arXiv:2302.00796v2 [cs.IR] 8 Feb 2023

时序知识图谱的无监督实体对齐

Xiaoze Liu 浙江大学 中国杭州
xiaoze@zju.edu.cn

Junyang Wu
浙江大学
中国杭州
wujunyang@zju.edu.cn

Tianyi Li
奥尔堡大学
奥尔堡，丹麦
tianyi@cs.aau.dk

Lu Chen 浙江大学 杭州，中国
luchen@zju.edu.cn

Yunjun Gao 浙江大学 杭州，中国
gaoyj@zju.edu.cn

摘要

实体对齐（EA）是一项基础的数据集成任务，旨在识别不同知识图谱（KGs）中的等价实体。时序知识图谱（TKGs）通过引入时间戳扩展了传统知识图谱，近年来受到越来越多的关注。最先进的时间感知实体对齐研究指出，时序知识图谱中的时间信息有助于提升实体对齐的性能。然而，现有研究尚未充分利用时序知识图谱中时间信息的优势。此外，这些研究通过预先对齐实体对来执行实体对齐，这一过程可能耗费大量人力，因而效率低下。本文提出 DualMatch，以有效融合实体对齐所需的关系信息和时间信息。DualMatch 将时序知识图谱上的实体对齐转化为加权图匹配问题。具体而言，DualMatch 配备了一种无监督方法，无需种子对齐即可实现实体对齐。DualMatch 包含两个步骤：（i）使用一种新颖的无标签编码器 Dual-Encoder，分别将时间信息和关系信息编码为嵌入
以及（ii）融合两方面的信息，并利用一种新颖的基于图匹配的解码器 GM-Decoder 将其转换为对齐结果。DualMatch 能够在有监督或无监督的情况下对时序知识图谱进行实体对齐，这得益于其有效捕获时间信息的能力。在三个真实世界时序知识图谱数据集上进行的大量实验表明，DualMatch 的表现显著优于最先进的方法。

关键词

实体对齐、知识图谱、无监督学习

ACM 引用格式：
Xiaoze Liu、Junyang Wu、Tianyi Li、Lu Chen 和 Yunjun Gao。2023年。《时序知识图谱的无监督实体对齐》。载于*2023*年 *ACM Web* 大会论文集（*WWW '23*），*2023*年*5*月*1*日至*5*日，美国德克萨斯州奥斯汀。ACM，纽约，美国，11页。https://doi.org/10.1145/3543507.3583381

1 引言

知识图谱（KGs）表示与现实世界对象相关的结构化知识，在许多现实世界应用中发挥着至关重要的作用，例如语义搜索 [52],实体 [18, 19, 58, 59] 和

允许为个人或课堂使用而免费制作本作品全部或部分内容的数字或纸质副本，前提是不得为营利或商业利益而制作或分发副本，并且副本首页须附上本声明和完整引文。必须尊重本作品中由作者以外的其他方拥有版权的组件。注明出处后，可以摘录。若要以其他方式复制、重新发表、发布到服务器或分发至列表，则需事先获得具体许可和/或支付费用。请向 permissions@acm.org 请求许可。*WWW '23*，*2023*年*5*月*1*日至*5*日，美国德克萨斯州奥斯汀© 2023年版权归所有者/作者所有。出版权授权给 ACM。ACM ISBN 978-1-4503-9416-1/23/04...$15.00
https://doi.org/10.1145/3543507.3583381

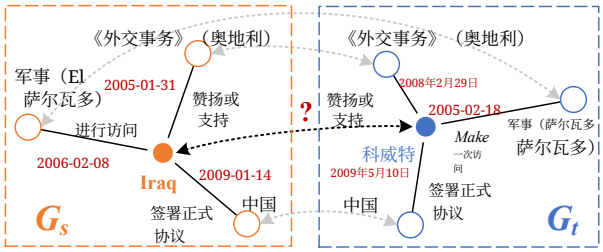

图1：未考虑时序信息时现有方法局限性的示例

关系抽取 [14–16, 28]。然而，现有的知识图谱通常非常不完整 [44]。由于不同的知识图谱由各种数据源构建而成，它们包含独特的信息，但也存在重叠的实体。这种情况为我们提供了利用重叠实体整合不同知识图谱的机会。

知识图谱整合的一种典型策略是**实体对齐**（实体对齐） [44], ，即对齐来自不同知识图谱、指向同一现实世界对象的实体。给定两个知识图谱和一小组预先对齐的实体（也称为种子对齐），实体对齐旨在识别它们之间所有可能的对齐关系。现有的基于嵌入的研究 [23, 31, 34, 40,43] 已证明，图神经网络（GNN） [12, 20, 46] 的应用显著提升了实体对齐的效果。此类研究假设，分属不同知识图谱的两个等价实体，其邻居也彼此等价 [29]。基于这一假设，它们通过对知识图谱进行表示学习来对齐实体。我们将基于嵌入的实体对齐过程概括为以下两个步骤：（i）使用一些种子对齐训练实体对齐模型，将两个知识图谱中的实体编码为嵌入向量；（ii）根据嵌入之间的特定相似度度量（例如余弦相似度），将实体的嵌入向量解码为对齐矩阵。

大多数现有的实体对齐研究都需要在训练嵌入模型之前预先对齐训练集，这不仅耗费大量人力，也降低了它们在现实世界应用中的实用性 [44]。通过纳入侧信息，基于名称的 [8–10, 29, 32, 48, 57] 或基于图像的 [27] 模型可以解决这些问题，因为执行实体对齐时所需的标签很少，甚至无需标签。然而，这类模型仍存在一些问题。首先，由于名称偏差，基于名称的模型的性能被高估了 [3, 27, 29, 39]。在这种情况下，真实标注（跨图/语言链接）通常是根据实体名称生成的，因此当它们被用作特征时会导致测试数据泄漏。其次，图像并非知识图谱的固有组成部分，因此难以收集。因此，纳入侧信息未必能提升实体对齐效果。此外，大多数现有的实体对齐方法 [2–4, 23, 27, 29–34] 都没有考虑时间信息。这会导致对实体作出错误预测，而这些实体

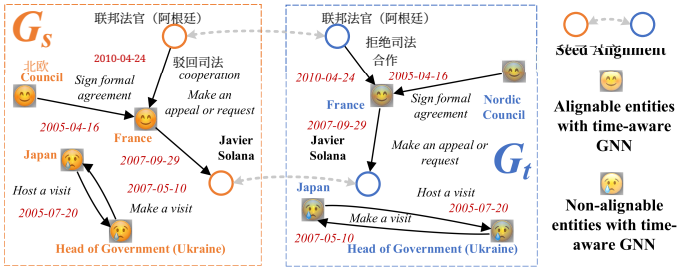

图2：以往时间感知图神经网络研究中可能发生错误对齐的一个边界案例

在关系三元组中具有相似的邻域结构，但对应不同的时间区间，如图1所示。

时间信息常用于提升各种应用的性能 [24, 25]。时序知识图谱（TKGs）[38, 47], 作为一种知识图谱，时序知识图谱通过将关系事实与时间信息关联起来而被提出。由于时序知识图谱中存储时间信息的格式几乎完全相同，因此可以轻松获取对齐信息，从而自然地提升实体对齐性能。然而，近期将时序知识图谱应用于实体对齐的研究 [53, 54] 仍存在以下两个问题。

• **“冗余”的预对齐种子**。这些研究 [53, 54] 沿袭了将知识图谱应用于实体对齐的研究，必须先获得预对齐种子才能进行实体对齐。然而，与知识图谱不同，时序知识图谱中关系三元组的时间信息天然对齐，因为它们表示现实世界中的时间点或时间段 [53, 54]。时序知识图谱的这一特点为开发无监督方式的时间感知实体对齐方法提供了机会，即将时间信息视为种子对齐。

• **时序信息发现不足**。 它们 [53, 54] 只是为每个时间戳创建嵌入，并将时间间隔视为关系类型，以增强图学习过程。这些过程高度依赖种子对齐，但其中的时间信息尚未得到充分利用，因此实体对齐的准确率受到限制。下面通过一个示例加以说明。

示例 *1.1*。 假设我们已在这些图上训练出一个准确的时间感知图神经网络，能够最大限度地传播所有对齐信息，如图2所示。在这种情况下，实体“日本”可能会被错误对齐，原因有二：(1)“日本”与种子位于不同的连通分量中，图神经网络不太可能将对齐信息传播到“日本”的嵌入中；(2)“日本”与种子没有共同的时间戳和关系类型，因此种子信息无法通过共享的时序或关系嵌入传递到“日本”。这种对日本的错误对齐使其被排除在所学模型的优化目标之外。然而，如图2所示，每个实体至少有一条带有唯一时间戳的出边。因此，根据一对一映射假设 [4], , 只需利用实体出边的时间戳对实体进行哈希，就能正确对齐“日本”。

我们提出了一种用于时序知识图谱的实体对齐方法 DualMatch，能够解决上述两个问题。为解决第一个问题，我们不再将时间区间简单地视为关系 [53, 54], , 而是使用免学习编码器对时间信息进行建模。具体而言，我们从编码器中生成伪种子，用于训练基于图神经网络的实体对齐模型。这些模型无需任何训练数据即可编码关系信息。为解决第二个问题，我们开发了一种独立于种子对齐的解码器，用于融合编码为特征向量后的时序信息和关系信息。我们将实体对齐转化为加权图匹配，以便合理平衡时序信息和关系信息。我们的贡献总结如下：

• 框架。 我们设计了 DualMatch 1，这是一种面向时序知识图谱的实体对齐框架，能够有效利用时序信息和关系信息。该框架包含两个步骤：(i) 对时序知识图谱中的信息进行编码；(ii) 将嵌入数据解码为对齐矩阵。通过采用编码器 **双编码器**和解码器 **GM-Decoder**，DualMatch 能够在有或没有训练数据的情况下对时序知识图谱执行实体对齐。

• 编码器 我们开发了一种新颖的双编码器，分别对时序信息和关系信息进行编码。双编码器包含两个组件：(i) 时间编码器，用于保留时序知识图谱中的时序信息；(ii) 关系编码器，利用图结构学习关系信息。

