include parts of the background and is difficult to separate objects from their surroundings.

Segmentation Anything Model (SAM, [24]) has released as a state-of-the-art foundation model for image segmentation by Meta AI and can be used to generate masks for all objects in an image. As for an image $I \in \mathbb{I}$, $O = \{o_1, ..., o_n\}$ is a set of objects segmented from I using SAM. Each object $o_i \in O$ is related to a cropped pixel of I and associated with different concepts. Here n is the number of segmented objects from I. Object $o_i = (b_i, m_i)$ consists of a bounding box b_i and a mask feature map m_i. A bounding box b_i is a tuple (x, y, w, h, c), where (x, y) is the top-left corner coordinate of bounding box, w and h are its width and height, respectively. $b_i[c]$ is the segmentation confidence score of b_i. A feature map m_i is an object representation in high-dimensional space.

Graph Neural Network (GNN). GNNs learn the representation x_i of vertex v_i following an iterative neighborhood aggregation scheme, which captures the structural information associated by its neighbors via graph convolution layers [15]. Let X denote the feature matrix of G, in which $x_i = X[v_i, :]$ is a high-dimensional attribute vector of $v_i \in V$.

For each v_i, its representation at the $(l+1)$-th layer $x_i^{(l+1)}$ is learned by firstly aggregating the neighbor feature vectors and then concatenating with the l-th layer vertex representation. Let $N(v_i)$ be a set of neighbors of v_i. $x_i^{(l+1)}$ is formally defined as follows:

$$x_i^{(l+1)} = \text{Combine}^{(l+1)}(x_i^{(l)}, x_{N(v_i)}^{(l+1)}), \tag{1}$$

$$x_{N(v_i)}^{(l+1)} = \text{Aggregate}^{(l+1)}(W^{(l+1)}, \{x_j^{(l+1)}, v_j \in N(v_i)\}), \tag{2}$$

where $x_i^{(l)}$ is the representation of v_i at the l-th layer with $x_i^{(0)} = x_i$. The Aggregate and Combine are message passing functions in GNN. $W^{(l+1)} \in \mathbb{R}^{d^{(l)} \times d^{(l+1)}}$ is the learnable aggregation matrix in the $(l+1)$-th layer, and $d^{(l)}$ denotes the hidden dimension at the l-th layer.

B. Problem Formulation

Entity Matching. Entity matching (EM) [1], [2] aims to determine whether two entities refer to the same real-world entity, which is regarded as a *pairwise matching problem* where two matched entities are called a *matching pair*. Regarding the entity *semantic* as an *embedding* generated by some representation-based methods [9], [10], and entities with related semantics are close in the vector space. We formally define as follows.

TABLE I
FREQUENTLY USED NOTATIONS.

Symbol	Description
I	an image in the image repository $\mathbb{I}$
O	the segmented objects from I
$G = (V, E, L)$	a directed data graph
$o_i = (b_i, m_i)$	the bounding box and feature map of o_i
$\mathcal{M}, \mathcal{H}, \mathcal{F}, \mathcal{A}$	the matching model including a explorer $\mathcal{H}$, a scoring function $\mathcal{F}$ and an aggregator $\mathcal{A}$
$\mathcal{S}$	the matching pairs for V and $\mathbb{I}$
O^*	the key objects in an image
$f_b(b, R)$	the occlusion factor for a box b in R
G_s	the sampled subgraphs for candidates C
Z_v, Z_o	the refined vertex and object representations
$p(v_j \mid v_i)$	the sampling probability of v_j for v_i
$f(V_s)$	the information entropy of the subgraph induced by vertices V_s
$g(v_j, S)$	the information gain of adding v_j into S
$\nabla_\theta \mathcal{L}$	the gradient of the training loss $\mathcal{L}$ with respect to θ in the sampling operation
v_i^c	the scene-aware vertex representation of v_i
$L_s(v, I)$	the matching loss between v and I

Definition 1: (Matching Pair) For each entity pair (x_1, x_2), x_1 and x_2 form a *matching pair* if and only if their *semantics* are related. The semantic relevance can be measured by using a given similarity function sim to their embeddings, where sim is usually the cosine similarity function. The larger $sim(x_1, x_2)$ is, the more semantically related they are.

In this paper, we consider the task of entity matching across two data sources with different modalities. We formally formulate our *semantic-driven entity matching* problem across graph and image as follows.

Definition 2: (Semantic-driven Entity Matching) Given a graph data $G = (V, E, L)$ and an image repository $\mathbb{I} = \{I_i\}_{1 \le i \le N}$ with N images, *entity matching* is to find all matching pairs $\mathcal{S}$ of the vertices in V and the images in $\mathbb{I}$ such that:

$$\mathcal{S} = \{(v, I) | v \Leftrightarrow I, v \in V, I \in \mathbb{I}\}, \tag{3}$$

where $\Leftrightarrow$ denotes the equivalent entities in the real world. (v, I) is a matching pair calculated by the semantic relevance of their embeddings.

We do not assume a one-to-one mapping across two different data sources [25], but mainly focus on the binary relation between V and $\mathbb{I}$ that indicates whether two entities match or not [9], [11]. The left entity of the matching pair is called source entity and the right one is called target entity. For each source entity, the target entity is ranked according to their semantic *matching scores*. The higher score indicates the more likelihood that the target entity matches the source entity.

Definition 3: (Matching Score) For each matching pair of $v \in V$ and $I \in \mathbb{I}$, its matching score is computed as:

$$C(v, I) = \frac{1}{|V_s|} \sum_{v_i \in V_s} \max_{o_j \in O} sim(v_i, o_j) \tag{4}$$

where $sim(\cdot, \cdot)$ is an embedding similarity function between objects and vertices. V_s is the neighboring vertices of v and

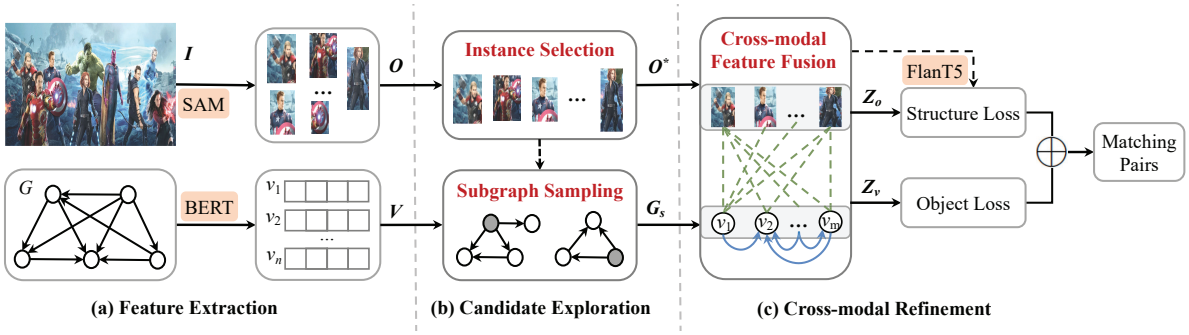


Fig. 3. Overall architecture of our proposed framework CrossETR. The orange zones are collaborated by excellent pre-trained large models.

O is a set of segmented objects from I. How to extract V_s for v and O for I will be detailed in the following sections.

Example 1: Figure 1 shows a matching pair of the vertex v_1 and the image connected by red dashed line, both of which describes similar semantics of "Members of The Avengers include Iron Man, Thor, Caption America, etc.". This matching pair is driven by the structural semantics of v_1 and the scene semantics depicted by the objects contained in I.

III. Framework Overview

Following previous studies on embedding-based entity matching [9], [11], to determine whether two entities match or not, a matching model is trained to capture their semantics and calculate the similarities of their representations by using a matching score. Typically, the semantics of vertices are represented by considering their neighborhood structures, which can be achieved through some graph representation methods such as GraphSage [15], GNN [16]. While the semantics of images are captured by the objects or instances contained in the scenes, which can be detected in object detection and instance segmentation methods [26], [27]. Therefore, we consider the scene semantics of images and the structural semantics of vertices to perform semantic-driven entity matching and decide whether an image and a vertex form a matching pair.

