HySAE: An Efficient Semantic-Enhanced Representation Learning Model for Knowledge Hypergraph Link Prediction

Anonymous Author(s)

ABSTRACT

Representation learning technique is an effective link prediction paradigm to alleviate the incompleteness of knowledge hypergraphs. However, the n-ary complex semantic information inherent in knowledge hypergraphs causes existing methods to face the dual limitations of weak effectiveness and low efficiency. In this paper, we propose a novel knowledge hypergraph representation learning model, HySAE, which can achieve a satisfactory trade-off between effectiveness and efficiency. Concretely, HySAE builds an efficient semantic-enhanced *3D scalable end-to-end embedding architecture* to sufficiently capture knowledge hypergraph n-ary complex semantic information with fewer parameters, which can significantly reduce the computational cost of the model. In particular, we also design an efficient position-aware entity role semantic embedding way and two enhanced semantic learning strategies to further improve the effectiveness and scalability of our proposed method. Extensive experimental results on all datasets demonstrate that HySAE consistently outperforms state-of-the-art baselines, with an average improvement of 9.15%, a maximum improvement of 39.44%, an average 10.39x faster, and 75.79% fewer parameters. The code for our proposed method is available at this link https://anonymous.4open.science/r/HySAE-1026.

CCS CONCEPTS

• **Computing methodologies → Knowledge representation and reasoning**.

KEYWORDS

Knowledge Hypergraph, Knowledge Representation Learning, Link Prediction

ACM Reference Format:

Anonymous Author(s). 2025. HySAE: An Efficient Semantic-Enhanced Representation Learning Model for Knowledge Hypergraph Link Prediction. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (WWW '25)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Although knowledge graphs (KGs) have been widely used to improve technical applications in the Web community, the (h, r, t) triple structure cannot accurately and efficiently express semantic information about complex facts [21, 35]. Knowledge hypergraphs are prevalent in the real world, offering greater semantic expressiveness than traditional binary relational KGs [10, 32]. For instance, the mathematical community recognizes *Newton* and *Leibniz* as co-inventors of *Calculus*. The binary relational triple cannot represent this fact, but the n-ary knowledge hypergraph can be defined intuitively and clearly as `InventorOf`(*Calculus*, *Newton*, *Leibniz*). Furthermore, knowledge hypergraphs suffer from the same incompleteness issue as binary relational knowledge bases, which can be alleviated through link prediction (also known as knowledge completion) [11, 20, 29]. In recent years, knowledge hypergraph representation learning is an effective link prediction paradigm that has received extensive attention from academia and industry.

Limitations of existing methods. Since knowledge hypergraphs inherently have n-ary complex semantic information, this poses a dual challenge to the effectiveness and efficiency of representation learning models. Existing methods usually extend or refer to binary relational KG models to compute n-ary semantic structures, which cannot achieve end-to-end knowledge hypergraph representation learning [10, 14, 15, 19, 32, 36]. These solutions require separate decomposition operations for knowledge tuples of different arities, which inevitably destroys and loses the original semantic structure and cannot adequately capture knowledge hypergraph n-ary complex semantic information. Consequently, recent methods can only enhance effectiveness by increasing complex local structures, which requires more elaborate and potentially redundant model architectures [4, 11, 20, 28, 29]. This complexity undoubtedly leads to higher computational costs and parameter amounts, significantly compromising model efficiency. **Existing solutions do not provide a good trade-off between the effectiveness and efficiency of knowledge hypergraph representation learning.**

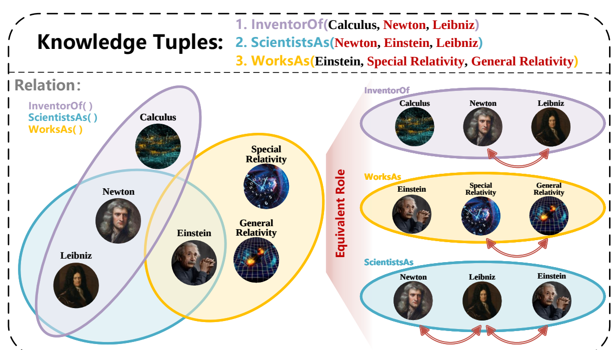


Figure 1: Real-world instances of knowledge hypergraphs where some entities have equivalent role semantics. Swapping the positions of entities with the equivalent role does not affect the correctness of the original knowledge tuple.

Our idea and solution. Current works assume that the n-ary relations in knowledge tuples have different semantic couplings

WWW '25, 28 April-2 May, 2025, Sydney, Australia

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

with each entity, and they need to capture the unique semantic information of all entities in the knowledge hypergraph to enhance effectiveness. However, capturing the unique semantic information of each entity equally and repetitively is not an efficient and reasonable way. As illustrated in Figure 1, many different entities in knowledge tuples have equivalent roles, meaning that replacing the positions of some entities does not affect the correctness of the original knowledge tuples. Furthermore, an analysis of several real-world knowledge hypergraph benchmark datasets reveals that entities are highly repetitive across different knowledge tuples, with the uniqueness ratio of entities being less than 10% [11, 20, 29]. Therefore, our solution treats all entities in a knowledge tuple equally, and the relations and all entities in each knowledge tuple should feature interactions together during representation learning. Importantly, knowledge tuples of different arities in a knowledge hypergraph are processed simultaneously, which can significantly reduce the computational cost and the number of model parameters.

Contributions. To address the above limitations, we propose an efficient **S**em**A**ntic-**E**nhanced knowledge **Hy**pergraph representation learning model, HySAE, which focuses on striking a superior trade-off between effectiveness and efficiency. HySAE designs a novel knowledge hypergraph *3D scalable end-to-end embedding architecture* that efficiently enhances the ability to capture n-ary complex semantic information from two perspectives. On the one hand, using semantic-enhanced 3D dilated convolutional neural networks in relation and entities feature plane dimensions can expand the semantic-aware region with fewer parameters to more adequately capture latent n-ary semantic information. Specifically, we propose two enhanced semantic learning strategies that provide a more powerful effectiveness advantage over traditional knowledge convolutional representation learning architectures [6, 18, 27]. On the other hand, in relation and entities feature interaction dimensions, efficient end-to-end knowledge hypergraph representation learning is achieved by scalably adjusting the structural parameters of the 3D embedding architecture to match the arity number of different knowledge tuples adaptively. In addition, based on the *3D scalable end-to-end embedding architecture*, we propose an efficient position-aware entity role semantic embedding way, which further improves the effectiveness and scalability of the HySAE model with fewer parameters.

Our contributions are summarized as follows:

- We propose a novel knowledge hypergraph representation learning model, HySAE, which constructs an efficient semantic-enhanced *3D scalable end-to-end embedding architecture* that realizes a better trade-off between model effectiveness and efficiency.
- HySAE designs an efficient position-aware entity role semantic embedding way and two enhanced semantic learning strategies, which can adequately capture n-ary complex semantic information with fewer parameters to improve the effectiveness and scalability of the model further.
- Extensive experimental results demonstrate the effectiveness and scalability of our proposed method, which consistently outperforms all state-of-the-art baselines. Compared with the best baseline, HySAE improved by an average of 9.15% and a maximum of 39.44% across all datasets.
- HySAE attains satisfactory model efficiency compared with state-of-the-art baselines. Across all datasets, HySAE reduces the number of model parameters by an average of 75.79%, speeds up by an average of 10.39x, and reduces memory usage by an average of 45.42%.