• 解码器 我们设计了一种新颖的 GM-Decoder，通过在实体对齐任务与加权图匹配任务之间搭建桥梁，巧妙地平衡图结构信息和时序信息。

• 实验我们在三个真实世界数据集上进行了全面实验，结果表明，DualMatch 在有监督和无监督设置下的准确率均优于现有时序知识图谱模型 [53, 54] 。

2 相关工作

实体对齐编码器。大多数现有的实体对齐编码器旨在学习知识图谱的图结构。它们可分为两类：知识图谱嵌入-基础编码器 [4, 42, 45, 60] 和基于图神经网络的编码器 [2, 23, 31, 33, 40, 43, 49]。前者使用知识图谱嵌入模型（例如 TransE [1]）来学习实体嵌入，后者则使用图神经网络 [20]。近年来，基于图神经网络的模型展现出了卓越的性能 [31]。这得益于图神经网络借助各向异性注意力机制捕捉图结构的强大建模能力 [46]。然而，它们无法对时序知识图谱中的时间信息进行建模。

时序实体对齐编码器。 TEA-GNN [53] 和 TREA [54] 将时间戳中的时间信息融入图神经网络架构。然而，这些方法尚未充分探索图神经网络编码器嵌入中时间信息的内在效用，且实体对齐需要种子对齐。相反，所提出的双编码器将时间信息与关系信息分别处理，以提升性能，并且无需种子对齐即可运行。

实体对齐解码器。大多数现有的实体对齐解码器都旨在根据学习到的嵌入寻找等价实体。最常用的解码器是贪心搜索 [44]。它本质上是根据实体嵌入之间的相似度寻找最近邻 [4, 42, 44, 60]。为提升实体对齐的性能，CSLS [21] 提出对相似度矩阵进行归一化。然而，CSLS 的贪心最近邻搜索性能有限，因为它只能利用局部最近邻对相似度矩阵进行部分归一化。CEA [57] 采用延迟接受算法（DAA）[36] 来寻找两个知识图谱中实体之间的稳定匹配，其结果质量高于 CSLS。尽管如此，由于 DAA 需要通过迭代来匹配和取消匹配实体对，因此难以在图形处理器上并行化。SEU [32] 通过采用匈牙利算法 [13] 或 Sinkhorn算法 [5]将实体对齐转化为指派问题

[1]代码可在 https://github.com/ZJU-DAILY/DualMatch/ 获取

以对相似度矩阵进行归一化。DATTI [30] 使用三阶张量同构对嵌入进行解码。近期研究 [6, 9, 26, 51,56] 还提出了多种利用采样处理大规模实体对齐的解码器。与以往的解码器不同，我们提出的 GM-Decoder 旨在解码时序知识图谱的特征。它需要两组输入特征，并能有效融合它们以生成高质量的对齐结果。

无监督实体对齐。 所有现有的无监督实体对齐方法都利用知识图谱的侧信息，包括实体名称的字面信息 [8, 32] 和视觉信息 [27]。这类方法无需种子对齐即可执行实体对齐 [44]。然而，采用机器翻译或预对齐词嵌入的模型可能会因名称偏差问题而被高估 [3, 27, 29, 39]。此外，从知识图谱中提取视觉信息也很困难。相比之下，本文提出的 DualMatch 仅需从本文的时序知识图谱中轻松获取的信息，且不会受到名称偏差问题的影响。

3 预备知识

接下来，我们将介绍一些预备定义。在此基础上，我们将形式化定义时间感知实体对齐问题。

3.1 指派问题

指派问题是一个成熟的组合优化问题。其目标是找到最优指派 P，将源分布转换为目标分布，同时使整体收益最大化。寻找最优 P 可视为最优传输问题 [35]的一个特例。本文采用 Sinkhorn算法 [5] 求解指派问题，通过迭代 k 步来缩放相似度/代价矩阵。

指派问题的一种扩展是二次指派问题，它通过用二次不等式定义收益函数来扩展指派问题。它也通常称为加权图匹配问题 [11]。给定两个邻接矩阵 $K_X \in \mathbb{R}^{N\times N}$ 和 $K_Y \in \mathbb{R}^{N\times N}$，加权图匹配问题旨在找到一个最优 $P \in \mathbb{P}_N$，使 $\|K_X P - P K_Y\|_{22}$ 最小。

3.2 图核

图核是一种广泛用于结构挖掘的核函数，用于计算图上的内积。图核可用于计算两个图的相似程度。因此，我们采用 Weisfeiler-Lehman（WL）子树核 [37]，这是最常用的图核之一，用于检测两个邻接矩阵之间的相似度。它通过运行 WL 同构测试 [50]来计算图的相似度。具体而言，给定一个图集合 $\mathbb{S}_G$，我们计算 WL核矩阵 $K^{(h)}_{WL} \in \mathbb{R}^{|\mathbb{S}_G|\times|\mathbb{S}_G|}$。对于每一对图 $(G_i, G_j) \in \mathbb{S}_G \times \mathbb{S}_G$，有

$$k^{(h)}_{WL}(G_i, G_j) = \sum_{k=0}^{h} \left\langle \phi\left(G_i^{(k)}\right), \phi\left(G_j^{(k)}\right) \right\rangle \tag{1}$$

其中，h 是 WL迭代次数，$\mathbb{n}\ G(0), \ldots, G(h)\mathbb{o}$ 是一个图 G的 WL序列，而 $\phi(G)$ 表示通过 WL同构测试获得的特征映射。对核矩阵进行归一化的传统方法是将其除以对角线上的值，其中每个元素表示图与自身的相似度，即 $\hat{k}_{WL}(G_i, G_j) = \frac{k_{WL}(G_i, G_j)}{\sqrt{diag(K_{WL}) \otimes diag(K_{WL})}}$。

3.3 问题定义

定义 1. 时序知识图谱 *(TKG)* 可表示为 $G = (E, R, T, Q)$，其中 E 是实体集合，R 是关系集合，T 是时间区间集合，$Q = \{(h, r, t, \tau) \mid h, t \in E, r \in R, \tau \in T\}$ 是四元组集合，每个四元组表示主体实体 h 在时间区间 τ 内与宾语实体 t 具有关系 r。τ 表示为 $[t_b, t_e]$，其起始时间戳为 t_b，结束时间戳为 t_e。某些事件型四元组，例如关系类型为“出生于”的四元组，可能具有 $t_b = t_e$。

定义 2。 时间感知实体对齐 *(TEA)* *[53]* 旨在找出从源时序知识图谱 $G_s = (E_s, R_s, T^*, Q_s)$到目标时序知识图谱 $G_t = (E_t, R_t, T^*, Q_t)$ 的实体 ϕ一对一映射，其中 T^* 是 G_s与 G_t之间重叠的时间区间集合。 $\phi = \{(e_s, e_t) \in E_s \times E_t \mid e_s \equiv e_t\}$，其中 $e_s \in E_s$、$e_t \in E_t$和 $\equiv$ 是两个实体之间的等价关系。

在**有监督** 设置中，一小组等价实体 $\phi' \subset \phi$ 是预先已知的，并用作种子对齐。在**无监督** 设置中，种子对齐 ϕ' 未知。

4 方法

我们提出了面向时序知识图谱的框架 DualMatch。我们首先概述 DualMatch，然后详细介绍其各个组件。

4.1 概述

图3展示了 DualMatch 框架。输入（图3左侧）包括两个时序知识图谱 G_s、G_t，以及一个可选的种子对齐槽位。为了将时序知识图谱中的时间信息和关系信息转化为更好的对齐结果，DualMatch 采用双编码器（参见图3中的步骤1）。它包含两个组件，用于对输入的时序知识图谱进行编码：(i) 时间编码器，旨在保留时序知识图谱中的时间信息；(ii) 关系编码器，旨在利用图结构学习关系信息。通过双编码器，时间信息和关系信息被编码为独立的实体特征。接下来，DualMatch 开发了 GM-Decoder（参见图3中的步骤1），通过将实体对齐任务转化为加权图匹配任务，有效融合时间特征与关系特征。转换后，GM-Decoder 能够生成近似最优的融合对齐矩阵（图3右侧）。算法1给出了 DualMatch 框架的伪代码。

4.2 时序信息编码

由于待对齐的两个时序知识图谱共享同一组记录在四元组中的时间区间，因此可以合理地作出如下假设：如果两个实体 $e_s \in E_s$ 和 $e_t \in E_t$ 所关联的四元组之间有许多重叠的时间区间，那么 e_s和 e_t很可能表示同一个现实世界对象。基于这一假设，我们提出了一种简单而有效的时间编码器。时间编码器利用重叠的时间区间为时序知识图谱中的实体构建时间特征。对于一个时序知识图谱，我们提取一个二部图 A^t，连接 E和 T。对于每个 $a^t_{et} \in A^t$，我们有

$$a^t_{et} = \frac{\exp(|Q_{et}|)}{\sum_{\tau\in T}\exp(|Q_{e\tau}|)}, \tag{2}$$

其中，$|Q_{et}|$ 是同时包含实体 e和时间区间 t的四元组数量。在这里，我们分别计算头实体和尾实体的时间邻接矩阵，并将其拼接。因此，A^t的结果维度为 $|E| \times |2T|$。我们将这个二部图称为 E的时间特征矩阵 。