Given a graph G and a set of images $\mathbb{I}$, our CrossETR framework aims to align an image $I \in \mathbb{I}$ to a vertex $v \in V$ with similar semantics. To determine whether two entities match or not, we first employ representation models to generate the semantic representations of entities, and then leverage the matching score to calculate the relevance between these representations. During encoding these entities, CrossETR follows two phrases: exploration and refinement. It first explores candidate vertices for each image according to entity correlations and samples neighborhood subgraphs to capture the semantics of these candidates. The second phase refines the cross-modal entity representations to calculate their matching scores. Therefore, CrossETR is to learn a matching mechanism $\mathcal{M} = \{\mathcal{H}, \mathcal{F}, \mathcal{A}\}$ in an unsupervised manner, where $\mathcal{H}$ and $\mathcal{F}$ are a candidate *explorer* and a matching scoring function for vertices with respect to images, respectively. $\mathcal{A}$ is an *aggregator* designed to refine vertex and image representations by cross-modal feature fusion for $\mathbb{I}$ and V. Figure 3 overviews

Algorithm 1: CrossETR Training

Input: A graph $G = (V, E)$, a set of images $\mathbb{I}$, learning rate η, epoch number n.
Output: A cross-modal matching model $\mathcal{M}$

1 initialize weight matrices W_v and W_o;
2 initialize the explorer $\mathcal{H}$ and the aggregator $\mathcal{A}$;
3 **for** *epoch from 1 to n* **do**
4 $\quad V \leftarrow \text{BERT}(\{l(v)|v \in V\})$;
5 $\quad$ Split $\mathbb{I}$ into batches $B = \{\mathbb{I}_1, ..., \mathbb{I}_m\}$;
6 $\quad$ **for** $\mathbb{I}_i \in B$ **do**
7 $\quad\quad O \leftarrow \text{SAM}(\mathbb{I}_i)$;
8 $\quad\quad G_s, O^* \leftarrow \text{Explorer}(G, O, V, W_v, W_o)$;
9 $\quad\quad Z_v, Z_o \leftarrow \text{Aggregator}(G_s, O^*, W_v, W_o)$;
10 $\quad\quad L \leftarrow \text{Loss}(Z_v, Z_o, G_s, \mathbb{I}_i, \mathcal{F})$;
11 $\quad\quad \mathcal{M} \leftarrow \text{Back-propagate}(L, \mathcal{M}, \eta)$;
12 $\mathcal{M} \leftarrow \{\mathcal{H}, \mathcal{F}, \mathcal{A}\}$;
13 **return** $\mathcal{M}$, W_v, W_o;

the architecture of our CrossETR framework, which learns matching model $\mathcal{M}$ in the following three stages: feature extraction, candidate exploration and cross-modal refinement. The detailed training procedure is illustrated in Algorithm 1.

In the feature extraction stage, CrossETR extracts object features of images $\mathbb{I}$ and initializes vertex representations of V (lines 4 and 7). We consider the feature extraction of images as an instance segmentation task [24], [28] that focuses on delineating an object with a segmentation mask in a given image. Then, a set of segmented objects $O = \{o_1, ..., o_m\}$ for each $I \in \mathbb{I}$ can be obtained by taking SAM as an image encoder. Meanwhile, we leverage BERT [29] as a textual feature extractor and regard the output of the head projection in vertex labels as the initialized vertex representations of V.

In the candidate exploration stage, CrossETR explores appropriate candidate vertices as well as their neighbors that are most likely to be associated with the images (line 8). A key challenge is to train a high-quality explorer $\mathcal{H}$ to capture those vertices that are likely to have structural semantics similar to the scene semantics of images. To this end, we propose a candidate exploration policy in Section IV, which works in the following two steps. It first obtains key objects O^* by selecting the most representative visual instances for I to reduce redundant and unnecessary computation, detailed in Section IV-A.

Secondly, it selects candidate vertices that are most relevant to O^* and adaptive samples a set of neighborhood subgraphs G_s that are most informative for these vertices by interacting with I, illustrated in Section IV-B.

In the cross-modal refinement stage, CrossETR refines vertex representations Z_v and object features Z_o leveraging the aggregator $\mathcal{A}$ (line 9). The key challenge lies in extracting image and vertex features through structural and scene contextual interactions, and then unsupervised back-propagate them to train the matching model $\mathcal{M}$ (lines 10-11). Therefore, our cross-modal refinement mechanism processes feature fusion between vertices and objects based on a multi-head cross-attention [30], which is described in Section V-A. During the training of matching model $\mathcal{M}$, CrossETR introduces a vertex-level loss and a structure-level loss into $\mathcal{F}$ guided by a pre-trained text generation model FlanT5 [31].

The following sections will mainly introduce the exploration and refinement stages of our CrossETR framework.

IV. Candidate Exploration Policy

Taking a set of images $\mathbb{I}$ as inputs, our explorer $\mathcal{H}$ designs a candidate exploration policy based on the associations between the visual instances of $\mathbb{I}$ and the vertices in V. The goal of exploration is to sample a set of neighborhood subgraph G_s for candidate vertices of image $I \in \mathbb{I}$, so that CrossETR can capture the structural semantics related to I. To achieve this purpose, the explorer is required to address two problems: (1) determine which are the key instances for each image to reduce redundancy, as presented in Section IV-A. (2) Decide how to select candidate vertices relevant to the image and sample informative neighbors to represent the structural semantics of these vertices, as described in Section IV-B.

A. Visual Instance Selection

Although the instances O segmented by the segmentation model from the image $I \in \mathbb{I}$ are sufficient because they are separated from the surroundings and backgrounds compared to those detected by object detection models [23], [24]. It is still insufficient for our task since some of the segmented instances are highly overlapped, which leads to excessive correlation calculations and introducing redundant information in capturing the scene semantics of I. In this subsection, we propose a *visual instance selection* method to choose the most important instances, which are representative objects of the image and have low overlapping area with other objects.

Motivation. As multiple segmented instances with highly overlapping bounding boxes may refer to the same object in the image, it is necessary to reduce redundancy by considering these instances. The common practice is to leverage non-maximum suppression (NMS) [32] to select the objects whose boxes with highest confidence scores and suppress the other overlapping boxes. However, the confidence score does not reflect the overlapping of object boxes. Considering the example shown in Figure 4(a), boxes in the solid lines represent different objects with confidence scores. There are three boxes b_1, b_2 and b_3 are highly overlapped. The standard NMS is likely to select all of them because they both have large confidence scores, but this obviously introduces a lot of redundancy. Next, we present how to select the most representative instances for an image to address this problem.

Algorithm 2: Instance Selection

Input: The instances $O = \{o_1, ..., o_n\}$ with boxes B, overlapping threshold δ, confidence threshold ϵ.
Output: A set of selected instances O^*.

1 $O^* \leftarrow \emptyset$;
2 **while** O *is not empty* **do**
3 $\quad i \leftarrow \arg\max_{i \in [1,|O|]} \{b_i[c]\}$;
4 $\quad R \leftarrow \emptyset$; $L \leftarrow \emptyset$; $overlap \leftarrow$ False;
5 $\quad$ **for** $j \in [1, |O|]$ **do**
6 $\quad\quad$ **if** $IoU(b_i, b_j) > \delta$ **then**
7 $\quad\quad\quad R \leftarrow R \cup \{b_j\}$; $L \leftarrow L \cup \{o_j\}$;
8 $\quad\quad\quad O \leftarrow O - \{o_j\}$;
9 $\quad\quad\quad overlap \leftarrow$ True;
10 $\quad$ **if** *overlap is not true* **then**
11 $\quad\quad O^* \leftarrow O^* \cup \{o_j\}$;
12 $\quad$ **else**
13 $\quad\quad b \leftarrow \text{Mean}(R)$;
14 $\quad\quad m \leftarrow m_k$, $k \leftarrow \arg\max_{k \in [1,|L|]} IoU(b_k, b)$;
15 $\quad\quad b[c] \leftarrow f_o(b, R) \cdot \sum_{b_k \in R} IoU(b, b_k) \cdot b_k[c]$;
16 $\quad\quad$ **if** $b[c] > \epsilon$ **then**
17 $\quad\quad\quad o \leftarrow (b, m)$;
18 $\quad\quad\quad O^* \leftarrow O^* \cup \{o\}$;
19 $\quad\quad$ **else**
20 $\quad\quad\quad O^* \leftarrow O^* \cup R$;
21 **return** O^*.

Main idea. The main idea of our method is to select instances with lower overlapping areas and higher confidence scores. Taking objects O of I as inputs, it works in the following two steps: (i) check whether an instance highly overlaps with others; and (ii) create new instances by merging these overlaps and select them if they have higher confidences.