2 RELATED WORK

Existing knowledge hypergraph representation learning methods can be classified into three categories: translation-based, semantic matching, and neural network methods.

2.1 Translation-Based Methods

Models in this category are the classical methods for early knowledge hypergraph representation learning. m-TransH [32] is an n-ary extended variant of the binary relational KG model TransH [31]. By employing a fully connected network to incorporate the correlation of related entities into the loss function and enhance performance, RAE [36] expands upon the m-TransH model. Inspired by the RAE model, NaLP [15] explicitly models the relevance of all role-value pairs in n-ary relational facts. Translation-based methods are weakly expressive, limiting the types of relations for knowledge modeling, and the effectiveness of the model is no longer competitive with the current state-of-the-art models.

2.2 Semantic Matching Methods

HSimplE [10] is inspired by the binary relational KG model SimplE [17], which attempts to concatenate the embedding vectors of all positions to generate the entity representation. GETD [19] expands the TuckER [2] tensor decomposition model for binary relational KGs to n-ary knowledge hypergraphs. S2S [7] extends the GETD paradigm to support *mixed arity* knowledge hypergraphs. RAM [20] enhances the quality of entity representation in knowledge hypergraphs by constructing linear combinations of constrained entity role semantic vectors. PosKHG [4] construct role semantic vectors for each entity to create the role matrix, considerably increasing the number of model parameters. ReAlE [11] investigates knowledge hypergraph completion through the lens of relational algebra and its basic operations. Most semantic matching methods rely on linear model designs with tensor decomposition, which usually fail to capture the implicit deep complex semantic information.

2.3 Neural Network Methods

HypE [10] equates entity roles to entity position information and builds a separate convolutional layer and fully connected layer to capture entity role information. HyperMLN [3] proposes a Markov Logic network modeling framework to learn semantic using a variational EM algorithm. tNaLP+ [14] introduces type constraints on roles and role-value pairs using a convolutional neural network. RD-MPNN [37] is a knowledge hypergraph model based on relational message-passing neural networks. EnhancE [28] incorporates entity neighborhood information for a more expressive entity representation. HyConvE [29] jointly models n-ary complex semantics using different convolutional paths. The architecture of neural network models has become increasingly intricate in embedding richer semantic information, and the memory and time usage during training have severely constrained the real-world applications.

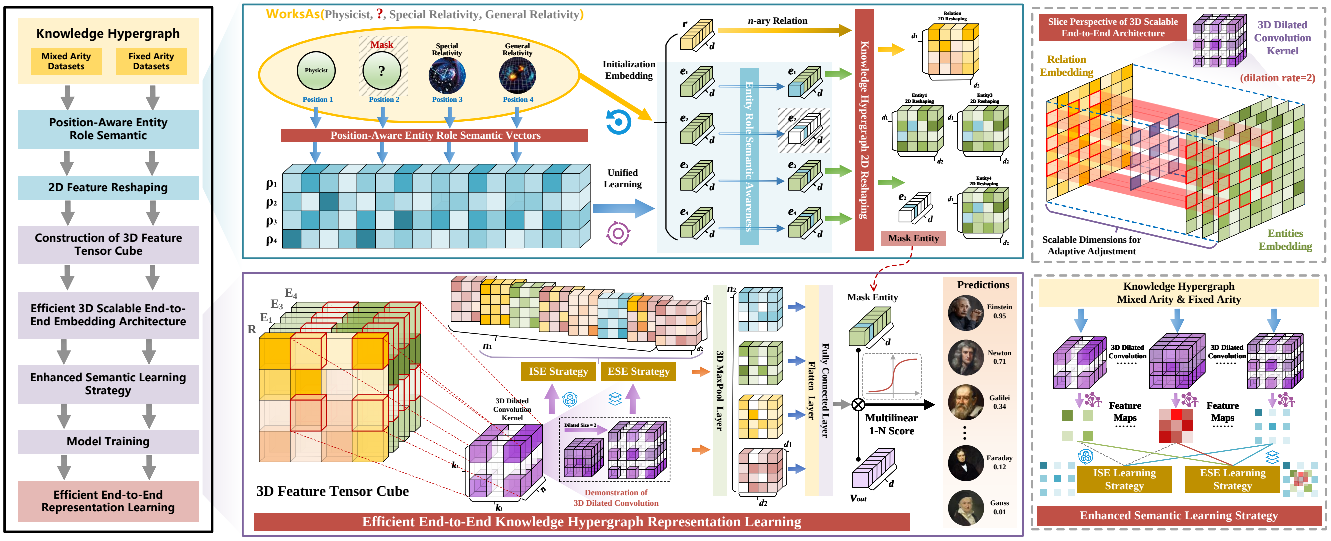


Figure 2: The overall framework of the HySAE model. On the right is a slice perspective of the 3D scalable end-to-end embedding architecture and the diagram of two enhanced semantic learning strategies.

3 PROBLEM STATEMENT

3.1 Knowledge Hypergraph

Given a finite set of entities $\mathcal{E}$, relations $\mathcal{R}$, and knowledge tuples $\mathcal{T}$, a knowledge hypergraph can be represented as $\mathcal{H} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$. Each observed fact is represented as a tuple $\mathcal{T}_i = r(e_1, e_2, ..., e_i, ..., e_n)$ in a knowledge hypergraph $\mathcal{H}$, where $\mathcal{T}_i \subseteq \mathcal{T}$, $r \in \mathcal{R}$ is a relation within the set of relations, $e_i \in \mathcal{E}$ is an entity within the set of entities, i is the position of entity in knowledge tuple, and n is the non-negative arity of relation r representing the number of entities involved within each relation. In case $n = 2$, $\mathcal{T}_i$ represents a binary relational fact. When $n \geq 2$, $\mathcal{T}_i$ represents a beyond-binary (n-ary) relational fact. KGs based on the binary relational triple $r(e_1, e_2)$ are a special case of knowledge hypergraphs.

3.2 Knowledge Representation Learning

Traditionally, knowledge representation learning generally projects entities and relations into a low-dimensional continuous latent space to perform downstream tasks. Knowledge hypergraph representation learning is an efficient link prediction method that essentially learns an n-ary knowledge tuple $\mathcal{T}_i = r(e_1, e_2, ..., e_i, ..., e_n)$ mapping function $f : \{r \mapsto \boldsymbol{r} \in \mathbb{R}^d; e_i \mapsto \boldsymbol{e_i} \in \mathbb{R}^d\}$, where $\boldsymbol{r}$ is the embedding of the n-ary relation r, $\boldsymbol{e_i}$ is the embedding of the entities e_i, and d is the embedding dimension size of the representation learning. The knowledge hypergraph link prediction task aims at predicting missing component in n-ary facts, where the missing component can be either an entity in the i-th position of the tuple $r(e_1, e_2, ..., ?, ..., e_n)$ or an n-ary relation $?(e_1, e_2, ..., e_i, ..., e_n)$.