利用 G_s 的 $\boldsymbol{A}^t_t$ 和 G_t的 $\boldsymbol{A}^t_s$ 时间特征矩阵，可以通过计算两两

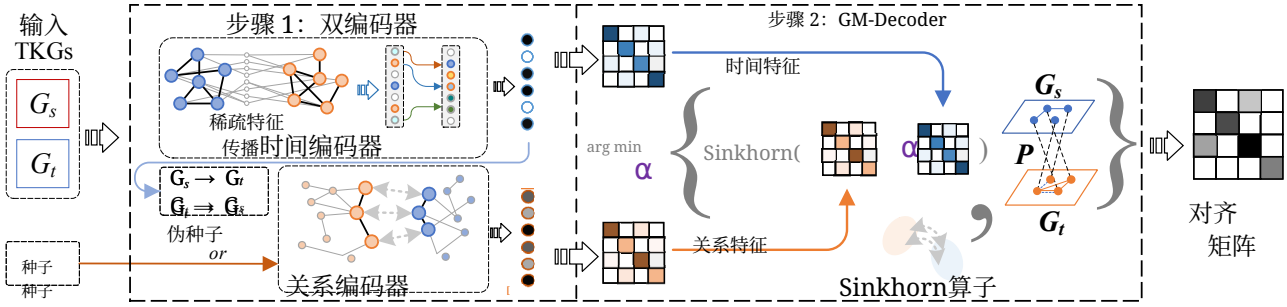

图3：DualMatch 框架

相似度： $P = A_s^t (A_t^t)^T$。然而，有些实体可能不包含足够的时间信息，从而导致对齐结果不佳。为了解决这个问题，我们从邻居聚合信息，以利用其邻域中可能存在的充足时间事实，来补全时间特征；因此，遵循 [32]执行 L层类图卷积前向传播，以获得聚合特征矩阵： $H^t = [A^t||AA^t||A^2A^t||...||A^LA^t]$（参见算法1第3–4行），其中， L 是一个超参数，表示类图卷积前向传播的层数；$[X||Y]$表示 X 和 Y 沿最后一个维度进行张量拼接操作； $A^l = \hat{1}_{i\in 1,...,l} A$ 是 G的 l跳邻接矩阵。这里，为了利用不同关系类型对知识图谱的影响，我们遵循 [30–34] 来构建关系特征矩阵。具体而言，对于每个 $a_{ij} \in A$，

$$a_{ij} = \frac{\sum_{r_j \in R_{i,j}} \ln\left(|Q|/|Q_{r_j}|\right)}{\sum_{k\in\mathcal{N}_i}\sum_{r_k\in R_{i,k}} \ln\left(|Q|/|Q_{r_k}|\right)}, \tag{3}$$

其中，N_i 表示实体 e_i 的邻居集合，$R_{i,\ j}$ 是 e_i 和 e_j 之间的关系集合，而 $|Q|$ 和 $|Q_r|$分别表示所有四元组的总数以及包含关系 r 的四元组数量。

4.3 编码关系信息

最先进的关系信息编码方法通常使用图神经网络 [20] 传播种子对齐 ϕ'所提供的信息。研究 [53, 54] 表明，整合时间信息能够提升实体对齐的性能。这些方法使用图神经网络，将实体的关系特征与时间特征结合起来进行编码。具体而言，它们通过传播邻域信息 [20, 34, 46]来训练实体嵌入。实体 $v \in E$ 在图神经网络 $h(l)_v$的第 l^{th} 层的嵌入通过聚合局部信息计算得出：

$$\begin{aligned} a_v^{(l)} &= \mathrm{AGGREGATE}^{(l)}\left(\left\{\left(h_u^{(l-1)}, \mathbf{r}_{uv}, \mathbf{t}_{uv}\right) \mid u \in \mathcal{N}(v)\right\}\right) \\ h_v^{(l)} &= \mathrm{UPDATE}^{(l)}\left(a_v^{(l)}, h_v^{(l-1)}\right), \end{aligned} \tag{4}$$

其中，$h_v^0 \in \mathbb{R}^D$ 是使用 Glo- rot 初始化的可学习嵌入向量，$\mathbf{r}_{uv} \in R$、$\mathbf{t}_{uv} \in T$ 分别表示实体 u 与 v之间的关系类型和时间区间，$\mathrm{N}(v)$ 是 v周围邻居实体的集合。模型的最终输出记为 $H^r \in \mathbb{R}^{|E_s \cup E_t| \times D}$，其中 D是嵌入的维度。

关系编码器采用最先进的实体对齐方法 [31]，构建一个 L 层图神经网络模型，以学习知识图谱结构的表示（参见算法1第9行）。具体而言， [31] 中的双方面嵌入将实体嵌入与关系嵌入拼接起来。我们在此基础上加入额外的时间嵌入，形成整体的三方面嵌入。对于每个$\boldsymbol{h}_{e_i} \in H^r$:

$$\boldsymbol{h}_{e_i} = \left[\boldsymbol{h}_{e_i}^{out} \| \left(\frac{1}{|\mathcal{N}_{e_i}^r|}\sum_{r_j\in\mathcal{N}_{e_i}^r} \boldsymbol{h}_{r_j} + \frac{1}{|\mathcal{N}_{e_i}^t|}\sum_{t_j\in\mathcal{N}_{e_i}^t} \boldsymbol{h}_{t_j}\right)\right], \tag{5}$$

其中， $\boldsymbol{h}_{e_i}^{out} = [h(0)_{e_i} \ ||h_{(1}_{e_i}||...||h^{(L)}_{e_i}]$、$\boldsymbol{h}_{r_j}$和 $\boldsymbol{h}_{t_j}$分别表示对应实体、关系和时间区间的向量，$\mathrm{N}_{e_i}^r$和$\mathrm{N}_{e_i}^t$ 表示实体 e_i周围的关系集合和时间区间集合。请注意，近期研究通常会拼接图神经网络层 [31, 33, 34], 中的所有输出特征，以取得更好的性能。以往研究采用负采样 [4, 29, 33] 构建损失函数并训练编码器模型。我们采用这一策略以实现快速收敛 [31, 54]。训练损失定义为 $\mathrm{L} = \mathrm{LSE}(\lambda z(e_s, e_t)) + \mathrm{LSE}(\lambda z(e_t, e_s))$，其中 $e_s \in E_s$、$e_t \in E_t$、$(e_s, e_t) \in \phi'$，$\mathrm{LSE}(X) = log(\sum_{x\in X} e^x)$ 是一种能够平滑生成困难负样本的运算符，λ 是 LSE 的平滑因子，而$z \in \mathrm{R}^{|E_t|}$是归一化三元组损失。具体而言，z定义为 $z(e_s, e_t) = \text{z-score}(\{\gamma + \mathrm{sim}(e_s, e_t) - \mathrm{sim}(e_s, e'_t)|e'_t \in E_t\})$，其中 $\text{z-score}(X) = \frac{X-\mu}{\sigma}(X)$ 表示标准分数归一化，而 $\mathrm{sim}(e_s, e_t) = \boldsymbol{h}_{e_s}^r \cdot \boldsymbol{h}_{e_t}^r$表示通过点积得到的两个实体之间的相似度。

4.4 信息融合与解码

朴素策略研究提出了多种技术，通过将嵌入解码为对齐矩阵来提升实体对齐性能。然而，在 GM-Decoder 中，时序信息和关系信息分别编码为 H^t和 H^r。一种朴素策略是通过 $\hat{H} = [H^r||H^t]$直接拼接这两种特征。随后，我们可以根据拼接后的特征得到对齐矩阵： $P = \hat{H}_s(\hat{H}_t)^T$.

我们的方法。 朴素的解码策略存在两个问题。首先，H^r 是一个稠密矩阵，而 H^t 在实际应用中是稀疏矩阵。由于稀疏矩阵通常采用的存储格式，将一个稀疏矩阵与一个稠密矩阵拼接会消耗大量内存。其次， H^r来自关系编码器，而 H^t来自时序编码器。因此，无须将二者视为同等重要。为解决上述两个问题，我们引入权重 α： $P = \alpha \cdot H_s^t(H_t^t)^T + H_s^r(H_t^r)^T$，其中 α是用于平衡这两种特征对对齐结果影响的权重。随后，为了对对齐矩阵进行归一化，我们采用近期的一项工作 [30, 32] ： 该工作将实体对齐重新表述为指派问题，并通过 Sinkhorn算法 [6, 8, 30, 32]求解： $\hat{P} = \mathrm{Sinkhorn}(P)$（参见算法1第13行）。请注意，在许多现实场景中， $|E_s| \neq |E_t|$。在这种情况下，实体对齐问题会转化为非平衡指派问题，进而可以直接重新表述为平衡指派问题 [32]。

算法1： 所提出的 DualMatch 流程

输入： 时序知识图谱 G_s、G_t、可选的种子对齐 ϕ'，以及范围 R_α
输出： 对齐矩阵 $\hat{P}$

/* 步骤1：编码时序和关系信息
通过双编码器转换为向量 */
$A_s,\ A_s^t,\ \leftarrow G_s,\ A_t,\ A_t^t \leftarrow G_t$ /* 提取特征矩阵 */
$H_s^t \leftarrow$ 时间编码器$(A_s,\ A_s^t)$
$H_t^t \leftarrow$ 时间编码器$(A_t,\ A_t^t)$
如果 ϕ' 是 *not given* **则** /* 计算伪种子 */
  $P^{G_sG_t} \leftarrow \text{Sinkhorn}(H_s^t(H_t^t)^T)$
  $P^{G_tG_s} \leftarrow \text{Sinkhorn}(H_t^t(H_s^t)^T)$
  $\phi' \leftarrow \{(i,\ j) \mid$ 取最大值$_{j'}\ P_{ij}^{G_sG_t}$,$= j \wedge$ 取最大值$_{i'}\ P_{ji}^{G_tG_s}$,$= i\}$
$H_s^r,\ H_t^r \leftarrow$ 关系编码器$(G_s,\ G_t,\ \phi')$
/* 步骤 2：使用 GM-Decoder 解码特征向量 */
$\hat{P}, D \leftarrow zeros(|E_s|, |E_t|) + \infty$
对于 $\alpha \in R_\alpha$ **do**
  $\hat{P}_\alpha \leftarrow \text{Sinkhorn}(\alpha \cdot H_s^t(H_t^t)^T + H_s^r(H_t^r)^T)$
  $D_{G_sG_t}^\alpha = k^r \left\|A_s\hat{P}_\alpha - \hat{P}_\alpha A_t\right\|_2^2 + k^t \left\|A_s^t - \hat{P}_\alpha A_t^t\right\|_2^2$
  如果 $D_{G_sG_t}^\alpha < D$**则**
    $D, \hat{P} \leftarrow D_{G_sG_t}^\alpha,\ \hat{P}_\alpha$ /* 更新对齐矩阵 */
返回 $\hat{P}$.