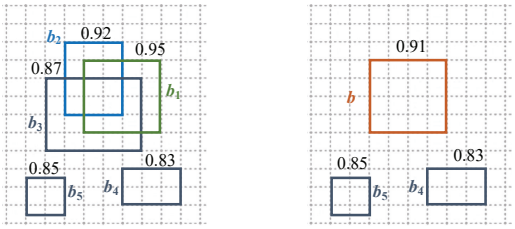
We create a new instance $o = (m, b)$ to represent these overlapping instances R by merging their boxes, and calculate its confidence score based on the Intersection-of-Union (IoU) [26] and the occupied area ratio of the boxes in R. Specifically, an occlusion factor is defined to represent the area ratio of the box b towards the overlapping of R, denoted by $f_o(b, R)$. The occlusion factor f_o associated with b and R is calculated by:

$$f_o(b, R) = \frac{1}{b[w]b[h]} \prod_{z \in \{x,y\}} (\max_{b_i \in R} b_i[z] - \min_{b_j \in R} b_j[z]). \quad (5)$$

Subsequently, we can formulate the confidence score $b[c]$ of b as follows.

$$b[c] = f_o(b, R) \sum_{b_k \in R} IoU(b, b_k) \cdot b_k[c]. \quad (6)$$

Algorithm 2 illustrates the procedure of instance selection. We first iteratively pick the bounding box b_i of object o_i with the highest score (lines 1-2) and then collect the overlapped boxes R for each o_i by checking the overlapping between two boxes using IoU (lines 5-9). Subsequently, a new instance o is



(a) Boxes of instances (b) Selected boxes after merging

Fig. 4. Example of visual instance selection.

created based on R (lines 12-20). Its box b is computed by the mean position of the overlapped boxes in R (line 13), and the feature map is set to the one that has most overlaps with the box b (line 14). The confidence score $b[c]$ of b is measured by using Equations 6 (line 15). Finally, the new created instance is selected if its confidence is larger than ϵ.

For example, Figure 4(a) shows three highly overlapped instances, depicted by their boxes b_1, b_2 and b_3. A new instance shown in the red solid line in Figure 4(b) is merged from them and denotes by $o = (b, m)$, where b is computed by the average position of boxes and m is feature map inherited from b_1 that has the largest IoU with b. Subsequently, we can select a set of boxes $\{b, b_4, b_5\}$, which is almost no repetition compared to $\{b_1, b_2, b_3, b_4, b_5\}$ obtained by the standard NMS.

The time complexity of this process mainly consists of two parts: (i) sorting the bounding boxes; and (ii) merging instances based the IoUs of redundant boxes. The time complexity of visual instance selection totally takes $O(N \log N)$ where N is the number of instances segmented from I.

B. Adaptive Subgraph Sampling

To determine how to select candidate vertices related to the image and sample informative neighbors to represent their structural semantics, we propose an adaptive subgraph sampling method in this section.

Motivation. The original GNN shown in Equations 1-2 can capture the structural semantics of vertices, which requires fully expanding their neighbors for all vertices V across layers and thereby incurs a time complexity of $O(|V||E|)$. To circumvent this barrier, sampling operations are introduced into GNNs to regulate the size of neighbors. Let $N(v_i)$ be the direct neighbors of v_i. By defining the sampling probability of a neighbor v_j for the given v_i as $p(v_j|v_i) = 1/N(v_i)$, the Equation 1 can be reformulated as follows [15]:

$$v_i^{(l+1)} = \sigma\Big(\sum_{v_j \sim p(v_j|v_i)}^{k} \alpha(v_i, v_j) v_i^{(l)} W^{(l)}\Big), \tag{7}$$

where $\sigma(\cdot)$ is an activation function, $\alpha(v_i, v_j)$ is an aggregation weight, $v_i^{(l)}$ and $W^{(l)}$ are the hidden embedding of v_i and the transformation parameter at the l-th layer, respectively. k is the sampling number for each vertex. Until now, the size of neighbors is regulated as k. Therefore, GNN with sampling operation take a time complexity of $O(k|V|)$ with $k \ll |V|$ [33], [34]. Although those methods such as PASS [33] and GCN-BS [35] successfully address the scalability issue, they do not sample neighborhood subgraphs for a set of vertices relevant to a given scene to capture their structural semantics to further support EM across images and graphs.

Main idea. Given a graph G and an image I with the selected instances O^*, our sampling aims to extract a set of subgraphs G_s from G to capture the structural semantics of the candidate vertices related to O^* of I. To do this, two issues are required to be considered: (1) how to select the candidate vertices C that are most relevant to I; and (2) how to sample the most informative neighbors for the vertices in C.

Our explorer $\mathcal{H}$ presents an *adaptive subgraph sampling* method to answer these issues by interacting with I, which works in the following two phases: (1) The first phrase is *vertex anchor selection* that picks a set of *anchors* $C \subset V$ from G based on the correlations between objects of O^* and vertices in V, with the expectation of that the model $\mathcal{M}$ can identify the most likely candidates to match with I; (2) The second phrase is *scene-aware neighbor sampling* which samples neighbors V_s for each anchor $v_i \in C$ by interacting with I, and a neighborhood subgraph is induced by V_s. To do this, the sampling probabilities of neighbors with respect to the anchors and I are computed, and then the sampled subgraphs associated with I are used to represent the structural semantics of anchors. This phase enables the model to learn the sampling strategy of candidates and their structures relevant to I. Details are shown in Algorithm 3.

First, we define a sampling graph as an *induced subgraph* of G and formally present the definition of *vertex-wise sampling*.

Definition 4: (Subgraph sampling problem) Given a graph $G = (V, E)$ and a sampling fraction ϕ, we sample a set of vertices $V_s \subset V$ such that $|V_s|/|V| = \phi$. A sampling subgraph $S = (V_s, E_s)$ is induced by the vertex set V_s and edge set consisting of all the edges that have both endpoints in V_s.

Definition 5: (Vertex-wise sampling) The sampling probability of a neighbor v_j in $N(v_i)$ for a given vertex v_i is defined as $p(v_j|v_i)$, where $N(v_i)$ is a set of direct neighbors for v_i. Each vertex samples n neighbors following the sampling distribution $P(\cdot|v_i) = \{p(v_j|v_i)|v_j \in N(v_i)\}$ at the l-th layer.

Different from layer-wise sampling approaches [34], [36] where each layer samples n neighbors following their sampling distribution. The sampling probability of v_j is defined as $p(v_j|v_1, ..., v_n)$ for a given vertex set $\{v_1, ..., v_n\}$, and they may suffer from sparse connection problem as discussed in [33]. Vertex-wise sampling selects a fixed number of vertices and allows us to control the memory footprint of the algorithm during training. At the same time, it allows the policy to explore the neighbors that are most important to capture the structural semantics related to the scene semantics of I.

Next, we present the details of each phrase in our vertex-wise sampling method.

Phase 1: vertex anchor selection. Anchors C are treated as the vertices that have the higher correlations with the objects

Algorithm 3: Sampling

Input: a graph $G = (V, E)$, an image I with objects O^* and a sampling size k.
Output: A set of sampled graphs G_s, learned projection matrices W_v and W_o.

1 $G_s \leftarrow \emptyset$;
2 $P_{ov} \leftarrow (W_o O^*) \cdot (W_v V)$;
3 $C \leftarrow \text{Top-n}_{v \in V} P_{ov}$;
4 $\mu \leftarrow \arg_{o \in O^*} \max P_{ov}$;
5 **for** $v \in C$ **do**
6 $V_s \leftarrow \{v\}$; $Queue \leftarrow \{v\}$;
7 **while** *Queue is not empty and* $|V_s| < k$ **do**
8 $v_i \leftarrow Queue.pop()$;
9 **for** $v_j \in N(v_i)$ **do**
10 $p(v_j|v_i) = \dfrac{\exp(\tau \cdot sim(W_v v_j, W_o \mu))}{\sum \exp(\tau \cdot sim(W_v v_k, W_o \mu))}$;
11 $f(V_s) = -\sum_{v_j \in V_s} p(v_j|v_i) \log p(v_j|v_i)$;
12 $g(v_j, S) = f(V_s \cup \{v_j\}) - f(V_s)$;
13 **if** $g(v_j, S) \geq \delta$ **then**
14 $V_s \leftarrow V_s \cup \{v_j\}$;
15 $Queue \leftarrow V_s \cup \{v_j\}$;
16 subgraph S is induced by V_s from G;
17 $G_s \leftarrow G_s \cup S$;
18 **return** G_s, W_v, W_o.

in O^*. We first project the feature maps of objects in O^* and attribute embeddings of vertices in V into the a uniform feature space through learnable transformation matrices W_o and W_v. The correlation coefficients between them are represented as a matrix calculated by dot product between the projected features of O^* and V, with $(|O^*| \times |V|)$ dimensions (line 2). Subsequently, those vertices with top ranked correlation coefficients are selected as anchors (line 3). Formally,

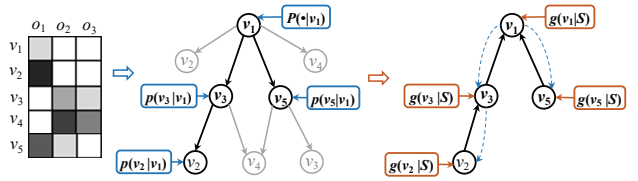
$$v_i = \arg_{v \in V} \max(W_o O^*) \cdot (W_v V). \tag{8}$$

Here $v_i \in C$ is an anchor with the highest correlation for I.