4 THE HYSAE MODEL

The overall framework of HySAE, an efficient semantic-enhanced knowledge hypergraph representation learning model, is shown in Figure 2. Each core component of HySAE helps enhance the ability to capture n-ary complex semantic information while constraining the model to have fewer parameters, thus achieving a good trade-off between effectiveness and efficiency.

4.1 Position-Aware Entity Role Semantic

Previous research has shown that embedding entity role information in knowledge hypergraphs is an essential semantic-enhanced way to significantly improve the effectiveness of representation learning. As a result, the entity role semantic embedding modules of existing methods are usually designed to be complex and coupled, and the redundant model structures require high computational costs and parameter amounts [4, 10, 20]. However, most entities in real-world knowledge hypergraphs have equivalent role semantic information, and exchanging their positions does not affect the semantic correctness of the original knowledge tuple. Inspired by this observation, HySAE proposes an efficient position-aware entity role semantic embedding way, which can effectively capture entity role semantic information with fewer parameters.

Specifically, given an n-ary knowledge tuple $r(e_1, e_2, ..., e_n)$, we first randomly initialize the relation r and the entities e_i into d-dimensional embedding vectors $\boldsymbol{r} \in \mathbb{R}^d, \boldsymbol{e_i} \in \mathbb{R}^d$. Then, each entity in the n-ary knowledge tuple corresponds to a position ρ_i and is similarly initialized to the d-dimensional embedding vectors $\boldsymbol{\rho_i} \in \mathbb{R}^d$. The unified entity position embedding matrix $\mathbf{P}$ is constructed in the d-dimensional embedding space, defined as

$$\mathrm{P} = (\boldsymbol{\rho_1}, \boldsymbol{\rho_2}, ..., \boldsymbol{\rho_i}, ..., \boldsymbol{\rho_\alpha}) \tag{1}$$

where α denotes the maximum arity of the n-ary tuple in the knowledge hypergraph, that is, the maximum number of entity positions in the knowledge hypergraph. Although a knowledge hypergraph has a large number of different n-ary knowledge tuples, HySAE only learns α position-aware entity role semantic vectors. Obviously, $|n| \leq |\alpha| \ll |\mathcal{E}|$, so the position-aware entity role semantic embedding way of HySAE has minimal parameters.

Furthermore, the position-aware entity role semantic embedding way of HySAE is a decoupled structure, which facilitates the design of an efficient end-to-end knowledge hypergraph representation learning architecture. During knowledge hypergraph model training and reasoning, different n-ary knowledge tuples only need to integrate the same shared entity role semantic vector $\boldsymbol{\rho_i}$ according to the position ρ_i of each entity. HySAE uniformly learns and optimizes entity role semantic vectors in the process of learning entity and relation embedding vectors. The unified entity role semantic awareness way is represented as

$$\boldsymbol{e_i^{rol}} = \mathbf{RA}\left(\boldsymbol{e_i}, \mathbf{P}_{\boldsymbol{i}\cdot}\right) \tag{2}$$

where $\mathbf{P}_{\boldsymbol{i}\cdot} = \boldsymbol{\rho_i}$ as the i-th position unified embedding of the n-ary knowledge tuples, and $\boldsymbol{e_i^{rol}} \in \mathbb{R}^d$ denotes the entity embedding vector containing the role semantic information. $\mathbf{RA}(\cdot)$ is the role-aware operator, which can be flexibly selected according to different tasks. HySAE adopts a simple element-wise addition of the role-aware operator to improve knowledge hypergraph model efficiency.

4.2 Efficient 3D Scalable End-to-End Embedding

HySAE constructs a novel *3D scalable end-to-end embedding architecture* that efficiently and skillfully captures knowledge hypergraph n-ary complex semantic information. Compared with traditional knowledge hypergraph representation learning model architectures, the *3D scalable end-to-end embedding architecture* of HySAE can have a more powerful capability of n-ary latent semantic embedding with fewer model parameters. As with other knowledge representation learning methods [6, 27], the initial vectors of relations and entities need to be reshaped into 2D embedding matrices to enhance the interaction of feature information. The 2D reshaping embeddings of relations and entities are defined as

$$\overline{\boldsymbol{r}} = \Psi\left(\boldsymbol{r}\right),\ \overline{\boldsymbol{e}}_{\boldsymbol{i}}^{\boldsymbol{rol}} = \Psi\left(\boldsymbol{e_i^{rol}}\right) \tag{3}$$

where Ψ is a 2D reshaping function: $\{\mathbb{R}^d \mapsto \mathbb{R}^{d_1 \times d_2}\}$ transforms embeddings $\boldsymbol{r}$ and $\boldsymbol{e_i^{rol}}$ into matrices $\overline{\boldsymbol{r}} \in \mathbb{R}^{d_1 \times d_2}, \overline{\boldsymbol{e}}_{\boldsymbol{i}}^{\boldsymbol{rol}} \in \mathbb{R}^{d_1 \times d_2}$, and $d_1 \times d_2 = d$.

To further improve the degree of feature interaction between relations and entities in the knowledge hypergraph, HySAE concatenates 2D reshaping embeddings into the knowledge tuple 3D feature tensor cube. It is important to note that HySAE masks the entity e_m to be predicted, allowing 1-N multilinear scoring to accelerate knowledge hypergraph representation learning. The knowledge tuple 3D feature tensor cube $\boldsymbol{C_n}$, as follows

$$\boldsymbol{C_n} = \left(\overline{\boldsymbol{r}} || \overline{\boldsymbol{e}}_{\boldsymbol{1}}^{\boldsymbol{rol}} || \cdots || \overline{\boldsymbol{e}}_{\boldsymbol{m-1}}^{\boldsymbol{rol}} || \overline{\boldsymbol{e}}_{\boldsymbol{m+1}}^{\boldsymbol{rol}} || \cdots || \overline{\boldsymbol{e}}_{\boldsymbol{n}}^{\boldsymbol{rol}}\right) \tag{4}$$

where $||$ is the concatenation operation, and the relation 2D reshaping embedding $\overline{\boldsymbol{r}}$ and each entity 2D reshaping embeddings $\overline{\boldsymbol{e}}_{\boldsymbol{i}}^{\boldsymbol{rol}}$ in $\boldsymbol{C_n} \in \mathbb{R}^{d_1 \times d_2 \times n}$ generate feature interactions.

Existing research shows that convolutional embedding is a successful and efficient paradigm for knowledge representation learning [6, 10, 14, 27, 29]. Aiming at the inherent n-ary complex semantic features of knowledge hypergraphs, we propose a novel 3D dilated convolutional neural network based on the *3D scalable end-to-end embedding architecture* of HySAE for efficient knowledge hypergraph representation learning, defined as

$$\begin{aligned} \mathcal{F}(x, y, z) &= (\boldsymbol{C_n} * \boldsymbol{w_l})(x, y, z) \\ &= \sum_{a+lk_h=x} \sum_{b+lk_w=y} \sum_{c+lk_d=z} \boldsymbol{C_n}(a, b, c)\, \boldsymbol{w_l}(k_h, k_w, k_d) \end{aligned} \tag{5}$$

where $*$ is the 3D dilated convolution operation and $\boldsymbol{w_l}$ is the 3D dilated convolution kernel. l is the dilated size of the 3D dilated convolution, and the standard convolution can be equated to the dilated convolution with $l = 1$.