然而，由于真实值未知，难以确定最优的 α。为解决这一问题，我们研究待对齐的两个时序知识图谱的同构性质，并提出通过最小化给定 $\hat{P}$ 时加权图匹配 [11] 的目标函数来设置权重。具体而言，我们将源邻接矩阵映射到目标邻接矩阵。接下来，通过计算映射后的矩阵与目标矩阵之间的距离，我们得出对齐矩阵 $\hat{P}$ 的正确性（参见算法1第14行）。

$$D_{G_sG_t} = k^r \left\|A_s\hat{P} - \hat{P}A_t\right\|_2^2 + k^t \left\|A_s^t - \hat{P}A_t^t\right\|_2^2, \tag{6}$$

其中 $k^r = \hat{k}_{WL}(A_s, A_t)$ 和 $k^t = \hat{k}_{WL}(A_s^t, A_t^t)$ 是基于邻接矩阵同构性的权重。

令 $d^r = \left\|A_s\hat{P} - \hat{P}A_t\right\|_2^2$ 和 $d^t = \left\|A_s^t - \hat{P}A_t^t\right\|_2^2$ 分别表示根据关系特征矩阵和时间特征矩阵测得的两个时序知识图谱的距离。由于两个时序知识图谱可能非同构，且关系邻接矩阵和时间邻接矩阵是独立构建的，d^r和 d^t的尺度可能不同。因此，为它们赋予不同的权重是合理的。我们用 $sim(\cdot)$ 表示两个邻接矩阵之间的相似度。直观而言，如果 $sim(A_s, A_t) > sim(A_s^t, A_t^t)$，则可以赋予 d^r更大的权重。这促使我们使用 WL 图核 [37]测得的图相似度来设定权重。

我们在 R_α 的范围内搜索 $\alpha = \arg\min_\alpha D_{G_sG_t}$，以获得最优的 $\hat{P}$（参见算法1第11–16行）。使用映射矩阵匹配两个图时，我们无需将源图中的多个实体映射到目标图中。此处，$zeros(S)$ 表示一个形状为 S、填充值为 0 的张量。然而，$\hat{P}$ 的每一行中都有多个非零值。这意味着，对于源时序知识图谱中的一个实体，在目标时序知识图谱中存在多个具有加权分数的可能对齐。为此，我们决定仅保留排名第一的对应关系，以稀疏化 $\hat{P}$。为使矩阵保持双随机，我们将排名第一的对应关系的值设为 1：

$$\hat{P}_{ij} = \begin{cases} 1, & j = \arg\max_{j'}(\hat{P}_{ij'}) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

4.5 无监督学习

由于关系编码器需要种子对齐，为了执行无监督实体对齐，我们遵循双向策略 [33]，从时间编码器生成伪种子。考虑到对齐方向 [6, 9]，的非对称性，我们首先通过在两个方向上应用时间编码器，从时间特征中推导出双向对齐矩阵 $P^{G_sG_t} = \text{Sinkhorn}(H_s^t(H_t^t)^T)$ 和 $P^{G_tG_s} = \text{Sinkhorn}(H_t^t(H_s^t)^T)$（参见算法1第6–7行）。此过程无需监督即可完成。然后，我们从这两个矩阵中提取伪种子。具体而言，当且仅当 $\arg\max_{j'} P_{ij'}^{G_sG_t} = j$ 且 $\arg\max_{i'} P_{ji'}^{G_tG_s} = i$ 时，我们将 (e_i, e_j) 中满足 $e_i \in E_s,\ e_j \in E_t$ 的项视为有效种子对（参见算法1第8行）。最后，利用生成的伪种子对齐，我们训练关系编码器（参见算法1第9行），并通过 GM-Decoder 获得无监督对齐结果 $\hat{P}$（参见算法1第11–17行）。

讨论。 许多以往研究尝试从机器翻译系统或预训练嵌入中生成伪种子 [27, 31, 33, 34, 41]，以提升实体对齐性能。与这些研究不同，我们利用时序知识图谱固有的信息生成伪种子。这意味着，DualMatch 可以充分利用数据集的固有信息来执行实体对齐，而不必依赖难以获取的信息。

5 实验

我们报告了旨在评估 DualMatch 性能的大量实验。

5.1 实验设置

数据集。 我们使用了 [53, 54] 05 15 [7], [38], 提供的三个真实世界时序知识图谱数据集。它们分别从 ICEWS、YAGO 和 Wikidata [47], 中提取，记为 DICEWS、WY50K 和 WY20K。所有数据集的详细统计信息见附录 B。

基线方法。 我们使用 12 种最先进的实体对齐方法作为基线方法。我们遵循 [6, 27, 29, 31, 34, 53, 54], 中的设置，不使用时序知识图谱中的时序和关系信息之外的侧信息，以确保比较公平。下面将基线方法分为两类。

- **时间无关基线方法**不利用时序信息执行实体对齐，包括：（1）*MTransE* [4],，首个基于嵌入的实体对齐模型；（2）*JAPE* [41],，利用属性相关性进行实体对齐；（3）*AlignE* [42],，以面向对齐的方式训练知识图谱嵌入；（4）*GCN-Align* [49],，通过图卷积网络对齐实体；（5）*MuGNN*[2],，通过多通道图神经网络学习面向对齐的嵌入；（6）*MRAEA* [33],，利用实体的邻居及其相连关系的元语义，学习跨图实体嵌入；（7）*HyperKA* [40],，为实体对齐学习双曲嵌入；（8）*RREA* [34],，利用关系反射变换，为实体对齐获取关系特定嵌入；（9）*KE-GCN* [55],，学习一种基于知识嵌入的图卷积网络；以及（10）*Dual-AMN* [31],，当前最先进的实体对齐模型，仅利用知识图谱结构中的关系信息。

- **时间感知基线方法**同时考虑关系信息和时间信息，包括：（1）*TEA-GNN* [53],，将时间信息建模为嵌入；以及（2）*TREA* [54],，这是采用时间关系注意力机制的最先进时间感知实体对齐模型。

表1：DICEWS 和 WY50K 上的实体对齐总体结果

方法		DICEWS-1K			DICEWS-200			WY50K-5K			WY50K-1K		
		H@1	H@10	平均倒数排名	H@1	H@10	平均倒数排名	H@1	H@10	平均倒数排名	H@1	H@10	平均倒数排名
时间无关	MTransE	10.1	24.1	0.150	6.7	17.5	0.104	24.2	47.7	0.322	1.2	6.7	0.033
	JAPE	14.4	29.8	0.198	9.8	21.0	0.138	27.1	48.8	0.345	10.1	26.2	0.157
	AlignE	50.8	75.1	0.593	22.2	45.7	0.303	75.6	88.3	0.800	56.5	71.4	0.618
	GCN-Align	20.4	46.6	0.291	16.5	36.3	0.231	51.2	71.1	0.581	21.7	39.8	0.279
	MuGNN	52.5	79.4	0.617	36.7	58.3	0.412	76.2	89.0	0.808	58.9	73.3	0.632
	MRAEA	67.5	87.0	0.745	47.6	73.3	0.564	80.6	91.3	0.848	62.3	80.1	0.685
	HyperKA	58.8	84.2	0.669	38.3	65.3	0.474	78.4	90.0	0.829	61.0	77.5	0.665
	RREA	72.2	88.3	0.780	65.9	82.4	0.719	82.8	93.8	0.868	69.6	85.9	0.753
	KE-GCN	54.9	82.7	0.650	37.3	62.5	0.451	78.0	91.0	0.831	60.0	76.1	0.654
	Dual-AMN	71.6	89.3	0.779	66.8	85.4	0.733	89.7	96.4	0.922	75.5	89.0	0.834
时间感知	TEA-GNN	88.7	94.7	0.911	87.6	94.1	0.902	87.9	96.1	0.909	72.3	87.1	0.775
	TREA	91.4	96.6	0.933	91.0	96.0	0.927	94.0	98.9	0.958	84.0	93.7	0.885
	DualMatch	**95.3**	**97.3**	**0.961**	**95.3**	**97.4**	**0.961**	**98.1**	**99.6**	**0.986**	**94.7**	**98.4**	**0.961**
无监督	DualMatch（无监督）	**94.6**	**97.1**	**0.956**	-	-	-	**96.4**	**99.1**	**0.975**	-	-	-

评估指标。 广泛采用的*Hits@N(H@N)*(N = 1，10)和平均倒数排名（*MRR*）被用作评估指标 [4, 30, 31, 42] N 。Hits@（以百分比表示）表示对齐矩阵$\hat{}$***P***中排名前N位的实体里正确对齐实体所占的比例。MRR 是正确对齐实体倒数排名的平均值，其中，倒数排名表示从$\hat{}$***P***中得到的正确对齐的平均排名。请注意，较高的 H@N和 MRR 表明更高的实体对齐准确率。

实验设置。 为了避免劳动密集型的种子对齐，我们希望尽可能少地使用训练数据。传统的实体对齐方法通常使用总配对数的 20%–30% 来训练实体对齐模型 [44]。时间信息可以作为时序知识图谱种子对齐的替代方案，因此需要使用更少的训练配对。我们采用三种设置 的训练配对比例来评估实体对齐性能，包括两种遵循以往研究 [53, 54], 的有监督 设置，以及一种不使用种子的无监督设置。

• **常规设置**对 DICEWS 和 WY50K 分别使用约 10% 的种子，即 1000 对和 5000 对，分别记为 DICEWS-1K 和 WY50K-5K。其余配对用于验证实体对齐性能。

• **少种子设置**对 DICEWS 和 WY50K 分别使用 200 个和 1000 个种子，分别记为 DICEWS-200 和 WY50K-1K。其余配对用于评估实体对齐性能。

• **无监督设置**[2] 不包含种子对齐，所有实体对均用于评估。由于以往的方法需要带标签的种子对齐，我们仅提供 DualMatch 的无监督实体对齐结果，记为 DualMatch（无监督）。