The time complexities of projection and sort in this phase are $O(|V||O^*|)$ and $O(|V||O^*| \log n)$, respectively, where n is the number of the selected anchors. With $n \ll |V|$, the totally time complexity of this phase is $O(|V||O^*|)$. The space complexity in this phase is also $O(|V||O^*|)$.

Phase 2: Scene-aware neighbor sampling. To interact with the scene semantics and reduce the disturbance of irrelevant neighbors during learning the structural semantics of anchors, we perform a scene-aware neighbor sampling.

For each anchor $v_i \in C$, $k(k = \phi \cdot |V| - 1)$ neighbors are sampled in a vertex-wise manner. Figure 5 shows an overview of neighbor sampling, which works in two steps: (i) *correlation-based sampling*, which first calculates *sampling probabilities* of neighbors with respect to v_i based on the correlations with I; and (ii) *gain-based sampling propagation*, which measures the benefits of sampled neighbors to the structural semantic representation of the induced subgraph in terms of *information gain*. Therefore, a subgraph S associated with v_i is formed by iteratively considering the most informative neighbors. And its structural semantic is updated by propagating through the graph model.



(a) Correlation-based sampling (b) Gain-based sampling propagation

Fig. 5. Scene-aware neighbor sampling consists of two steps: (a) correlation-based sampling and (b) gain-based sampling propagation.

Correlation-based sampling. As for a vertex v_i, the sampling probability distribution associated with its neighbors is denoted as $P(\cdot|v_i) = \{p(v_j|v_i)|v_j \in N(v_i)\}$ and measured by the importance scores with respect to the instances of O^*. Intuitively, the closer a neighbor is to an instance of O^*, the more important it is for exploring the scene-aware sampling subgraphs.

The sampling probability of a neighbor v_j with respect to I at the l-th is denoted by $p^{(l)}(v_j|v_i)$, which is computed by how much the correlations between v_j and the scene context of I. Treating the object that has the largest correlation with the vertices of V as the scene context of the image I, i.e., $\mu = \arg_{o \in O^*} \max(W_o O^*) \cdot (W_v V)$, we can compute $p^{(l)}(v_j|v_i)$ as follows.

$$p^{(l)}(v_j|v_i) = \frac{\exp(\tau \cdot < W_v v_j, W_o \mu >}{\sum_{v_k \in N(v_i)} \exp(\tau \cdot < W_v v_k, W_o \mu >)}, \tag{9}$$

where $< \cdot, \cdot >$ denotes the cosine similarity function and τ is a temperature hyper-parameter in range of $(0, 1]$. W_v and W_o are learnable parameters for V and O^*, respectively.

Gain-based sampling propagation. To measure the contribution of a sampled neighbor v_j for v_i to induce a sampling subgraph S, we formulate define its gain for S as follows:

$$g(v_j, S) = f(V_s \cup \{v_j\}) - f(V_s). \tag{10}$$

$$f(V_s) = -\sum_{v_j \in V_s} p^{(l)}(v_j|v_i) \log p^{(l)}(v_j|v_i), \tag{11}$$

Here $f(\cdot)$ is defined to measure how much structural information related to I the induced subgraph contains.

Sampling Algorithm. The details of sampling are illustrated in Algorithm 3. Lines 2-3 perform the vertex anchor selection to obtain a set of anchors C. For each anchor $v_i \in C$, lines 5-17 generate a set of subgraphs based on breath first search as follows. It first computes the sampling probability distribution $p(v_j|v_i)$ for each neighbor $v_j \in N(v_i)$ and then measures their gains for candidate subgraphs $f(V_s)$ induced by V_s in lines 9-12. Subsequently, the neighbors V_s with higher gains for S are treated as more important and selected in lines 13-15. It lastly returns all sampled subgraphs G_s and the learned weight matrices W_o and W_v.

Theoretical Analysis. Next, we describe how to update gradients by back-propagating through the sampling operation and then present the complexity of the sampling algorithm.

Gradient Calculation. Let θ denote the GNN parameters of transformation matrix $W^{(l)}$ and activation function σ in Equation 7. We abbreviate the neighbor sampling probability $p_\theta^{(l)}(v_j|v_i)$ to $p_{\theta_{ij}}^{(l)}$ for convenient presentation. Following previous vertex-wise sampling methods [33], [35], we can approximate the expectation of Equation 7 with $p_{\theta_{ij}}^{(l)}$ as follows:

$$v_i^{(l+1)} = \sigma_{W^(l)}(\mathbb{E}_{v_j \sim p_{\theta_{ij}}^{(l)}}[v_i^{(l)}]), l \in [0, L). \tag{12}$$

Given the training loss $\mathcal{L}$ and the hidden embedding $v_i^{(l)}$ of v_i at the l-th layer, the gradient of $\mathcal{L}$ with respect to θ of $p_{\theta_{ij}}^{(l)}$ is computed as follows:

$$\nabla_\theta \mathcal{L} = \frac{\partial \mathcal{L}}{\partial v_i^{(l+1)}} \sigma_{W^(l)} \mathbb{E}_{v_j \sim p_{\theta_{ij}}^{(l)}}[\nabla_\theta \log p_{\theta_{ij}}^{(l)} v_j^{(l)}]. \tag{13}$$

Proof. Firstly, the gradient of $v_i^{(l+1)}$ to θ can be computed as follows:

$$\begin{aligned}
\frac{\partial v_i^{(l+1)}}{\partial \theta} &= \nabla_\theta \sigma_{W^(l)}(\mathbb{E}_{v_j \sim p_{\theta_{ij}}^{(l)}}[v_j^{(l)}]) \\
&= \sigma_{W^(l)}(\nabla_\theta \sum_{k=0}^{N(v_i)} p_{\theta_{ik}}^{(l)} v_k^{(l)}) \\
&= \sigma_{W^(l)}(\sum_{k=0}^{N(v_i)} \nabla_\theta p_{\theta_{ik}}^{(l)} v_k^{(l)}) \\
&= \sigma_{W^(l)}(\sum_{k=0}^{N(v_i)} p_{\theta_{ik}}^{(l)} \cdot \nabla_\theta \log p_{\theta_{ik}}^{(l)} v_k^{(l)}) \\
&= \sigma_{W^(l)}(\mathbb{E}_{v_j \sim p_{\theta_{ij}}^{(l)}}[\nabla_\theta \log p_{\theta_{ij}}^{(l)} v_j^{(l)}]),
\end{aligned}$$

where the third equation leverages the logarithm property [37] of $\nabla_\theta a = a \cdot \nabla_\theta \log a$ to convert the sum into an expectation with respect to a.

Secondly, the gradient of $\mathcal{L}$ to θ is calculated according to chain rule as follows:

$$\begin{aligned}
\nabla_\theta \mathcal{L} &= \frac{\partial \mathcal{L}}{\partial v_i^{(l+1)}} \frac{\partial v_i^{(l+1)}}{\partial \theta} \\
&= \frac{\partial \mathcal{L}}{\partial v_i^{(l+1)}} \sigma_{W^(l)}(\mathbb{E}_{v_j \sim p_{\theta_{ij}}^{(l)}}[\nabla_\theta \log p_{\theta_{ij}}^{(l)} v_j^{(l)}]).
\end{aligned}$$

Here we omit the gradient propagation through the activation function σ for convent presentation, but there is no difference in the final form. In practice, we sum the gradients of its sampled k neighbors for an anchor v_i to train the matching model. Therefore, the hidden representations of the structural semantics for the anchors C can be propagated through the graph model with our sampling operation.

Time and Space Complexities. The time complexity of this phase is $O(|O^*|)+O(k^2 \cdot |C|)$ where k is the sampling number. The space complexity of this phase is $O(k^2|C|)$ where k^2 is the approximate size of the sampled subgraph. With $k \ll |C|$, the time and space complexities of this phase are $O(|O^*|+|C|)$ and $O(|C|)$, respectively.