Notably, the *3D scalable end-to-end embedding architecture* of HySAE has three feature dimensions with a flexible and efficient operating space. In the feature dimension plane of relation and entities reshaping embeddings, HySAE employs 3D dilated convolution to expand the scale of the semantic-aware region without increasing the parameters, as follows

$$k_{h-l} = k_h + (k_h - 1)(l - 1),\ k_{w-l} = k_w + (k_w - 1)(l - 1) \tag{6}$$

where k_{h-l} and k_{w-l} are the semantic-aware region scale on the feature dimension plane of relation and entities reshaping embeddings. In this paper, set $k_h = k_w = k$, k is the size of the 3D dilated convolution kernel.

In relation and entities feature interaction dimensions, HySAE adaptively matches the arities of different n-ary knowledge tuples by scalably adjusting the structural parameters of the 3D embedding architecture. Thus, HySAE can embed knowledge tuples of different arities end-to-end together without the cost of redundant operations such as tuple decomposition and summation. In addition, the end-to-end embedding architecture allows relations to sufficiently interact with all entities in different n-ary knowledge tuples, which helps further enhance the effectiveness of the representation learning model. The feature maps obtained after *3D scalable end-to-end embedding architecture* are

$$\boldsymbol{\mathcal{F}_i} = \boldsymbol{C_n} * \boldsymbol{w_l}\ (k_h = k, k_w = k, k_d = n) \tag{7}$$

where n is the number of features in 3D feature tensor cube $\boldsymbol{C_n}$, i.e., the arity number of the n-ary knowledge tuples. $\boldsymbol{\mathcal{F}_i} \in \mathbb{R}^{d_1 \times d_2}$, $i = 1, 2, ..., n_1$, and n_1 is the number of output channels of the 3D dilated convolution, which is set to 8 in this paper.

4.3 Enhanced Semantic Learning Strategy

The *3D scalable end-to-end embedding architecture* of HySAE provides an excellent technical foundation for effectively and efficiently capturing knowledge hypergraph n-ary complex semantic information with a larger semantic-aware region. To further enhance the effectiveness of knowledge tuples of different arities, we design two enhanced semantic learning (ESL) strategies for HySAE to seamlessly capture knowledge hypergraph latent n-ary complex semantic information: *Internal Semantic Enhancement Learning Strategy (ISE)* and *External Semantic Enhancement Learning Strategy (ESE)*. The ISE learning strategy of HySAE is specifically defined as

$$\mathbf{ISE}\left(\boldsymbol{\mathcal{F}_i}\right) = \boldsymbol{W_\theta}\left(\boldsymbol{C_n} * \boldsymbol{w_l}(k = k_\theta; l = l_\theta)\right) \tag{8}$$

where $\mathbf{ISE}\left(\boldsymbol{\mathcal{F}_i}\right)$ are the feature maps obtained after *3D scalable end-to-end embedding architecture* using the ISE learning strategy. $\boldsymbol{W_\theta}$ denotes the linear transformation matrix used by HySAE to control the dimension size of the feature maps during model training.

Convolution kernel size k_θ and dilated size l_θ in the ISE learning strategy are both hyperparameters that can be optimized, but they only capture features within a single semantic-aware scale. Moreover, the ESE learning strategy captures potential features at multiple external semantic-aware scales, which is advantageous on knowledge hypergraph datasets with less n-ary semantic information. The ESE learning strategy is specifically defined as

$$\mathbf{ESE}\left(\boldsymbol{\mathcal{F}_i}\right) = \frac{1}{n}\sum_{j=1}^{n} \lambda_j \cdot \boldsymbol{W_j}\left(\boldsymbol{C_n} * \boldsymbol{w_l}(k = k_\theta; l = j)\right) \tag{9}$$

where $\mathbf{ESE}\left(\boldsymbol{\mathcal{F}_i}\right)$ are the feature maps obtained after *3D scalable end-to-end embedding architecture* using the ESE learning strategy. $\boldsymbol{W_j}$ denote the linear transformation matrices to ensure seamless integration of multi-scale features and λ_i is the multi-scale semantic learning weight coefficient.

Since HySAE always focuses on model efficiency to constrain the complexity of model architecture and the number of parameters, the ESE learning strategy is set to dual scales in this paper, i.e., l=1 and l>1. Similar to the ISE learning strategy, the l>1 scale size of the ESE learning strategy will become an optimizable hyperparameter that determines the final value during model training. Additionally, the pooling layer can effectively reduce the amount of parameters, accelerate model convergence, and prevent overfitting. Hence, HySAE uses a 3D max-pooling layer to extract salient features:

$$\boldsymbol{\mathcal{F}_i^{MP}} = \mathbf{3DMaxPool}\left(\mathbf{ESL}\left(\boldsymbol{\mathcal{F}_i}\right)\right) \tag{10}$$

where $\mathbf{ESL}\left(\boldsymbol{\mathcal{F}_i}\right)$ denotes the feature maps obtained after using enhanced semantic learning strategies (ISE or ESE) based on *3D scalable end-to-end embedding architecture*. The size of **3DMaxPool** layer is set to (4,1,1), $\boldsymbol{\mathcal{F}_i^{MP}} \in \mathbb{R}^{d_1 \times d_2}$, $i = 1, 2, ..., n_2$, and $n_2 = n_1/4$. Then, we concatenate and flatten the feature maps to output a d-dimensional vector through the fully connected layer, as follows

$$\boldsymbol{v_{out}} = \mathrm{FC}\left(\mathbf{Flatten}\left(\boldsymbol{\mathcal{F}_1^{MP}} || \boldsymbol{\mathcal{F}_2^{MP}} || \cdots || \boldsymbol{\mathcal{F}_{n_2}^{MP}}\right)\right) \tag{11}$$

where $\boldsymbol{v_{out}} \in \mathbb{R}^d$ is the output feature vector, $\mathbf{Flatten}(\cdot)$ denotes the flatten operation, $||$ is the concatenation operation.