我们遵循各基线方法原有的设置进行实现。具体而言，表2–5中的**粗体**数字表示有监督设置和无监督设置下的最高性能。对于 DualMatch 中的双编码器，我们遵循以往研究 [31, 32]，设置聚合时间特征和训练基于图神经网络的实体对齐模型所需的超参数；对于 GM-Decoder，我们在 R_α = {0，0.25 0.5 0.75 1 1.25 1.5 1.75 2,,,,,,}中搜索使$D_{G_s G_t}$最小的最佳 α。我们按照 [32]设置 WL 迭代次数h = 8和 Sinkhorn 的步长 k = 10。所有算法均使用 Python 实现，实验在一台配备 Intel Core i9-10900K CPU、NVIDIA GeForce RTX3090 GPU 和 128GB 内存的计算机上运行。相关材料可在*https://do*获取

[2]无监督设置使用所有对齐对进行评估。为确保公平比较，我们将其与基线模型的常规设置进行比较。

5.2 比较

有监督性能。表1总结了 DICEWS 和 WY50K 上有监督设置下的实体对齐性能。首先，DualMatch 在两个数据集的所有指标上均取得了最佳性能。与两种最先进的时间感知基线方法 TEA-GNN [53] 和 TREA [54]相比，Dual-Match 的 H@1 提高了 3.9% – 10.7%。这验证了我们整合时间信息的方式具有优越性。具体而言，TEA-GNN [53] 和 TREA [54] 仅将时间信息建模为可学习权重矩阵，而 DualMatch 能够捕捉时序知识图谱中的时间信息，使实体对齐能够使用独立的时间编码器。其次，所有时间感知方法在 H@1 指标上均优于时间无关方法。这证实了在实体对齐中纳入时间信息的优越性。第三，与时间无关方法相比，DualMatch 的 H@1 提高了 8.4% – 28.5%，这证明了 Dual-Match 的准确性。最后，在所有时间无关基线方法中，Dual-AMN [31] 取得了最佳对齐性能。这是因为其图神经网络模型的设计最能捕捉隐藏在时序知识图谱图结构中的关系信息。这表明，由其扩展而来的关系编码器能够准确地学习关系信息。

无监督性能。 通过采用双向策略，时间编码器分别为 DICE WS、WY50K 和 WY20K 生成了 8168、35403 和 9778 个伪种子对。随后，我们基于这些伪种子训练关系编码器，并在表 1 中报告无监督设置下最终对齐矩阵的性能。可以观察到，DualMatch 的 H@1 比 TREA（表现最佳的基线方法）高.4% – 3.2%。此外，与 TREA 相比，DualMatch 的 H@1 提升了.6% – 12.4%；在种子设置较少时，TREA 在所有基线方法中表现最佳。总体而言，与最先进方法相比，DualMatch 在有监督和无监督设置下都能实现更好的实体对齐。

5.3 消融研究

我们逐一移除 DualMatch 的各个组件，并在表2中报告 H@1、H@10 和平均倒数排名。首先，移除时间编码器组件（DualMatch - TE）后，准确率有所下降。这表明 时间编码器确实捕获了更多时间信息 .

i .org/10.5281/zenodo.7149372.

其次，移除关系编码器组件（DualMatch - RE）后，DualMatch 的 H@1 下降。这证实了关系编码器在编码关系信息方面的有效性。请注意

表2：消融研究结果

方法	DICEWS-200			WY50K-1K		
	H@1	H@10	平均倒数排名	H@1	H@10	平均倒数排名
DualMatch	**95.3**	**97.4**	0.961	**94.7**	**98.4**	**0.961**
DualMatch - TE	90.3	95.2	0.922	88.4	96.2	0.913
DualMatch - RE	92.8	97.0	0.944	67.8	79.7	0.720
DualMatch - TAL	91.9	95.8	0.934	91.9	96.6	0.936
DualMatch - TA	94.1	96.5	0.951	94.2	98.1	0.957
DualMatch - α	**95.3**	**97.4**	**0.962**	93.9	98.2	0.955
DualMatch - GM-D	91.3	94.8	0.927	80.4	92.2	0.845

表3：敏感性分析结果

方法	高度时间敏感			低度时间敏感		
	H@1	H@10	平均倒数排名	H@1	H@10	平均倒数排名
DualMatch	99.1	**100.0**	0.994	35.9	49.7	0.407
DualMatch（无监督）	**99.8**	**100.0**	**0.999**	**43.1**	**52.4**	**0.467**
TREA	87.5	96.1	0.909	31.9	47.0	0.370
TEA-GNN	85.3	95.0	0.888	31.9	44.9	0.364

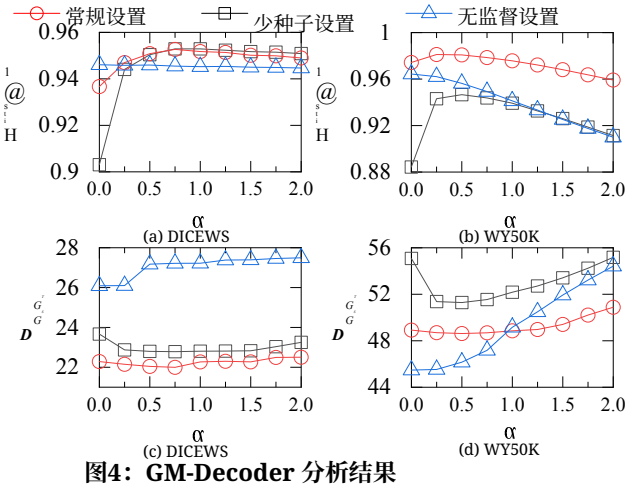

图4：GM-Decoder 分析结果

在 DICEWS 上，移除时间编码器对准确率的影响与移除关系编码器相似；而在 WY50K 上，移除关系编码器会导致准确率大幅下降。这是因为时间信息和关系信息的重要性因数据集而异。具体而言，关系信息对于在 WY50K 上取得更好的性能更为重要。第三，只需设置 $\alpha = 1$（DualMatch - α），所有方法在 DICEWS 上的准确率几乎保持不变，但在 WY50K 上则有所下降。这也表明，在 DICEWS 中，时间信息与关系信息相对同等重要；而在 WY50K 中，关系信息远比时间信息重要。有关此问题的更多细节将在第5.5节中介绍。第四，通过移除图卷积

与时间编码器（DualMatch - TA）中的类似聚合操作一样，准确率有所下降。这表明，聚合过程能够传播时间信息，从而提升整体性能。接下来，将三方面学习替换为原始的双方面学习（DualMatch - TAL）后，准确率有所下降。这说明，将时间信息纳入学习过程是提升实体对齐效果的关键。最后，将 GM-Decoder 替换为第5.5节所述的 朴素策略 （DualMatch - GM-D）后，DualMatch 的准确率显著下降。这验证了 GM-Decoder 的有效性。

5.4 敏感性研究

在现实世界应用中，某些实体不受时间信息影响。因此，我们使用包含一些非时序事实的时间混合数据集 WY20K，研究对齐结果对时间信息的敏感性。根据 [53

54], 我们根据实体所连接的时序事实数量占其连接的全部事实数量的比例，将实体对划分为高度时间敏感实体对和低度时间敏感实体对。使用 400 对作为监督模型训练的种子，6,898 对为高度时间敏感实体对，其余为低度时间敏感实体对。

我们将 DualMatch 的实体对齐性能与具备时间感知能力的基线方法进行比较，结果见表3。我们还报告了在数据集上执行无监督实体对齐时 DualMatch 的结果，此时不使用 400 个种子。首先，我们发现，在对齐高度敏感和低度敏感的实体对方面，DualMatch 均显著优于以往方法。这表明，在时间信息不足的真实应用场景中，DualMatch 整体表现出色。此外，对于高度时间敏感的实体对，DualMatch 的 H@1 几乎达到 100%。这证实了 DualMatch 通过独立的时间编码器，能够最大限度地捕获时间信息。其次，对于低度敏感的实体对，在两种设置下，DualMatch 的 H@1 均高于基线方法，这验证了即使对于时间信息较少的实体，我们的整体设计也更有效。第三，在低度敏感的实体对上，DualMatch 的无监督变体大幅优于其有监督版本。这是因为我们利用基于时间的伪种子，将时间信息转化为关系信息。最后，尽管 TREA [54] 在 DICEWS 和 WY50K 数据集上的表现都优于 TEA-GNN [53]，但 TREA 的性能与 TEA-GNN 几乎相同，尤其是在低度敏感的实体对上。这是因为在时间混合数据集 [53, 54]中，时间嵌入设计不佳，限制了时间信息的作用。相比之下，DualMatch 能更好地利用时间信息和关系信息，因此优于所有基线方法。

5.5 GM-Decoder 评估

GM-Decoder制定了一个目标 $D_{G_sG_t}$，通过将实体对齐问题转化为图匹配问题来优化实体对齐性能。理想情况下，准确率应与 $D_{G_sG_t}$ 成反比，也就是说，$D_{G_sG_t}$ 越低，实体对齐性能越好。这与最小化距离的目标一致。为了检验实际情况是否如此，我们通过绘制 $D_{G_sG_t}$ 与对齐性能之间的关系，评估 GM-Decoder 寻找最优 α的有效性。我们在三种设置下将 α 从 0 调整到 2（参见第5.1节），并记录每一步的 H@1 和 $D_{G_sG_t}$，如图4所示。首先，我们观察到，在三种设置下，H@1 均与 $D_{G_sG_t}$ 负相关，这验证了 GM-Decoder 在寻找最优 α方面的有效性。表4也证明了这一点，其中我们报告了调整 α 时 H@1 的最差和最佳结果。表4显示，GM-Decoder 在所有情况下都能找到最佳 α。其次，在无监督设置下，随着 α 下降，性能会变得更好。这是因为关系编码器中使用的伪种子由时间编码器生成，这表明关系编码器已经学到了时序信息。第三，常规设置的表现通常优于少种子设置。这证实了有更多种子对齐时，关系编码器能够