Totally, our exploration policy takes $O(|O^*||V|+|O^*|+|C|)$ and $O(|O^*||V| + |C|)$ time and space complexities, respectively. Since the original GNN requires $O(|V||E|)$ time complexity to extract neighborhood subgraphs in structure representation learning. Adapting it to explore vertices matching with a given image I, $O(|O^*||V||E|)$ time complexity will be required. Our sampling theoretically outperforms the original GNN in scalability, which will be verified in Section VI-B.

V. Cross-Modal Refinement

In this section, we present how to refine entity representations to break the modality heterogeneity in Section V-A, and then propose how to learn the matching model based on the semantics of scenes and sampled subgraphs in Section V-B.

A. Cross-modal Feature Fusion

Taking the sampled subgraphs G_s and an image I with objects O^* as inputs, cross-modal feature fusion it to learn the aggregator $\mathcal{A}$ to refine the vertex and object features to break the modality heterogeneity between V_s and O^*.

As for an image I with objects O^* and a vertex v with sampled subgraph S, we first fuse the scene knowledge entailed in O^* into the vertices V_s, and then calculate the representation of v by aggregating the neighbor features to preserve its structural semantics in S.

Firstly, the scene-aware vertex representation $v_i^{c(l+1)}$ at the $(l+1)$-th layer is obtained by assigning different weights to the objects in O^* based on a multi-head attention [30] as follows.

$$v_i^{c(l+1)} = \frac{1}{K} \sum_{k=1}^{K} \sum_{o_j \in O^*} \beta_{i,j}^{(l+1)} \cdot o_j, \tag{14}$$

$$\beta_{i,j}^{(l+1)} = \frac{\exp(W_v v_i \cdot W_o o_j)}{\sum_{o_k \in O^*} \exp(W_v v_i \cdot W_o o_k)}.$$

where K is the number of attention heads. $\beta_{i,j}^{(l+1)}$ is a cross-attention coefficient of o_j to vertex v_i at the $(l+1)$-th layer that indicates how important o_j to the refinement of v_i, and computed by the dot product between v_i and o_j. Two weight matrices $W_v^{(l+1)}$ and $W_o^{(l+1)}$ are designed as shared transformations for the vertices of V and objects in O^*. After the feature fusion, the refined object representations $Z_o^{(l+1)} = \{o_i^{v(l+1)} | i \in [1, |O^*|]\}$ can also be similar extracted.

Secondly, the vertex refinement $z_i^{(l+1)}$ is calculated by aggregating the scene-aware representations v_i^c and the vertices contained in sampled subgraph S to avoid the structure knowledge of vertex itself will vanish. This is computed by expending the GNN mechanism in Equations 1-2 as follows:

$$z_i^{(l+1)} = \lambda_1 v_i^{c(l+1)} + \lambda_2 \sigma(W v_i^{(l)} + \sum_{v_j \in N(v_i)} \alpha_{i,j}^{(l+1)} W v_j^{(l+1)}), \tag{15}$$

where σ is an activation function, λ_1 and λ_2 are two scalar for the sum of scene fusion and structure attention, respectively.

$\alpha_{i,j}^{(l+1)}$ is a normalized attention coefficient of v_j to v_i at the $(l+1)$-th layer, computed as follows:

$$\alpha_{i,j}^{(l+1)} = \frac{\exp(sim(v_i, v_j))}{\sum_{v_k \in N(v_i)} \exp(sim(v_i, v_k))}.$$

Finally, we can obtain the vertex refinement representations $Z_v^{(l+1)} = \{z_i^{(l+1)} | i \in [1, |V|]\}$ for all vertices of V at the $(l+1)$-th layer.

B. Training Processing

Considering that a matching vertex-image pair is defined as the one with similar objects and vertices as well as the semantics between scenes and structures. Therefore, CrossETR introduces a vertex-level loss for object-vertex and a structure-level loss for scene-substructure to guide the training of cross-modal matching model $\mathcal{M}$.

Vertex-level loss. Like other semantic-driven entity matching approaches [9], [11], we expect that the matched objects and vertices have close representations in the feature space during training. As for a vertex v and a subgraph $S \in G_s$, the vertex-level matching loss between v and I with respect to S and O^* is inductively calculated as follows.

$$F_v(v, I) = \frac{1}{|V_s|} \sum_{v_i \in V_s} \max_{o_j \in O^*} sim(v_i, o_j) \tag{16}$$

Structure-level loss. To determine whether v and I are similar in structures, we measure the similarity between the structural semantic of v and the scene semantic of I. Firstly, the structural semantic of v is represented by a sequence T_v serialized by the sampled subgraph S. Some existing graph serialization methods [3], [10] can be used to create sequence for v with S. Secondly, the scene semantic of I is regarded as an image-conditioned text T_I generated by the pre-trained large model FlanT5 [31], which is a decoder to generate text description for a given image following [38]. Therefore, a structure-level loss with respect to v and I is measured by the representations of T_v and T_I as $F_s(v, I) = sim(T_v, T_I)$, where $sim(\cdot, \cdot)$ is the cosine similarity function.

To introduce the two parts into the training process of CrossETR, we integrate them into a triplet-wise ranking objectives [14] to encourage the matched images and vertices to be closed and unmatched ones to be separated in the embedding spaces. During training the similarity between v and I is computed by $F(v, I) = F_v(v, I) + F_s(v, I)$ and the loss is measured as follows:

$$L_s(v, I) = \sum_{i=1}^{b} \max(0, \delta - F(v_i, I_i) + F(v_i', I_i)) \tag{17}$$

where δ is a margin value, b is the size of a batch examples. (v_i, I_i) is a matched pair of image and vertex that has the highest matching score. (v_i', I_i) is a negative example generated by randomly sample a vertex from the batch.

TABLE II
DATASET STATISTICS.

Datasets	# Vertices	# Triplets	# Images	# Pairs
WN18-IMG [7]	41,105	93,003	70,349	2891M
FB15K-IMG [40]	14,541	310,116	145,944	2122.2M
OpenImages [39]	600	374,768	9,178,275	5506.9M

VI. EXPERIMENT

Using three publicly available datasets, we experimentally evaluate the proposed method CrossETR by conducting extensive experiments in the following three aspects: (i) the overall performance of CrossETR in effectiveness and scalability compared to state-of-the-art baselines; (ii) ablation study for different modules of CrossETR in terms of effectiveness and efficiency; and (iii) a case study of multi-modal knowledge graph integration.

A. Experiment Setting

Datasets. We utilize three publicly available datasets OpenImages [39], WN18-IMG [7] and FB15K-237-IMG [40]. (i) OpenImages [39] is a large-scale dataset, which provides a large number of examples for object bounding boxes and object segmentation. It consists of a large subset of Freebase [17] and Google knowledge graph, and a set of diverse images including complex scenes containing several objects (8.3 per image on average) with totally 3,290,070 objects. (ii) WN18-IMG [7] is an extended dataset of WN18 [18] with 10 images for each entity, where WN18 is a knowledge graph originally extracted from WordNet. (iii) FB15K-IMG [40] consists of a subset of the large-scale knowledge graph Freebase [17] and a set of images associated with the entities in Freebase, which totally includes 2,122M entity pairs. It is a popular dataset in multi-modal knowledge completion. We derive three subset datasets from this to verify the efficiency and scalability of our methods namely FB-IMG-1, FB-IMG-2, FB-IMG-3 and FB-IMG-4, which include 154M (million), 616M, 924M and 1386M entity pairs, respectively. Detailed statistics are shown in Table II.

Metrics. To evaluate the performances of CrossETR quantitatively, we use the following two metrics: (i) Hits@k ($k \in \{1, 3, 5\}$) and MRR (Mean Reciprocal Rank) are used to measure the accuracy of entity matching, where Hits@k measures the number of correct results within the top-k predictions. MRR measures the average reciprocal ranks of the top k results i.e., MRR = $\frac{1}{n}\sum_{i=1}^{n} 1/rank_i$. Higher Hits@$k$ and MRR indicate better performance. (ii) Running time in hour is used to evaluate the training efficiency on ablation study and measure the scalability on different data size. The running time means the average training time of each epoch in every approach. The processing time of feature extraction is excluded for all methods to make a fair comparison.