4.4 Model Training

The 1-N multilinear scoring way has been proven to speed up model training in binary relational KG representation learning [6]. Inspired by previous work, we extend the 1-N multilinear scoring way to n-ary knowledge hypergraph representation learning to accelerate model training. Before constructing the scoring function, HySAE must integrate the masked entity e_m in the knowledge tuple with the role semantic vector of the corresponding position. Specifically, we treat each entity in the knowledge hypergraph as a candidate prediction entity set, and use the output feature vector $\boldsymbol{v_{out}}$ with each entity embedding to compute the knowledge tuple plausibility score $\phi(x)$, as follows

$$\phi(x) = \mathbf{softmax}\left(\boldsymbol{v_{out}} \cdot (\boldsymbol{e_m^{rol}})^{\mathrm{T}} + \boldsymbol{b}\right) \tag{12}$$

With the scoring function obtained above, we design the training loss as well as the learning objective for the HySAE model. As with the binary relational KG methods, generating negative samples during knowledge hypergraph model training helps to improve representation learning effectiveness and efficiency. However, observable instances in real-world knowledge hypergraph datasets are positive (true) samples. Thus, we employ a negative sampling strategy for the n-ary relational knowledge hypergraph representation learning model [20]. Specifically, for each positive (true) tuple $x \in \mathcal{T}$, we generate a set of negative samples as:

$$\bigcup_{i=1}^{n} \mathcal{N}_x^{(i)} \equiv \bigcup_{i=1}^{n} \{e_1, ..., \bar{e}_i, ..., e_n \notin \mathcal{T} | \bar{e}_i \in \mathcal{E}, \bar{e}_i \neq e_i\} \tag{13}$$

where $\mathcal{N}_x^{(i)}$ represents the set of knowledge tuples after replacing the i-th position entity, which is a generalization of the negative sampling strategy of the binary relational KG. Our model was trained using mini-batch Stochastic Gradient Descent and AdaGrad for tuning the learning rate [9]. The instantaneous multi-class log loss used by HySAE can be defined as

$$\mathcal{L} = \sum_{x \in \mathcal{T}} \sum_{i=1}^{n} -\log\left[e^{\phi(x)} / \left(e^{\phi(x)} + \sum_{y \in \mathcal{N}_x^{(i)}} e^{\phi(y)}\right)\right] \tag{14}$$

It can be concluded from the analysis that the instantaneous multi-class log loss is essentially the cross-entropy loss, which is a generalization of the binary cross-entropy loss commonly used in KG models. Furthermore, HySAE uses the necessary dropout [26] and batch normalization [16] during the convolution process to prevent model overfitting and stabilize convergence. For ease of description, we name the model using the ISE learning strategy HySAE-I and the model using the ESE learning strategy HySAE-E.

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Datasets. After a comprehensive investigation of previous works, the various experimental tasks of HySAE were conducted on seven public benchmark datasets of knowledge hypergraphs. The *mixed arity* knowledge hypergraph datasets are JF17K, WikiPeople, and FB-AUTO [20, 28, 29]. To further validate the scalability of the model, we conduct the *fixed arity* knowledge hypergraph task on four datasets: WikiPeople-3, JF17K-4, WikiPeople-4, and JF17K-5 [19]. A detailed summary of the datasets is provided in Table 1.

5.1.2 Baselines. To verify the effectiveness and efficiency of HySAE, we select 17 classical and state-of-the-art baseline methods for comparison, including translation-based models, semantic matching models, and neural network models. Specific comparison baseline methods are as follows:

- **Translation-Based Models:** RAE (2018), NaLP (2019).
- **Semantic Matching Models:** GETD (2020), n-CP (2020), n-TuckER (2020), HSimplE (2021), RAM (2021), S2S (2021), PosKHG (2023), ReAlE (2023).
- **Neural Network Models:** HypE (2021), HyperMLN (2022), tNaLP+ (2023), RD-MPNN (2023), EnhancE (2023), HyConvE (2023), and HySAE-2D.

To further prove the superiority of the knowledge hypergraph *3D scalable end-to-end embedding architecture*, we construct a 2D variant version of the HySAE model as a baseline. Since HySAE-2D cannot conduct end-to-end knowledge hypergraph representation learning, HySAE-2D adopts the ISE learning strategy to constrain the number of model parameters.

Table 1: Knowledge hypergraph dataset statistics. "Arity" denotes the involved arities of relations. "#Arity≥ 5" denotes the amount of facts with 5-ary relations and beyond.

Dataset		$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	Arity	#Train	#Valid	#Test	#Arity=2	#Arity=3	#Arity=4	#Arity≥ 5
Mixed *n*-ary	JF17K	28,645	322	2-6	61,104	15,275	24,568	54,627	34,544	9,509	2,267
	WikiPeople	47,765	707	2-9	305,725	38,223	38,281	337,914	25,820	15,188	3,307
	FB-AUTO	3,388	8	2, 4, 5	6,778	2,255	2,180	3,786	-	215	7,212
Fixed *n*-ary	WikiPeople-3	12,270	66	3	20,656	2,582	2,582	-	25,820	-	-
	JF17K-4	6,536	23	4	7,607	951	951	-	-	9,509	-
	WikiPeople-4	9,528	50	4	12,150	1,519	1,519	-	-	15,188	-
	JF17K-5	561	7	5	1,096	301	833	-	-	-	2,230

Table 2: Results of link prediction on mixed arity knowledge hypergraph datasets. The best results are in boldface, the second-best results are underlined, and the optimal baseline results are labeled with (). Experimental results for RAE, NaLP, HypE, and tNaLP+ are from [29], and other baseline experimental results are from the original paper. The experimental results not presented in the original paper and obtained locally are marked with "†".

Model	JF17K				WikiPeople				FB-AUTO			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
RAE [36]	0.392	0.312	0.433	0.561	0.253	0.118	0.343	0.463	0.703	0.614	0.764	0.854
NaLP [15]	0.310	0.239	0.334	0.450	0.338	0.272	0.362	0.466	0.672	0.611	0.712	0.774
HypE [10]	0.494	0.399	0.532	0.650	0.263	0.127	0.355	0.486	0.804	0.774	0.824	0.856
HSimplE [10]	0.472	0.408	0.538	0.656	0.227†	0.221†	0.298†	0.351†	0.798	0.766	0.821	0.855
RAM [20]	0.539	0.463	0.573	0.690	(0.380)	0.279	(0.445)	(0.539)	0.830	0.803	0.851	0.876
S2S [7]	0.528	0.457	0.570	0.690	0.372	0.277	0.439	0.533	-	-	-	-
HyperMLN [3]	0.556	(0.482)	0.597	0.717	-	-	-	-	0.831	0.803	0.851	0.877
tNaLP+ [14]	0.449	0.370	0.484	0.598	0.339	0.269	0.369	0.473	0.729	0.645	0.748	0.826
PosKHG [4]	0.545	0.469	0.582	0.706	0.315†	0.214†	0.377†	0.475†	0.856	0.821	0.876	0.895
EnhancE [28]	0.498	0.404	0.542	0.662	0.358	(0.285)	0.392	0.511	0.830	0.802	0.842	0.870
ReAlE [11]	0.559	(0.482)	0.594	0.705	0.332†	0.207†	0.417†	0.514†	(0.873)	(0.852)	(0.886)	(0.909)
RD-MPNN [37]	0.512	0.445	0.573	0.685	-	-	-	-	0.810	0.714	0.880	0.888
HyConvE [29]	(0.580)	0.478	(0.610)	(0.729)	0.362	0.275	0.388	0.501	0.847	0.820	0.872	0.901
HySAE-I (Ours)	**0.596**	**0.521**	**0.628**	**0.742**	**0.454**	<u>0.372</u>	**0.496**	<u>0.601</u>	<u>0.892</u>	<u>0.873</u>	<u>0.902</u>	**0.926**
HySAE-E (Ours)	<u>0.592</u>	<u>0.516</u>	<u>0.626</u>	<u>0.741</u>	**0.454**	**0.373**	<u>0.495</u>	**0.603**	**0.893**	**0.876**	**0.904**	<u>0.924</u>