表4：变化 α时 H@1 的最差值和最佳值

设置	DICEWS			WY50K		
	最差	最佳	DualMatch	最差	最佳	DualMatch
正常	93.7	**95.3**	**95.3**	95.9	**98.1**	**98.1**
少量种子	90.3	**95.3**	**95.3**	88.4	**94.7**	**94.7**
无监督	94.5	**94.6**	**94.6**	91.0	**96.4**	**96.4**

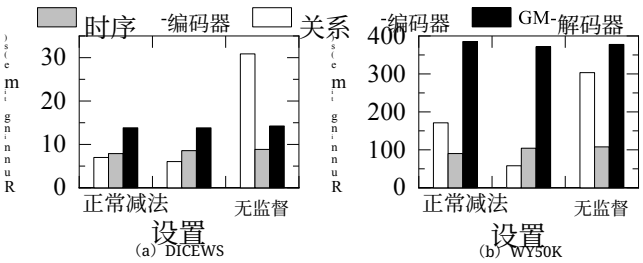

图 5：不同设置下的可扩展性分析

捕获更多关系信息。最后，我们观察到，DICEWS 的最优 α约为 1，而 WY50K 的最优值约为 0.5。这意味着，DICEWS 中时序信息和关系信息的重要性几乎相同，而 WY50K 中两者的重要性则存在偏差。这可能是由于生成 DICEWS [53], 的算法将 ICEWS05-15 数据集复制为两个分布几乎相同的时序知识图谱。根据比较结果，WY50K 更适用于现实世界应用，因为时序信息和关系信息的数量通常存在偏差。

5.6 可扩展性研究

为了验证 DualMatch 的可扩展性，我们在图 5 中绘制了 DICEWS 和 WY50K 上各组件的运行时间，其中 Normal、Less 和 Unsup 分别表示常规设置、少种子设置和无监督设置。具体而言，各组件包括：(1)关系编码器，用于训练图神经网络模型以编码关系信息；(2) 时间编码器，用于接收 TKG 输入并编码时间信息；以及 (3) *GM-Decoder*，用于融合两种编码信息以生成对齐矩阵。我们观察到，在 DICEWS 和 WY50K 上，每个组件的运行时间均不超过 103 秒，这证实了 DualMatch 在大型数据集上的可扩展性。我们还观察到，无监督设置下的训练时间显著高于其他设置。这是因为无监督设置需要更多的训练迭代次数，其中伪种子的数量是常规设置下的 7.0 – 8.2 倍。为了进一步研究 DualMatch 的可扩展性，我们还在附录 E 中基于采样子图提供了可扩展性研究。

6 结论

在本研究中，我们将时序知识图谱的实体对齐问题重新诠释为加权图匹配问题，并探究利用带时间戳的孤立关系三元组，在时序知识图谱之间构建无监督实体对齐方法的潜力。我们提出了 DualMatch，它通过融合关系和时间数据，以无监督方式对齐实体。所提出的 DualMatch 包含两个步骤：(i) 将关系信息和时间信息分别编码为嵌入；(ii) 整合这两类信息，并通过一种新颖的基于图匹配的解码器将其转换为对齐结果。使用我们的方法，可以在有监督或无监督的情况下进行时序知识图谱上的实体对齐。此外，该方法不需要实体名称或早期无监督实体对齐技术所需的预训练词嵌入等额外辅助信息。在多个真实世界时序知识图谱数据集上的大量实验结果表明，即使没有训练数据，我们的方法也显著优于当前最先进的有监督方法。未来，探索时序数据集上的悬空实体对齐设置 [39] 将是一个值得研究的方向。

7 致谢

本研究部分由国家重点研发计划（项目编号：2021YFC3300303）、国家自然科学基金委员会（项目编号：62025206、61972338 和 62102351）以及宁波市科技创新专项（项目编号：2022Z095 和 2021Z019）资助。Lu Chen 是本文的通讯作者。

参考文献

[1] Antoine Bordes、Nicolas Usunier、Alberto García-Durán、Jason Weston 和 Oksana Yakhnenko。2013。用于建模多关系数据的嵌入翻译。载于 NeurIPS。2787–2795。[2] Yixin Cao、Zhiyuan Liu、Chengjiang Li、Zhiyuan Liu、Juanzi Li 和 Tat-Seng Chua。2019。用于实体对齐的多通道图神经网络。载于 ACL。1452–1461。[3] Muhao Chen、Weijia Shi、Ben Zhou 和 Dan Roth。2021。借助偶然监督进行跨语言实体对齐。（2021），645–658。[4] Muhao Chen、Yingtao Tian、Mohan Yang 和 Carlo Zaniolo。2017。用于跨语言知识对齐的多语言知识图谱嵌入。载于 IJCAI。1511–1517。[5] Marco Cuturi。2013。Sinkhorn距离：以闪电般的速度计算最优传输。载于 NeurIPS。2292–2300。[6] Yunjun Gao、Xiaoze Liu、Junyang Wu、Tianyi Li、Pengfei Wang 和 Lu Chen。2022。ClusterEA：采用随机训练和归一化小批量相似度的可扩展实体对齐。载于 KDD。421–431。[7] Alberto García-Durán、Sebastijan Dumancic 和 Mathias Niepert。2018。学习用于时序知识图谱补全的序列编码器。载于 EMNLP。4816–4821。[8] Congcong Ge、Xiaoze Liu、Lu Chen、Baihua Zheng 和 Yunjun Gao。2021。化繁为简：一种有效的端到端实体对齐框架。载于 SIGIR。777–786。[9] Congcong Ge、Xiaoze Liu、Lu Chen、Baihua Zheng 和 Yunjun Gao。2022。LargeEA：面向大规模知识图谱的实体对齐。PVLDB 15，2（2022），237–245。[10] Congcong Ge、Pengfei Wang、Lu Chen、Xiaoze Liu、Baihua Zheng 和 Yunjun Gao。2021。CollaborEM：一种利用多特征协作的自监督实体匹配框架。TKDE（2021）。[11] Steven Gold 和 Anand Rangarajan。1996。一种用于图匹配的渐进式指派算法。TPAMI 18，4（1996），377–388。[12] William L Hamilton、Rex Ying 和 Jure Leskovec。2017。大图上的归纳式表示学习。载于 NeurIPS。1025–1035。[13] Harold W. Kuhn。1955。用于指派问题的匈牙利算法。《海军研究后勤季刊》2，1，83–97。[14] Xuming Hu、Fukun Ma、Chenyao Liu、Chenwei Zhang、Lijie Wen 和 Philip S Yu。2021。通过增量元自训练进行半监督关系抽取。载于 EMNLP Findings。487–496。[15] Xuming Hu、Lijie Wen、Yusong Xu、Chenwei Zhang 和 Philip S. Yu。2020。Self-ORE：用于开放关系抽取的自监督关系特征学习。载于 EMNLP。3673–3682。[16] Xuming Hu、Chenwei Zhang、Yawen Yang、Xiaohe Li、Li Lin、Lijie Wen 和 Philip S. Yu。2021。面向低资源关系抽取的梯度模仿强化学习。载于 EMNLP。2737–2746。[17] Prachi Jain、Sushant Rathi、Mausam 和 Soumen Chakrabarti。2020。时序知识库补全：新算法与评估协议。载于 EMNLP。3733–3747。[18] Chengyue Jiang、Wenyang Hui、Yong Jiang、Xiaobin Wang、Pengjun Xie 和 Kewei Tu。2022。召回、扩展与多候选交叉编码：快速且准确的超细粒度实体类型判定。arXiv预印本 arXiv:abs/2212.09125（2022）。[19] Chengyue Jiang、Yong Jiang、Weiqi Wu、Pengjun Xie 和 Kewei Tu。2022。利用神经成对条件随机场对超细粒度实体类型判定中的标签相关性建模。arXiv预印本 arXiv:2212.01581（2022）。[20] Thomas N. Kipf 和 Max Welling。2017。利用图卷积网络进行半监督分类。载于 ICLR。[21] Guillaume Lample、Alexis Conneau、Marc'Aurelio Ranzato、Ludovic Denoyer 和 Hervé Jégou。2018。无平行数据的词语翻译。载于 ICLR。[22] Jennifer Lautenschlager、Steve Shellman 和 Michael Ward。2015。ICEWS事件聚合。https://doi.org/ 10 7910 28117 。/DVN/ Chengjiang Li、Yixin [23] Cao、Lei Hou、Jiaxin Shi、Juanzi Li 和 Tat-Seng Chua。2019。通过联合知识嵌入模型和