Baseline comparisons. To verify the performance of our proposed method, we typically compare to two types of state-of-the-art cross-modal matching approaches for a comprehensive evaluation. (1) Dual-based approaches, which directly measure the distance of cross-modal representations,

as illustrated in Figure 2(a). These consist of two parts: (i) famous models such as CLIP [12] and ALIGN [41]; and (ii) unified matching methods perform semantic-driven EM in a uniform modality, which are naive designed for describing the necessity of our proposed method. (2) Fusion-based approaches which map multi-modal data into a common feature space as described in Figure 2(b), including VisualBERT [42], ViLBERT [43], IMRAM [14] and TransAE [44] .

Dual-based approaches: Firstly, starting from the unified matching methods: String matching encodes the matching of vertices and images as a string semantic matching problem that measures the similarity between graph sequences [3] and image captions [31] by using pre-trained language model [29]. Graph matching converts our problem as a semantically subgraph matching problem. We first generate the scene graphs of images based on Faster RCNN [26], and then perform subgraph matching [45] on these scene graphs and the knowledge graph. We relax it to only consider the similarity between vertices and objects. Secondly, CLIP [12] is to contrastively learn a transferable language-image pre-training model from natural language supervision by pairing images with relevant language descriptions. ALIGN [41] trains large-scale models using large amounts of noisy text data to scale up vision-language representation learning for various tasks such as image classification.

Fusion-based approaches: VisualBERT [42] consists of a stack of Transformer layers that implicitly align elements of an input text and regions in an associated input image with self-attention mechanism. ViLBERT [43] is a pre-trained visual-language model that processes both visual and textual inputs in separate streams, and then interacts through co-attention transformer layers for learning joint representations. IMRAM [14] utilizes recurrent attention memory to integrate information from both text and image modalities for cross-modal image-text retrieval. TransAE [44] combines multi-modal auto-encoder with TransE to encode the visual and textual knowledge into the unified representation, where the hidden layer of the auto-encoder is regarded as entity representations in the TransE model.

To make a fair comparison, we modify these model by formatting sequence graph into texts using some graph serialization methods. For the pre-trained methods, we use public models to predict the results on test dataset. For the last two methods, we leverage the source code from the original repositories to produce the results.

To describe the contributions of different modules, we design variants to conduct ablation study in the following aspects. (i) For visual instance merging proposed in Section IV-A, there are two variants by removing the merging or replacing it with the original NMS. (ii) For scene-aware subgraph sampling presented in Section IV-B, we derive six variants: (a) removing sampling from CrossETR; and (b) replacing it with the state-of-the-art sampling methods: GraghSage [15], GCN-BS [35], PASS [33], FastGCN [36], where the first four are vertex-wise sampling methods that use a uniform sampling, a variance reduced sampling based on multi-armed bandits and an adaptive gradient-propagated sampling, respectively. The last is a layer-wise sampling method with independent-identical-distributions in every layer. For fair comparison, all methods share the same network structure with two GCN layer and 64 hidden dimensions. (iii) For training objective introduced in Section IV-B, there are two variants derived by removing the structure-level loss F_s and the object-level loss F_v in CrossETR, respectively.

To describe the application of our cross-modal EM task, we perform a case study to illustrate the advantage of our methods in multi-modal knowledge graph integration by comparing with the following approaches, i.e., RotatE [46], PairRE [47], MKGformer [7], OTKGE [48], MoSE [49], IMF [50], ANALOGY [51], ComplEx-N3 [52], RSME [8], TransAE [44] as mentioned in VISTA [53].

Implementation details. We implement our methods in PyTorch [54] and Huggingface [55]. Unless particularly specified, CrossETR is trained using the Adam [56] optimizer with a learning rate of 0.0001 for all experiments. The three models BERT [29], SAM [24] and FlanT5 [31] are employed as the pre-trained language model, segmentation model in feature extraction stage and image-conditioned text generation module in refinement stage, respectively. The default sample size n of neighbors is set to 50. For each layer of the cross-modal feature fusion module, the hidden size and head number of cross-attention are set to 128 and 8, respectively. We fix the projection dimension of text and image to 768 and 512, respectively. The batch size is set to 32, the number of epochs is set to 20. All experiments are conducted on a machine with an Intel Core i9-10900K CPU, a NVIDIA GeForce RTX3090 GPU with 24GB memory. We report the results of all competitors in their optimal settings.

B. Experiment Results

Exp-1: Accuracy of entity matching. To demonstrate the accuracy of our proposed method CrossETR, we compare it with the eight competitors mentioned above on three public datasets.

From Table III, we have the following three findings. (1) Overall, CrossETR mostly outperforms other competitors on the three datasets, followed by CLIP and IMRAM. There are average 5.28% (from 34.79% to 40.07%) and 0.065 (from 0.287 to 0.352) improvements on three datasets in Hits@3 and MRR values, respectively. It verifies the superiority of our framework, especially on the OpenImages data with complex scenes. (2) As for the dual-based methods, the two naive modal unification based baselines are poorly effects. This is reasonable that the structures of vertices and the scenes of images are destroyed during modal unification, resulting in the inability to effectively represent their semantics for matching. On the other hand, the famous dual-based model CLIP is more competitive with ours than other methods. (3) As for the fusion-based methods, the accuracy performance is significantly different from ours since they do not consider the structure and scene semantics of entities to fuse features.

TABLE III
OVERALL ACCURACY ON DIFFERENT DATASETS.

	WN18-IMG				**OpenImages**				**FB15K-IMG**			
Methods	Hits@1	Hits@3	Hits@5	MRR	Hits@1	Hits@3	Hits@5	MRR	Hits@1	Hits@3	Hits@5	MRR
String matching	0.34	0.98	1.36	0.017	0.72	1.33	2.47	0.011	1.01	1.42	3.54	0.017
Graph matching	2.58	8.93	18.45	0.079	2.03	4.86	8.21	0.039	2.38	6.05	16.67	0.068
ALIGN [41]	22.69	29.45	32.70	0.237	19.79	22.3	29.7	0.219	24.51	35.23	40.28	0.321
CLIP [12]	**27.38**	34.17	38.07	0.291	27.83	32.4	38.45	0.283	26.06	37.81	41.91	0.287
VisualBERT [42]	2.39	6.07	9.32	0.195	8.32	15.54	21.32	0.176	21.70	32.40	43.90	0.273
ViLBERT [43]	2.43	7.13	12.46	0.284	12.73	27.29	33.12	0.233	23.30	33.50	45.70	0.268
TransAE [44]	0.53	10.54	20.72	0.075	13.8	16.47	24.92	0.149	19.80	37.60	44.10	0.352
IMRAM [14]	8.75	14.22	23.75	0.124	17.80	22.67	28.32	0.192	24.80	39.61	47.80	0.391
CrossETR	26.72	**38.20**	**45.71**	**0.328**	**29.71**	**36.8**	**42.43**	**0.334**	**27.91**	**45.21**	**62.37**	**0.394**

TABLE IV
ABLATION STUDY FOR DIFFERENT COMPONENTS OF CROSSETR.

	WN18-IMG				**OpenImages**				**FB15K-IMG**			
Methods	Hits@1	Hits@5	MRR	Time	Hits@1	Hits@5	MRR	Time	Hits@1	Hits@5	MRR	Time
CrossETR w/o VIM	25.33	42.01	0.278	4.05	26.36	38.72	0.295	0.59	23.42	53.98	0.251	2.26
CrossETR w/ NMS	25.86	44.28	0.281	3.46	27.68	41.12	0.307	0.32	24.96	57.64	0.323	2.19
w/o Sampling	22.16	38.92	0.249	4.29	26.03	35.49	0.286	0.53	18.29	34.45	0.214	3.83
w/ FastGCN [36]	22.98	38.36	0.231	3.16	26.79	36.16	0.277	0.44	20.24	42.37	0.239	3.27
w/ GraghSage [15]	23.17	40.08	0.238	3.22	26.54	36.10	0.271	0.47	21.98	44.71	0.254	3.19
w/ GCN-BS [35]	23.04	39.77	0.228	3.09	27.91	40.23	0.298	0.38	24.31	47.75	0.287	2.86
w/ PASS [33]	25.32	42.74	0.261	3.26	28.41	40.94	0.312	0.32	24.77	50.39	0.302	2.74
CrossETR w/ F_v	24.36	41.27	0.259	2.76	28.03	39.74	0.281	0.29	25.73	54.48	0.269	1.95
CrossETR w/ F_s	21.38	37.98	0.247	**2.39**	24.67	35.82	0.259	0.28	23.65	47.98	0.257	**1.81**
CrossETR (full)	**26.72**	**45.71**	**0.328**	2.71	**29.71**	**42.43**	**0.334**	**0.27**	**27.91**	**62.37**	**0.394**	1.89

TABLE V
SCALABILITY IN DIFFERENT DATA SIZE OF FB15K-237-IMG.