5.1.3 Evaluation Metrics and Hyperparameters. Consistent with previous works [11, 20, 29], we use two standard evaluation metrics, namely mean reciprocal rank (MRR) and Hits@k, where k is set to 1, 3, and 10. In our experiments, the embedding dimension d is set to 400 and the batch size is taken from $\{64, 128, 256, 384, 512\}$. The size of the 3D dilated convolution kernel k is taken from 1 to 7, and the size of the convolution dilated l is taken from 1 to 5. Furthermore, the learning rate is selected from 0.00001 to 0.00100, the decay rate is chosen from 0.900 to 0.999, the dropout rate is selected from 0.0 to 0.9, and the multi-scale semantic learning weight coefficient λ_i is chosen from 0.1 to 0.9. The maximum number of model training iterations is set to 500 epochs.

5.2 Mixed Arity Knowledge Hypergraph Results

The experimental results of the *mixed arity* knowledge hypergraph link prediction are shown in Table 2, which demonstrate the effectiveness of our proposed model. **HySAE consistently outperforms all baseline models, with an average improvement of 8.19% and a maximum improvement of 30.88% over the optimal baseline model across all datasets and metrics.** In contrast, other baseline models do not obtain consistently good results and are less scalable on datasets of different sizes and semantic richness. Based on the semantic-enhanced *3D scalable end-to-end embedding architecture*, HySAE can sufficiently capture n-ary complex semantic information to ensure better effectiveness and scalability.

5.3 Fixed Arity Knowledge Hypergraph Results

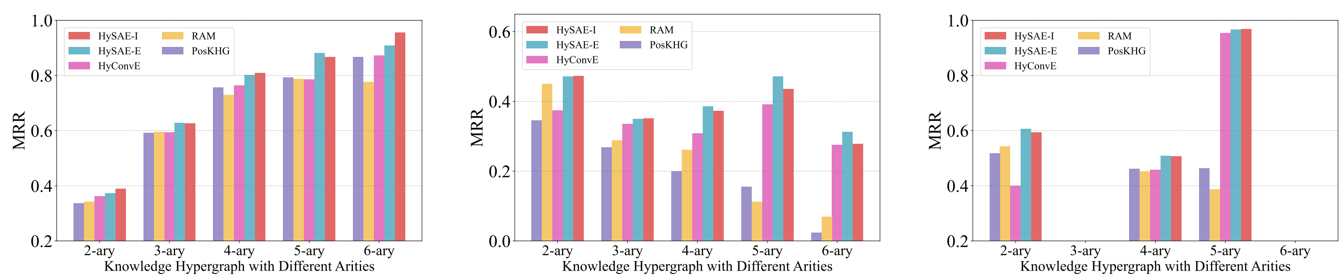
The experimental results of the *fixed arity* knowledge hypergraph link prediction are shown in Table 3, which demonstrate the effectiveness and scalability of our proposed model. **On the evaluation metrics across all datasets, HySAE improved by an average of 5.40% over the optimal baseline, with a maximum improvement of 14.21%.** GETD, n-CP, and n-TuckER can only handle *fixed arity* knowledge hypergraph task, and their poor model scalability is not conducive to real-world applications. HySAE is highly competitive on *fixed arity* knowledge hypergraph tasks, maintaining a significant advantage in model scalability. Furthermore, the performance of HySAE-E is more advantageous than HySAE-I on the *fixed arity* knowledge hypergraph task.

5.4 Performance Breakdown

The performance breakdown is an extension of knowledge hypergraph link prediction tasks, which can more comprehensively

Table 3: Results of link prediction on fixed arity knowledge hypergraph datasets. The best results are in boldface, the second-best results are underlined, and the optimal baseline results are labeled with (). Experimental results of RAE, NaLP, n-CP, n-TuckER, and GETD are from [19], and other baseline experimental results are obtained locally.

Model	WikiPeople-3			JF17K-4			WikiPeople-4			JF17K-5		
	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
RAE [36]	0.239	0.168	0.379	0.707	0.636	0.835	0.150	0.080	0.273	-	-	-
NaLP [15]	0.301	0.226	0.445	0.719	0.673	0.805	0.342	0.237	0.540	-	-	-
n-CP [19]	0.330	0.250	0.496	0.787	0.733	0.890	0.265	0.169	0.445	-	-	-
n-TuckER [19]	0.365	0.274	0.548	0.804	0.748	0.902	0.362	0.246	0.570	-	-	-
GETD [19]	(0.373)	(0.284)	(0.558)	0.810	0.755	0.913	(0.386)	(0.265)	(0.596)	-	-	-
HypE [10]	0.300	0.219	0.468	0.743	0.671	0.870	0.281	0.181	0.479	0.629	0.506	0.870
RAM [20]	0.321	0.250	0.462	0.790	0.739	0.883	0.200	0.135	0.328	(0.799)	(0.725)	0.924
ReAlE [11]	0.310	0.220	0.495	0.710	0.647	0.834	0.264	0.140	0.517	0.786	0.679	(0.941)
HyConvE [29]	0.320	0.257	0.498	(0.823)	(0.770)	(0.922)	0.321	0.195	0.576	0.791	0.722	0.930
HySAE-I (Ours)	0.386	0.292	0.577	0.833	0.779	**0.932**	0.403	0.278	**0.639**	0.886	0.827	0.984
HySAE-E (Ours)	**0.389**	**0.297**	**0.578**	**0.834**	**0.780**	0.928	**0.410**	**0.289**	0.636	**0.887**	**0.828**	**0.985**



(a) JF17K Dataset (b) WikiPeople Dataset (c) FB-AUTO Dataset

Figure 3: Results of knowledge hypergraph performance breakdown.

evaluate the effectiveness and scalability of knowledge hypergraph representation learning models [20, 29]. The experimental results of knowledge hypergraph performance breakdown are shown in Figure 3 and Table 6 (in the Appendix). **HySAE consistently outperformed all baseline models across all datasets, with a significant performance advantage of 10.59% on average and 39.44% on maximum improvement over the optimal baseline.** Moreover, we find that the effectiveness of HySAE-I using the ISE learning strategy is generally better on low-arity knowledge tuples. Conversely, the effectiveness of HySAE-E using the ESE learning strategy is superior in terms of high-arity knowledge tuples.

Table 4: Results of the number of model parameters (Millions). The best results are in boldface and the second-best results are underlined.