跨图模型。载于 *EMNLP*。2723–2732。[24] Tianyi Li、Lu Chen、Christian S Jensen 和 Torben Bach Pedersen。2021。TRACE：道路网络中流式轨迹的实时压缩。*PVLDB* 14，7（2021），1175–1187。[25] Tianyi Li、Ruikai Huang、Lu Chen、Christian S Jensen 和 Torben Bach Pedersen。2020。道路网络中不确定轨迹的压缩。*PVLDB* 13，7（2020），1050–1063。[26] Bing Liu、Wen Hua、Guido Zuccon、Genghong Zhao 和 Xia Zhang。2022。面向大规模实体对齐的高质量任务划分。*arXiv*预印本 *arXiv:2208.10366*（2022）。[27] Fangyu Liu、Muhao Chen、Dan Roth 和 Nigel Collier。2021。用于（无监督）实体对齐的视觉枢轴。载于 *AAAI*。4257–4266。[28] Shuliang Liu、Xuming Hu、Chenwei Zhang、Shu'ang Li、Lijie Wen 和 Philip S. Yu。2022。HiURE：用于无监督关系抽取的层次化样例对比学习。载于 *NAACL*。5970–5980。[29] Zhiyuan Liu、Yixin Cao、Liangming Pan、Juanzi Li 和 Tat-Seng Chua。2020。探索并评估实体对齐中的属性、值与结构。载于 *EMNLP*。6355–6364。[30] Xin Mao、Meirong Ma、Hao Yuan、Jianchao Zhu、ZongYu Wang、Rui Xie、Wei Wu 和 Man Lan。2022。一种基于三阶张量同构的有效且高效的实体对齐解码算法。载于 *ACL*。5888–5898。[31] Xin Mao、Wenting Wang、Yuanbin Wu 和 Man Lan。2021。提升实体对齐速度 10×：采用归一化困难样本挖掘的双重注意力匹配网络。载于 *WWW*。821–832。[32] Xin Mao、Wenting Wang、Yuanbin Wu 和 Man Lan。2021。从对齐到指派：极其简单的无监督实体对齐。载于 *EMNLP*。2843–2853。[33] Xin Mao、Wenting Wang、Huimin Xu、Man Lan 和 Yuanbin Wu。2020。MRAEA：一种高效且稳健的跨语言知识图谱实体对齐方法。载于 *WSDM*。420–428。[34] Xin Mao、Wenting Wang、Huimin Xu、Yuanbin Wu 和 Man Lan。2020。关系反射实体对齐。载于 *CIKM*。1095–1104。[35] Gonzalo E. Mena、David Belanger、Scott W. Linderman 和 Jasper Snoek。2018。利用 Gumbel-Sinkhorn 网络学习潜在置换。载于 *ICLR*。[36] Alvin E Roth。2008。延迟接受算法：历史、理论、实践与未解问题。国际博弈论期刊36，3（2008），537–569。[37] Nino Shervashidze、Pascal Schweitzer、Erik Jan van Leeuwen、Kurt Mehlhorn 和 Karsten M. Borgwardt。2011。Weisfeiler-Lehman 图核。*JMLR* 12（2011），2539–2561。[38] Fabian M. Suchanek、Gjergji Kasneci 和 Gerhard Weikum。2007。Yago：语义知识的核心。载于 *WWW*。697–706。[39] Zequn Sun、Muhao Chen 和 Wei Hu。2021。识别无匹配：处理悬空情况的实体对齐。载于 *ACL*。3582–3593。[40] Zequn Sun、Muhao Chen、Wei Hu、Chengming Wang、Jian Dai 和 Wei Zhang。2020。利用双曲知识图谱嵌入进行知识关联。载于 *EMNLP*。5704–5716。[41] Zequn Sun、Wei Hu 和 Chengkai Li。2017。通过联合属性保持嵌入进行跨语言实体对齐。载于 *ISWC*。628–644。[42] Zequn Sun、Wei Hu、Qingheng Zhang 和 Yuzhong Qu。2018。利用知识图谱嵌入自举实体对齐。载于 *IJCAI*。4396–4402。[43] Zequn Sun、Chengming Wang、Wei Hu、Muhao Chen、Jian Dai、Wei Zhang 和 Yuzhong Qu。2020。采用门控多跳邻域聚合的知识图谱对齐网络。载于 *AAAI*。222–229。[44] Zequn Sun、Qingheng Zhang、Wei Hu、Chengming Wang、Muhao Chen、Farahnaz Akrami 和 Chengkai Li。2020。知识图谱基于嵌入的实体对齐基准测试研究。*PVLDB* 13，11（2020），2326–2340。[45] Bayu Distiawan Trisedya、Jianzhong Qi 和 Rui Zhang。2019。使用属性嵌入对齐知识图谱中的实体。载于 *AAAI*。297–304。[46] Petar Velickovic、Guillem Cucurull、Arantxa Casanova、Adriana Romero、Pietro Liò 和 Yoshua Bengio。2018。图注意力网络。载于 *ICLR*。[47] Denny Vrandecic 和 Markus Krötzsch。2014。维基数据：一个免费的协作式知识库。*Commun. ACM* 57，10（2014），78–85。[48] Pengfei Wang、Xiaocan Zeng、Lu Chen、Fan Ye、Yuren Mao、Junhao Zhu 和 Yunjun Gao。2022。PromptEM：面向低资源通用实体匹配的提示调优。*PVLDB* 16，2（2022），369–378。[49] Zhichun Wang、Qingsong Lv、Xiaohan Lan 和 Yu Zhang。2018。通过图卷积网络实现跨语言知识图谱对齐。载于 *EMNLP*。349–357。[50] Boris Weisfeiler 和 Andrei Leman。1968。将图化为规范形式及其中出现的代数。*NTI, Series* 2，9（1968），12–16。[51] Kexuan Xin、Zequn Sun、Wen Hua、Wei Hu、Jianfeng Qu 和 Xiaofang Zhou。2022。通过知识图谱合并、划分和嵌入实现大规模实体对齐。*arXiv preprint arXiv:2208.11125*（2022）。[52] Chenyan Xiong、Russell Power 和 Jamie Callan。2017。通过知识图谱嵌入实现学术搜索的显式语义排序。载于 *WWW*。1271–1279。[53] Chengjin Xu、Fenglong Su 和 Jens Lehmann。2021。用于时序知识图谱间实体对齐的时间感知图神经网络。载于 *EMNLP*。8999–9010。[54] Chengjin Xu、Fenglong Su、Bo Xiong 和 Jens Lehmann。2022。使用时序关系注意力进行时间感知实体对齐。载于 *WWW*。788–797。[55] Donghan Yu、Yiming Yang、Ruohong Zhang 和 Yuexin Wu。2021。基于知识嵌入的图卷积网络。载于 *WWW*。1619–1628。[56] Weixin Zeng、Xiang Zhao、Xinyi Li、Jiuyang Tang 和 Wei Wang。2022。大规模实体对齐。*VLDBJ*（2022），1–25。[57] Weixin Zeng、Xiang Zhao、Jiuyang Tang 和 Xuemin Lin。2020。通过自适应特征实现集体实体对齐。载于 *ICDE*。1870–1873。[58] Xin Zhang、Guangwei Xu、Yueheng Sun、Meishan Zhang、Xiaobin Wang 和 Min Zhang。2022。利用极高噪声众包标注识别中文意见表达。载于 *ACL*。2801–2813。[59] Xin Zhang、Guangwei Xu、Yueheng Sun、Meishan Zhang 和 Pengjun Xie。2021。作为领域自适应的众包学习：命名实体识别案例研究。载于 *ACL*。5558–5570。[60] Hao Zhu、Ruobing Xie、Zhiyuan Liu 和 Maosong Sun。2017。通过联合知识嵌入迭代式实体对齐。载于 *IJCAI*。4258–4264。

表5：示例时序知识图谱的ID分配 $G_{\star s}$ 和 $G_{\star t}$

ID	名称	Type
0	联邦法官（阿根廷）	实体
1	北欧理事会	
2	法国	
3	哈维尔·索拉纳	
4	日本	
5	乌克兰政府首脑	
0	拒绝司法合作	关系
1	提出申诉或请求	
2	签署正式协议	
3	接待访问	
4	进行访问	
0	2010-04-24	时间戳
1	2007-09-29	
2	2005-04-16	
3	2007-05-10	
4	2005-07-20	

计算示例

我们提供一个分步示例，以阐明图2所示的计算过程。我们将图2中的两个时序知识图谱记为 $G_{\star s}$ 和 $G_{\star t}$。首先，如表5所示，为实体、关系和时间戳分配唯一标识符。接下来，根据公式2统计出边和入边，并拼接矩阵。为简洁起见，我们省略入边的计算，仅展示出边的处理过程如下。

$$\boldsymbol{A}^t = \begin{bmatrix} .40 & .15 & .15 & .15 & .15 \\ .15 & .15 & .40 & .15 & .15 \\ .15 & .40 & .15 & .15 & .15 \\ .20 & .20 & .20 & .20 & .20 \\ .15 & .15 & .15 & .15 & .40 \\ .15 & .15 & .15 & .40 & .15 \end{bmatrix}, \tag{8}$$

其中， $\boldsymbol{A}^t$ 是使用 $G_{\star s}$ 构建的 $\boldsymbol{A}^t$。接下来，我们根据公式3构建关系邻接矩阵：

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{9}$$

邻接矩阵 $\boldsymbol{A}$ 仅包含0和1，因为图2中的每个关系只出现一次。因此，它们不会影响邻接矩阵的数值。通过设置 $L = 1$，我们得到时间特征 $\boldsymbol{H}^t$，它由 $\boldsymbol{A}^t$ 和 $\boldsymbol{A}\boldsymbol{A}^t$ 拼接而成（即 $\boldsymbol{H}^t = \left[\boldsymbol{A}^t || \boldsymbol{A}\boldsymbol{A}^t\right]$）：

$$\boldsymbol{H}^t = \begin{bmatrix} .40 & .15 & .15 & .15 & .15 & .15 & .40 & .15 & .15 & .15 \\ .15 & .15 & .40 & .15 & .15 & .15 & .40 & .15 & .15 & .15 \\ .15 & .40 & .15 & .15 & .15 & .20 & .20 & .20 & .20 & .20 \\ .20 & .20 & .20 & .20 & .20 & .00 & .00 & .00 & .00 & .00 \\ .15 & .15 & .15 & .15 & .40 & .15 & .15 & .15 & .40 & .15 \\ .15 & .15 & .15 & .40 & .15 & .15 & .15 & .15 & .15 & .40 \end{bmatrix} \tag{10}$$

接着，我们设置 $\boldsymbol{A}_s = \boldsymbol{A}_t = \boldsymbol{A}$ 和 $\boldsymbol{A}_s^t = \boldsymbol{A}_t^t = \boldsymbol{A}^t$，因为 $G_{\star s}$ 和 $G_{\star t}$ 在图2中是相同的。这会得到归一化的 WL 权重 $k^r = 1$ 和 $k^t = 1$。由于关系编码器涉及图神经网络的训练，为简单起见，我们随机初始化一个 $|E| \times 10$ 的嵌入：

$$\boldsymbol{H}_s^r = \boldsymbol{H}_t^r = \begin{bmatrix} .30 & .44 & .33 & .40 & .61 & .35 & .70 & .85 & .32 & .56 \\ .81 & .61 & .97 & .84 & .53 & .00 & .76 & .44 & .19 & .45 \\ .44 & .51 & .70 & .72 & .59 & .21 & .38 & .88 & .90 & .67 \\ .13 & .21 & .77 & .25 & .44 & .71 & .66 & .37 & .35 & .61 \\ .33 & .33 & .70 & .29 & .79 & .51 & .78 & .60 & .24 & .59 \\ .65 & .94 & .77 & 1.00 & .50 & .56 & .50 & .38 & .54 & .34 \end{bmatrix} \tag{11}$$