Data sizes	154M	616M	924M	1386M	2122M
Hits@5	50.73	51.26	54.89	58.02	62.37
MRR	0.267	0.278	0.306	0.364	0.394
Time	0.12	0.56	0.71	1.52	1.89

Exp-2: Scalability in different data size. To investigate the scalability of our proposed method, we evaluate on the FB15K-IMG dataset with different scales of 154M, 616M, 924M, 1386M and 2122M vertex-image pairs.

Table V reports the accuracy (via Hits@5 and MRR value) and training time. We can see that as the data size increases, CrossETR has higher accuracy, and the training time linearly increases. This is because: (i) the time complexities of exploration and refinement stages are both proportional to the size of entity pairs, as discussed in Section IV-B. (ii) With the increasing of data sizes during training, the sampling policy learns how to sample vertices and subgraphs that have the most relevant structural semantics to the image.

Exp-3: Ablation study. We further conduct ablation study to verify the matching accuracy and training efficiency of different modules in CrossETR. We compare CrossETR with two visual instance merging (VIM) designs (i.e., w/o VIM and w/ NMS), five sampling variants (i.e., w/o sampling and w/ four methods mentioned above) and two loss function modifies (i.e., w F_v and w F_s).

Overall, we have the following findings from Table IV. (1) Compared with other variants, CrossETR requires less training time but achieves higher accuracy. (2) CrossETR outperforms other variants on matching accuracy in the three datasets, followed by CrossETR w/ NMS and CrossETR w/o F_s. They are average 2.49% (from 47.68% to 50.17%) and 5.01% (from 45.16% to 50.17%) improvements in Hits@3 value, and average 0.05 (from 0.303 to 0.352) and 0.083 (from 0.269 to 0.352) improvements in MRR value on three datasets, respectively. (3) CrossETR takes almost the least training time (following CrossETR w/o F_s), with training time about 1.5 times to the worst variant on the FB15K-IMG and WN18-IMG datasets, and about 2 times on the OpenImages.

The effect of visual instance merging. As shown in the 1-st and 2-nd rows, CrossETR w/ NMS has more close accuracy but takes too much time on all different datasets compared to CrossETR. It is because NMS does not sufficiently reduce instance redundancy and overlapping instances introduce excessive computations to further reduce training speed. This verifies the advantage of our proposed VIM.

The effect of subgraph sampling. From the 3-rd to 7-th rows, we can see that (i) CrossETR w/o sampling has the worst performance in accuracy and training time, which indicates that our sampling operation can improve the training efficiency, as discussed in Section IV-B. (ii) CrossETR w/ PASS has more close accuracy but takes too much time on all different datasets compared to CrossETR. PASS samples graph structures that are most informative to training objectives in an importance-based manner. However, CrossETR takes clearly less training time than CrossETR w/ PASS because the vertices that have most likely related image are early extracted in anchor selection stage.

The effect of loss function. As shown the 8-th and 9-th rows, we have the following two findings: (i) CrossETR w/ F_v has more closer accuracy than CrossETR w/ F_s to CrossETR because object-level loss considers the semantic similarity between vertices and objects; (ii) CrossETR w/ F_s takes more less training time followed by CrossETR and CrossETR w/ F_v.

TABLE VI
PERFORMANCE OF MULTI-MODAL KNOWLEDGE GRAPH INTEGRATION.

Methods	Hits@1	Hits@3	Hits@10	MRR
RotatE [46]	21.83	34.33	49.32	0.310
PairRE [47]	23.99	36.75	51.93	0.333
MKGformer [7]	22.78	33.56	47.40	0.310
OTKGE [48]	25.11	37.39	51.92	0.341
MoSE [49]	23.84	35.32	49.65	0.325
IMF [50]	27.35	40.40	55.73	0.368
ANALOGY [51]	15.16	26.77	43.01	0.242
ComplEx-N3 [52]	25.84	38.47	53.91	0.351
RSME [8]	24.2	34.43	46.70	0.345
TransAE [44]	14.32	22.67	34.59	0.212
VISTA [53]	26.73	41.58	57.18	0.381
CrossETR	**27.91**	**45.21**	**62.37**	**0.394**

These verify the advantage of our proposed structure-level and object-level losses in CrossETR.

Exp-4: Case study. Table VI presents the advantage of our cross-modal EM over multi-modal knowledge graph integration on the FB15K-237-IMG dataset. We can see that our proposed methods outperform other state-of-the-art approaches. For example, CrossETR improves 1.18%, 5.19% and 0.013 in Hits@1, Hits@10 and MRR values over the excellent VISTA, respectively. This demonstrates that cross-modal EM can benefit various downstream tasks such as multi-modal knowledge graph integration.

VII. RELATED WORK

We categorize the related work as follows.

Heterogeneous Entity Matching. Entity Matching (EM) is one of the fundamental and significant tasks in data management, and has been widely studied in the data management and machine learning communities. Heterogeneous entity matching is to link those entities with different data formats. JedAI [57] considers RDF and CSV by first converting entities to a set of name-value pairs, and then checking their labels and attributes. PathSim [58] extends SimRank to measure similarity of entities via topological matching under a meta path framework. MAGNN [59] combines graph neural network with meta-paths to extract embeddings and measure vertex similarity. Generalized EM [10], [60] usually converts different sources into a unify data format. TDmatch [60] performs relational table and text document matching in an unsupervised learning way via graph creation and random walk. HER [61] present a parametric simulation method for linking entities across relations and graphs. It first converts heterogeneous data to a canonical graph by direct mapping, and then links entities based on inductively topology matching. Subsequently, [3] presents a GNN-based method to integrate semantically related relational tuples, JSON keys and graph vertices, which first encodes various data into canonical graphs and then represents entities based on their attributes and structure representations. Machamp [62] benchmarks the GEM for different types such as structured tables, semi-structured, or textual data. Recently, representation learning technology has been used widely in EM task and achieved promising performance [9], [10]. For example, PromptEM [10] unifies heterogeneous data as textual sequences and designs specific prompt-tuning to transform GEM as a masked language task to predict target words in a low-resource setting.

Unlike previous works, we study entity matching across unstructured images and semi-structured graph not only relational data. Considering that the scene semantics of images and the structural semantics of graph, we perform semantic-driven entity matching across images and vertices by parsing object-vertex and scene-structure semantics. It is different from existing entity matching in that it requires address cross-modal candidate exploration and feature refinement.

Cross-modal Matching. Cross-modal matching approaches mostly are designed for visual and textual data to retrieve a set of related texts or images from another modalities. The key is to learn a comprehensive and unified representation for various data with different modalities. It can be typically divided into two categories: dual encoder methods that directly measure the distances of cross-modal representations such as CLIP [12], ALIGN [41], and fusion encoder methods that map these data into a common space via attention mechanism or generative adversarial network such as VisualBERT [42], ViLBERT [43], IMRAM [14]. Recently, multi-modal large models have received widespread attention and large-scale pre-trained encoders, e.g., CLIP [12], ALIGN [41] and Flamingo [63] have shown their superiority in cross-modal retrieval tasks.

These approaches work well for texts and images, but directly extending them to our task has limited by the modality heterogeneity between graph and image data. Some works attempt to integrate graph and image, such as multi-modal knowledge graph completion [7], which mainly focuses on link prediction, relation extraction and entity recognition. These are orthogonal to our work. We aims to perform cross-modal entity matching by considering the scene semantics of images and the structural semantics of graph.

VIII. CONCLUSION

In this paper, we address the problem of entity matching across images and graph. To alleviate the issue of training efficiency in current semantically matching solutions, we propose a novel semantic-driven framework, namely CrossETR. CrossETR follows an exploration-then-refinement paradigm that first explores candidates based on entity correlations and then performs cross-modal refinement. In exploration stage, it aims to boost the training efficiency through candidate exploration policy. It reduces redundant objects of images by early filtering out irrelevant candidate pairs during training and captures structural semantics by sampling the most informative neighboring subgraphs. In the refinement stage, it bridges the modal heterogeneity via the cross-modal feature fusion, and refines entity associations in an unsupervised training mechanism to address the semantic-driven entity matching. Extensive experimental evaluation on three real-world benchmarks verify the superiority of our proposed CrossETR in terms of effectiveness and efficiency compared with some state-of-the-art approaches. In the feature, we plan to extend our work to support more data management tasks such as data cleaning in a general framework.

REFERENCES

[1] Y. Li, J. Li, Y. Suhara, A. Doan, and W. Tan, "Deep entity matching with pre-trained language models," *Proc. VLDB Endow.*, vol. 14, no. 1, pp. 50–60, 2020.