Model	JF17K	WikiPeople	FB-AUTO
HypE [10]	≈ 6.41M	≈ 10.25M	≈ 3.84M
RAM [20]	≈ 14.24M	≈ 27.34M	≈ 1.63M
PosKHG [4]	≈ 14.34M	≈ 27.53M	≈ 1.65M
HyConvE [29]	≈ 12.80M	≈ 21.44M	≈ 4.80M
ReAlE [11]	≈ 14.88M	≈ 29.61M	≈ 1.64M
HySAE-2D	≈ 11.62M	≈ 19.30M	≈ 3.94M
HySAE-I (Ours)	**≈ 1.38M**	**≈ 2.34M**	**≈ 1.06M**
HySAE-E (Ours)	≈ 2.76M	≈ 4.68M	≈ 2.12M

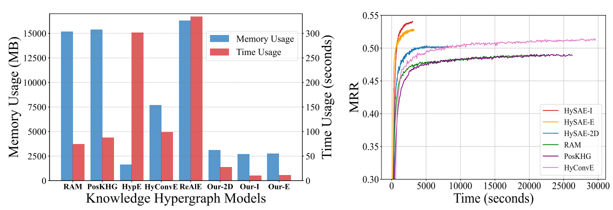
5.5 Model Efficiency Comparison

The model efficiency comparison was performed on *mixed arity* knowledge hypergraph datasets: JF17K, WikiPeople, and FB-AUTO. The state-of-the-art baseline models selected for comparison are HypE, RAM, PosKHG, HyConvE, ReAlE, and HySAE-2D. Through the investigation of existing works, we select the model efficiency evaluation metrics as the number of model parameters [1, 24], time and memory usage [8, 23, 34], and model training efficiency curve [7, 20]. Additionally, the time usage refers to the time required for each epoch iteration of models, calculated from the average of 10 epochs iteration time. The time usage contains the training, validation, and testing time of the knowledge hypergraph representation learning models. The training efficiency curves are plotted from 300 epochs iteration of knowledge hypergraph models. All experimental results are obtained in locally consistent software and hardware environment.

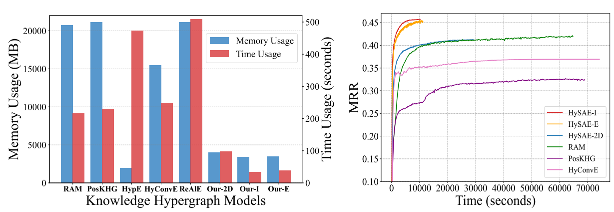
The results of the number of model parameters are shown in the Table 4. **Compared with the baseline models and HySAE-2D, HySAE reduces the parameters by an average of 75.79% and 83.03%, respectively.** The results of the model time usage, memory usage, and training efficiency curves are shown in Figure 4. **Compared with the baseline models and HySAE-2D, HySAE is on average 10.39x and 2.53x faster, and memory usage is on average 45.42% and 21.84% lower, respectively, with significantly faster and better convergence curves.** The training efficiency

Table 5: Results of ablation study. The best results are in boldface.

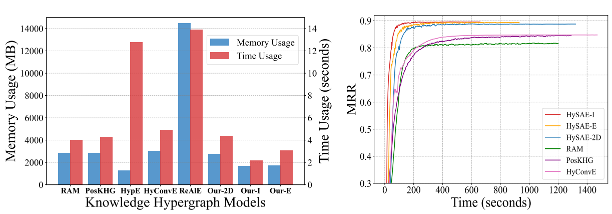
Model	JF17K				WikiPeople				FB-AUTO			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
HySAE-I	**0.596**	**0.521**	**0.628**	**0.742**	**0.454**	**0.372**	**0.496**	**0.601**	**0.892**	**0.873**	**0.902**	**0.926**
w/o Role Semantic	0.561	0.485	0.592	0.704	0.436	0.351	0.480	0.594	0.884	0.864	0.897	0.918
w/o 3D Architecture	0.556	0.483	0.588	0.696	0.409	0.329	0.450	0.558	0.885	0.868	0.893	0.919
HySAE-E	**0.592**	**0.516**	**0.626**	**0.741**	**0.454**	**0.373**	**0.495**	**0.603**	**0.893**	**0.876**	**0.904**	**0.924**
w/o Role Semantic	0.580	0.505	0.614	0.728	0.438	0.352	0.481	0.596	0.882	0.863	0.893	0.915
w/o ESE(l = 1)	0.574	0.498	0.608	0.725	0.441	0.354	0.486	0.597	0.892	0.875	0.902	0.923
w/o ESE(l > 1)	0.588	0.510	0.625	0.740	0.442	0.353	0.489	0.600	0.891	0.873	0.901	0.923



(a) Time Usage, Memory Usage, and Training Efficiency Curves for the JF17K Dataset



(b) Time Usage, Memory Usage, and Training Efficiency Curves for the WikiPeople Dataset



(c) Time Usage, Memory Usage, and Training Efficiency Curves for the FB-AUTO Dataset

Figure 4: Results of model efficiency comparison.

curve does not include HypE and ReAlE because their model speed is much slower than other methods. The model efficiency comparison shows the superiority and efficiency of the *3D scalable end-to-end embedding architecture*, which is highly competitive and valuable in real-world applications.

5.6 Ablation Study

The ablation study experiments were conducted on three datasets (JF17K, WikiPeople, and FB-AUTO) with hyperparameters consistent with the knowledge hypergraph link prediction task. To compare the ablation results more clearly, separate ablation study experiments are conducted for HySAE using different learning strategies (HySAE-I and HySAE-E). Specifically, the ablation experiments of HySAE-I include removing entity role semantic information (*w/o Role Semantic*) and removing 3D embedding architecture (*w/o 3D Architecture*). Notably, the ablation results of (*w/o 3D Architecture*) are equivalent to HySAE-2D. Since the core of HySAE-E lies in multi-scale features learning and integration, its ablation experiments focus on removing different scales, i.e., (*w/o ESE(l=1)*) and (*w/o ESE(l>1)*). In order to contrast with the ablation study of HySAE-I, (*w/o ESE(l=1)*) and (*w/o ESE(l>1)*) retain the 3D embedding architecture and also perform the ablation study of removing entity role semantic information (*w/o Role Semantic*).

The experimental results are shown in Table 5, and HySAE decreases performance on all evaluation metrics when removing any of the core components. The removal of entity role semantic information reduces the performance of HySAE-I and HySAE-E by an average of 3.42% and 2.16% across all datasets and metrics, respectively. It can be seen from the ablation results that the performance gains from the JF17K and WikiPeople datasets with rich entity role semantic information are more prominent. In addition, removing the 3D embedding architecture reduces HySAE-I performance by 5.63% on average. Removing the multi-scale features integration reduces the (*w/o ESE(l=1)*) and (*w/o ESE(l>1)*) performance by 1.91% and 1.07%, respectively. From the results of the ablation study experiments, it is further evident that the 3D embedding architecture of HySAE plays a substantial role in the knowledge hypergraph representation learning performance. Combined with the experimental results of model efficiency, removing 3D embedded architecture can also slow the model speed by 2.53x and increase the number of model parameters by 83.03%. In summary, the *3D scalable end-to-end embedding architecture* is a crucial foundation for realizing a satisfactory trade-off between the effectiveness and efficiency of knowledge hypergraph representation learning.