将关系编码器的权重设为 0 后，对齐矩阵 $\boldsymbol{P}_T$ 完全由时间特征决定：

$$\boldsymbol{P}_T = \boldsymbol{H}^t (\boldsymbol{H}^t)^T = \begin{bmatrix} .50 & .44 & .39 & .20 & .37 & .37 \\ .44 & .50 & .39 & .20 & .37 & .37 \\ .39 & .39 & .45 & .20 & .39 & .39 \\ .20 & .20 & .20 & .20 & .20 & .20 \\ .37 & .37 & .39 & .20 & .50 & .37 \\ .37 & .37 & .39 & .20 & .37 & .50 \end{bmatrix} \tag{12}$$

有了 $\boldsymbol{P}_T$，我们根据公式6得到 $\boldsymbol{D}_{G_s^\star G_t^\star} = 4\,61$。请注意，为简单起见，我们不考虑 Sinkhorn迭代及点积以外的其他过程。

类似地，将时间编码器和关系编码器的权重都设为1，得到 $\boldsymbol{P}$；通过平衡时间特征和关系特征获得的对齐矩阵计算如下：

$$\boldsymbol{P} = \boldsymbol{H}^t (\boldsymbol{H}^t)^T + \boldsymbol{H}_s^r (\boldsymbol{H}_t^r)^T = \begin{bmatrix} 2.70 & 2.72 & 3.00 & 2.24 & 2.72 & 2.81 \\ 2.72 & 3.97 & 3.41 & 2.42 & 2.98 & 3.75 \\ 3.00 & 3.41 & 4.03 & 2.59 & 3.02 & 3.67 \\ 2.24 & 2.42 & 2.59 & 2.49 & 2.61 & 2.60 \\ 2.72 & 2.98 & 3.02 & 2.61 & 3.04 & 2.97 \\ 2.81 & 3.75 & 3.67 & 2.60 & 2.97 & 4.26 \end{bmatrix} \tag{13}$$

这会导致 $\boldsymbol{D}_{G_s^\star G_t^\star} = 667.57$，其值显著大于4.61。这表明应为关系编码器分配较低的权重。在这种情况下，由于关系特征是随机生成的，因此更适合更多地考虑时间特征。仅使用时间编码器时，H@1为100%；而同时使用权重为1的时间编码器和关系编码器时，H@1会降至66.7%。这与观察到的距离值一致。

B 数据集统计

所有数据集的完整信息，包括 DICEWS、WY50K 和 WY20K，如表 6 所示。

• **DICEWS**源自 ICEWS05-15 [7], , 后者是较大规模 ICEWS [22] 中涵盖 2005 年至 2015 年时间段的事实子集。ICEWS05-15 [7] 被社区广泛用作时序知识图谱比较的基准数据集 [17]。数据生成遵循 [60] 中的方法。该数据集包含 8,566 个实体对，其全部事实均包含时间信息。DICEWS 中两个时序知识图谱 Q_s 与 Q_t 之间共享四元组的重叠率约为 50%，即 $2 * \frac{|Q_s \cap Q_t|}{|Q_s|+|Q_t|} \approx 0.5$。• **WY50K** 提取自维基数据 [47] 和 YAGO [38]。WY50K 包含 49,172 个实体对，其全部事实均包含时间信息。

• **WY20K** 也从维基数据 [47] 和 YAGO [38]中提取。不过，与 WY50K 不同，WY20K 是一个包含 19,462 个实体对的混合数据集，其中 YAGO 的事实有 17.5%、维基数据的事实有 36.6% 不包含时间信息。

表 6：DICEWS、WY50K 和 WY20K 的统计数据

数据集	#实体	#关系	#时间	#四元组
DICEWS	9,517-9,537	247-246	4,017	307,552-307,553
WY50K	49,629-49,222	11-30	245	221,050-317,814
WY20K	19,493-19,929	32-130	405	83,583-142,568

表7：可扩展性分析结果

	DICEWS					WY50K				
	20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
R	1.1	1.5	3.1	4.6	7.0	6.0	20.9	48.5	92.1	171.1
T	0.7	1.9	3.3	5.1	7.9	2.4	10.3	26.0	52.5	90.2
D	1.2	3.1	5.1	7.9	13.8	25.2	80.7	158.6	250.6	385.2

表8：开放世界实体对齐的结果

方法	未观测实体对			已观测实体对		
	H@1	H@10	平均倒数排名	H@1	H@10	平均倒数排名
DualMatch	**77.4**	**89.5**	**0.829**	**94.2**	**96.6**	**0.952**
TREA	34.2	74.8	0.479	54.9	82.5	0.643
TEA-GNN	15.5	59.0	0.324	39.2	74.8	0.513
RREA	7.5	48.3	0.253	36.1	58.0	0.407

C 抽样数据集上的可扩展性研究

为进一步研究 DualMatch 的可扩展性，我们按照 [9] 从 DICEWS 和 WY50K 中采样子图。我们将实体数量从 20% 调整到 100%，并在表7中记录运行时间（秒）；其中，R、T 和 D 分别表示关系编码器、时间编码器和 GM-Decoder。结果表明，DualMatch 具有很强的可扩展性。具体而言，随着实体数量增加，运行时间以二次速率增长，这符合 Sinkhorn算法的二次时间复杂度预期。尽管如此，即使使用 100% 的实体，运行时间仍相对较短：在 DICEWS 上，关系编码器、时间编码器和 GM-Decoder 分别耗时 7 秒、7.9 秒和 13.8 秒；在 WY50K 上，则分别耗时 171.1 秒、90.2 秒和 385.2 秒。需要注意的是，DualMatch 各组件的运行时间有所不同，其中 GM-Decoder 的运行时间最长。这表明，通过优化解码过程，仍有望进一步提升整个 DualMatch 的可扩展性。

总体而言，可扩展性研究表明，DualMatch 能够处理大型数据集，即使面对大规模数据，也能保持相对较低的运行时间。因此，DualMatch 非常适合需要高效且有效处理数据的现实世界应用。

D 开放世界实体对齐

现实世界中的大多数知识图谱都是动态的，随着时间推移，会不断出现新的实体和时间戳 [54]。这使开放世界实体对齐成为一项关键任务。然而，现有的大多数实体对齐方法都采用封闭世界假设，无法对齐新出现的实体[8, 29, 31, 34] [54]。

TREA 学习时间戳的嵌入，使其能够满足从实际时间戳到嵌入的特定映射函数，从而实现对新实体的实体对齐。然而，它只能粗略估计未来时间戳的嵌入，这可能导致后续预测出现误差。另一方面，DualMatch 主要采用无监督方式，可通过在线直接应用无监督组件，轻松适配开放世界实体对齐。随着新实体和时间戳不断出现，可以重新运行时间编码器和 GM-Decoder，以获得最准确的结果。

为了评估实体对齐在开放世界中的效果，我们按照 TREA 中的开放世界设置重新整理了 DICEWS 的四元组 [54]。两个训练集 $G_{\prime s}$ 和 $G_{\prime t}$ 包含来自 DICEWS 中 G_s 和 G_t 的四元组，涵盖 2014 年 1 月 1 日之前的数据。我们按照 [54]，从 $G_{\prime s}$ 和 $G_{\prime t}$ 中随机抽取 1,000 个实体对，用于训练关系编码器。未见实体和关系的嵌入采用随机初始化。

我们将 DualMatch 与三个最先进的方法进行比较：TREA [54]，TEA-GNN [53]，和 RREA [34]。结果如表8所示。在开放世界实体对齐任务中，DualMatch 显著优于基线方法：对于未观测实体对，H@1 提高了 43.2%；对于已观测实体对，H@1 提高了 34.9%。这是因为 DualMatch 的时序编码器和 GM-解码器均为无监督方法，无需训练即可轻松在线部署。

E 案例研究

我们在 DICEWS 上使用 DualMatch、Dual- Match-wt 和 DualMatch-wr 对时序知识图谱对齐进行案例研究（分别记为 $G_{\heartsuit s}$ 和 $G_{\heartsuit t}$）。其中，DualMatch-wt 表示不使用时间编码器的 DualMatch，DualMatch-wt 表示不使用关系编码器的 DualMatch。给定真实标注 {(Abderrahim, Abderrahim), (Looter, Looter)}，DualMatch 得出的结果与真实标注完全一致。DualMatch-wt 和 DualMatch-wr 得出的结果详述如下。

对于 $G_{\heartsuit s}$ 中的实体“Abderrahim”，DualMatch 将其正确地对齐到 $G_{\heartsuit t}$ 中的“Abderrahim”，而 DualMatch-wr 则得出了错误的对齐结果“Chief”。$G_{\heartsuit s}$ 中“Abderrahim”相关的四元组为 (Abderrahim, 签署正式协议, 外国, 2007-09-28)，而 $G_{\heartsuit t}$ 中“Chief”相关的四元组为 (Chief, 表示会面或谈判意向, 安哥拉, 2007-09-28)。以上案例表明，仅凭时间戳 2007-09-28 无法区分这两个实体。

对于 $G_{\heartsuit s}$ 中的实体“Looter”，DualMatch 能正确地将其与 $G_{\heartsuit t}$ 中的“Looter”对齐，而 DualMatch-wt 则错误地将其对齐为“Rioter”。$G_{\heartsuit s}$ 中“Looter”相关的四元组为 (Looter, 使用非常规暴力, 法国, 2005-11-09)，而 $G_{\heartsuit t}$ 中“Rioter”相关的四元组为 (Rioter, 使用非常规暴力, 警察, 2005- 03-14)。尽管 $G_{\heartsuit s}$ 和 $G_{\heartsuit t}$ 具有不同的三元组，关系编码器仍无法区分它们。这是因为我们仅使用 10% 的实体对来训练关系模型，这有时会导致不准确。

总之，DualMatch-wt 和 DualMatch-wr 与真实标注存在显著差异。这表明有必要开发并结合时序和关系编码器。