[2] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, and V. Raghavendra, "Deep learning for entity matching: A design space exploration," in *Proceedings of the 2018 international conference on management of data*, 2018, pp. 19–34.

[3] Q. Yuan, Y. Yuan, Z. Wen, H. Wang, C. Chen, and G. Wang, "Exploring heterogeneous data lake based on unified canonical graphs," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1834–1838.

[4] Q. Yuan, Y. Yuan, Z. Wen, H. Wang, and S. Tang, "An effective framework for enhancing query answering in a heterogeneous data lake," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 770–780.

[5] X. Zhu, Z. Li, X. Wang, X. Jiang, P. Sun, X. Wang, Y. Xiao, and N. J. Yuan, "Multi-modal knowledge graph construction and application: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 2, pp. 715–735, 2022.

[6] Z. Chen, Y. Zhang, Y. Fang, Y. Geng, L. Guo, X. Chen, Q. Li, W. Zhang, J. Chen, Y. Zhu *et al.*, "Knowledge graphs meet multi-modal learning: A comprehensive survey," *arXiv preprint arXiv:2402.05391*, 2024.

[7] X. Chen, N. Zhang, L. Li, S. Deng, C. Tan, C. Xu, F. Huang, L. Si, and H. Chen, "Hybrid transformer with multi-level fusion for multimodal knowledge graph completion," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 904–915.

[8] M. Wang, S. Wang, H. Yang, Z. Zhang, X. Chen, and G. Qi, "Is visual context really helpful for knowledge graph? a representation learning perspective," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 2735–2743.

[9] R. Wang, Y. Li, and J. Wang, "Sudowoodo: Contrastive self-supervised learning for multi-purpose data integration and preparation," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 1502–1515.

[10] P. Wang, X. Zeng, L. Chen, F. Ye, Y. Mao, J. Zhu, and Y. Gao, "Promptem: prompt-tuning for low-resource generalized entity matching," *arXiv preprint arXiv:2207.04802*, 2022.

[11] Z. Zhong, M. Zhang, J. Fan, and C. Dou, "Semantics driven embedding learning for effective entity alignment," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 2127–2140.

[12] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[13] P. Xu, X. Zhu, and D. A. Clifton, "Multimodal learning with transformers: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 12 113–12 132, 2023.

[14] H. Chen, G. Ding, X. Liu, Z. Lin, J. Liu, and J. Han, "Imram: Iterative matching with recurrent attention memory for cross-modal image-text retrieval," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 655–12 663.

[15] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[16] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp. 57–81, 2020.

[17] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.

[18] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[19] P. R. Asveld, "Generating all permutations by context-free grammars in greibach normal form," *Theoretical computer science*, vol. 409, no. 3, pp. 565–577, 2008.

[20] Y. Gao, T.-H. Huang, and R. J. Passonneau, "Abcd: A graph framework to convert complex sentences to a covering set of simple sentences," *arXiv preprint arXiv:2106.12027*, 2021.

[21] X. Shang, T. Ren, J. Guo, H. Zhang, and T.-S. Chua, "Video visual relation detection," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1300–1308.

[22] X. Sun, T. Ren, Y. Zi, and G. Wu, "Video visual relation detection via multi-modal feature fusion," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2657–2661.

[23] R. Sharma, M. Saqib, C. Lin, and M. Blumenstein, "A survey on object instance segmentation," *SN Computer Science*, vol. 3, no. 6, p. 499, 2022.

[24] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[25] C. Ge, X. Liu, L. Chen, B. Zheng, and Y. Gao, "Largeea: Aligning entities for large-scale knowledge graphs," *Proc. VLDB Endow.*, vol. 15, no. 2, pp. 237–245, 2021.

[26] G. Zhu, L. Zhang, Y. Jiang, Y. Dang, H. Hou, P. Shen, M. Feng, X. Zhao, Q. Miao, S. A. A. Shah *et al.*, "Scene graph generation: A comprehensive survey," *arXiv preprint arXiv:2201.00443*, 2022.

[27] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, "A comprehensive survey of scene graphs: Generation and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1–26, 2021.

[28] H. Yuan, X. Li, C. Zhou, Y. Li, K. Chen, and C. C. Loy, "Open-vocabulary sam: Segment and recognize twenty-thousand classes interactively," *arXiv preprint arXiv:2401.02955*, 2024.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

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

[31] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, "Scaling instruction-finetuned language models," *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.

[32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[33] M. Yoon, T. Gervet, B. Shi, S. Niu, Q. He, and J. Yang, "Performance-adaptive sampling strategy towards fast and accurate graph neural networks," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2046–2056.

[34] W. Huang, T. Zhang, Y. Rong, and J. Huang, "Adaptive sampling towards fast graph representation learning," *Advances in neural information processing systems*, vol. 31, 2018.

[35] Z. Liu, Z. Wu, Z. Zhang, J. Zhou, S. Yang, L. Song, and Y. Qi, "Bandit samplers for training graph neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6878–6888, 2020.

[36] J. Chen, T. Ma, and C. Xiao, "Fastgcn: Fast learning with graph convolutional networks via importance sampling," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[37] A. Gibbons, *Algorithmic graph theory*. Cambridge university press, 1985.

[38] D. Ghosal, N. Majumder, R. K.-W. Lee, R. Mihalcea, and S. Poria, "Language guided visual question answering: Elevate your multimodal language model using knowledge-enriched prompts," *arXiv preprint arXiv:2310.20159*, 2023.

[39] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5410–5419.

[40] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[41] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International conference on machine learning*. PMLR, 2021, pp. 4904–4916.

[42] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, 2019.

[43] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *Advances in neural information processing systems*, vol. 32, 2019.

[44] Z. Wang, L. Li, Q. Li, and D. Zeng, "Multimodal data enhanced representation learning for knowledge graphs," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[45] H. Wang and C. C. Aggarwal, *A Survey of Algorithms for Keyword Search on Graph Data*. Managing and Mining Graph Data, 2010.

[46] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[47] L. Chao, J. He, T. Wang, and W. Chu, "Pairre: Knowledge graph embeddings via paired relation vectors," *arXiv preprint arXiv:2011.03798*, 2020.

[48] Z. Cao, Q. Xu, Z. Yang, Y. He, X. Cao, and Q. Huang, "Otkge: Multimodal knowledge graph embeddings via optimal transport," *Advances in Neural Information Processing Systems*, vol. 35, pp. 39 090–39 102, 2022.

[49] Y. Zhao, X. Cai, Y. Wu, H. Zhang, Y. Zhang, G. Zhao, and N. Jiang, "Mose: Modality split and ensemble for multimodal knowledge graph completion," *arXiv preprint arXiv:2210.08821*, 2022.

[50] X. Li, X. Zhao, J. Xu, Y. Zhang, and C. Xing, "Imf: interactive multimodal fusion model for link prediction," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2572–2580.

[51] H. Liu, Y. Wu, and Y. Yang, "Analogical inference for multi-relational embeddings," in *International conference on machine learning*. PMLR, 2017, pp. 2168–2178.

[52] T. Lacroix, N. Usunier, and G. Obozinski, "Canonical tensor decomposition for knowledge base completion," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2863–2872.

[53] J. Lee, C. Chung, H. Lee, S. Jo, and J. Whang, "Vista: Visual-textual knowledge graph representation learning," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 7314–7328.

[54] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 8024–8035.

[55] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, Q. Liu and D. Schlangen, Eds. Association for Computational Linguistics, 2020, pp. 38–45.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[57] G. Papadakis, L. Tsekouras, E. Thanos, G. Giannakopoulos, T. Palpanas, and M. Koubarakis, "The return of jedai: End-to-end entity resolution for structured and semi-structured data," *Proceedings of the VLDB Endowment, Vol. 11, No. 12*, vol. 11, no. 12, pp. 1950–1953, 2018.

[58] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[59] X. Fu, J. Zhang, Z. Meng, and I. King, "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proceedings of The Web Conference 2020*, 2020, pp. 2331–2341.

[60] N. Ahmadi, H. Sand, and P. Papotti, "Unsupervised matching of data and text," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 1058–1070.

[61] W. Fan, L. Geng, R. Jin, P. Lu, R. Tugay, and W. Yu, "Linking entities across relations and graphs," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 634–647.

[62] J. Wang, Y. Li, and W. Hirota, "Machamp: A generalized entity matching benchmark," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 4633–4642.

[63] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 716–23 736, 2022.