6 CONCLUSION

In this paper, we propose a novel knowledge hypergraph representation learning model, HySAE, which designs the semantic-enhanced *3D scalable end-to-end embedding architecture* to efficiently and adequately capture n-ary complex semantic information, achieving a superior trade-off between model effectiveness and efficiency. Extensive experimental tasks on datasets with different data sizes and semantic information richness prove the superior scalability and efficiency of our proposed method, and HySAE consistently outperforms all baseline models on all metrics. Besides, the ESE learning strategy is essentially multi-scale learning and integration, and we control dual-scale learning to simplify the model architecture and parameters. However, the number of scales for feature learning and integration is an open question, and its relationship with model effectiveness and efficiency is worthy of future work.

REFERENCES

[1] Peyman Baghershahi, Reshad Hosseini, and Hadi Moradi. 2023. Self-attention presents low-dimensional knowledge graph embeddings for link prediction. *Knowledge-Based Systems* 260 (2023), 110–124.

[2] Ivana Balažević, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (EMNLP'19)*. 5185–5194.

[3] Zirui Chen, Xin Wang, Chenxu Wang, and Jianxin Li. 2022. Explainable Link Prediction in Knowledge Hypergraphs. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM'22)*. 262–271.

[4] Zirui Chen, Xin Wang, Chenxu Wang, and Zhao Li. 2023. PosKHG: A Position-Aware Knowledge Hypergraph Model for Link Prediction. *Data Science and Engineering* 8, 2 (2023), 135–145.

[5] Chanyoung Chung, Jaejun Lee, and Joyce Jiyoung Whang. 2023. Representation learning on hyper-relational and numeric knowledge graphs with transformers. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD'23)*. 310–322.

[6] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI'18)*. 1811–1818.

[7] Shimin Di, Quanming Yao, and Lei Chen. 2021. Searching to sparsify tensor decomposition for n-ary relational data. In *Proceedings of The Web Conference 2021 (WWW'21)*. 4043–4054.

[8] Sicong Dong, Xupeng Miao, Pengkai Liu, Xin Wang, Bin Cui, and Jianxin Li. 2022. Het-kg: Communication-efficient knowledge graph embedding training via hotness-aware cache. In *2022 IEEE 38th International Conference on Data Engineering (ICDE) (ICDE'22)*. 1754–1766.

[9] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, 7 (2011).

[10] Bahare Fatemi, Perouz Taslakian, David Vazquez, and David Poole. 2021. Knowledge hypergraphs: prediction beyond binary relations. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (IJCAI'21)*. 2191–2197.

[11] Bahare Fatemi, Perouz Taslakian, David Vazquez, and David Poole. 2023. Knowledge hypergraph embedding meets relational algebra. *Journal of Machine Learning Research* 24, 105 (2023), 1–34.

[12] Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. 2020. Message Passing for Hyper-Relational Knowledge Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. 7346–7359.

[13] Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2020. NeuInfer: Knowledge inference on n-ary facts. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. 6141–6151.

[14] Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2023. Link prediction on n-ary relational data based on relatedness evaluation. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (2023), 672–685.

[15] Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2019. Link prediction on n-ary relational data. In *Proceedings of the 2019 World Wide Web Conference (WWW'19)*. 583–593.

[16] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning (ICML'15)*. 448–456.

[17] Seyed Mehran Kazemi and David Poole. 2018. SimplE embedding for link prediction in knowledge graphs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS'18)*. 4289–4300.

[18] Brenden M Lake and Marco Baroni. 2023. Human-like systematic generalization through a meta-learning neural network. *Nature* 623, 7985 (2023), 115–121.

[19] Yu Liu, Quanming Yao, and Yong Li. 2020. Generalizing tensor decomposition for n-ary relational knowledge bases. In *Proceedings of The Web Conference 2020 (WWW'20)*. 1104–1114.

[20] Yu Liu, Quanming Yao, and Yong Li. 2021. Role-aware modeling for n-Ary relational knowledge bases. In *Proceedings of The Web Conference 2021 (WWW'21)*. 2660–2671.

[21] Xiao Long, Liansheng Zhuang, Aodi Li, Houqiang Li, and Shafei Wang. 2024. Fact Embedding through Diffusion Model for Knowledge Graph Completion. In *Proceedings of the ACM on Web Conference 2024 (WWW'24)*. 2020–2029.

[22] Haoran Luo, E Haihong, Yuhao Yang, Yikai Guo, Mingzhi Sun, Tianyu Yao, Zichen Tang, Kaiyang Wan, Meina Song, and Wei Lin. 2023. HAHE: Hierarchical Attention for Hyper-Relational Knowledge Graphs in Global and Local Level. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL'23)*. 8095–8107.

[23] Xiao Qin, Nasrullah Sheikh, Chuan Lei, Berthold Reinwald, and Giacomo Domeniconi. 2023. Seign: A simple and efficient graph neural network for large dynamic graphs. In *2023 IEEE 39th International Conference on Data Engineering (ICDE) (ICDE'23)*. 2850–2863.

[24] Feiliang Ren, Juchen Li, Huihui Zhang, Shilei Liu, Bochao Li, Ruicheng Ming, and Yujia Bai. 2020. Knowledge Graph Embedding with Atrous Convolution and Residual Learning. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING'20)*. 1532–1543.

[25] Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In *Proceedings of The Web Conference 2020 (WWW'20)*. 1885–1896.

[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[27] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. 2020. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI'20)*. 3009–3016.

[28] Chenxu Wang, Zhao Li, Xin Wang, and Zirui Chen. 2023. EnhancE: Enhanced Entity and Relation Embedding for Knowledge Hypergraph Link Prediction. In *Companion Proceedings of the ACM Web Conference 2023 (WWW'23)*. 115–118.

[29] Chenxu Wang, Xin Wang, Zhao Li, Zirui Chen, and Jianxin Li. 2023. HyConvE: A Novel Embedding Model for Knowledge Hypergraph Link Prediction with Convolutional Neural Networks. In *Proceedings of the ACM Web Conference 2023 (WWW'23)*. 188–198.

[30] Quan Wang, Haifeng Wang, Yajuan Lyu, and Yong Zhu. 2021. Link Prediction on N-ary Relational Facts: A Graph-based Approach. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 (ACL'21)*. 396–407.

[31] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI'14)*. 1112–1119.

[32] Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. 2016. On the representation and embedding of knowledge bases beyond binary relations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16)*. 1300–1307.

[33] Weijian Yu, Jie Yang, and Dingqi Yang. 2024. Robust Link Prediction over Noisy Hyper-Relational Knowledge Graphs via Active Learning. In *Proceedings of the ACM on Web Conference 2024 (WWW'24)*. 2282–2293.

[34] Chenyang Zhang, Feng Zhang, Kuangyu Chen, Mingjun Chen, Bingsheng He, and Xiaoyong Du. 2023. EdgeNN: Efficient Neural Network Inference for CPU-GPU Integrated Edge Devices. In *2023 IEEE 39th International Conference on Data Engineering (ICDE) (ICDE'23)*. 1193–1207.

[35] Honggen Zhang, June Zhang, and Igor Molybog. 2024. HaSa: Hardness and Structure-Aware Contrastive Knowledge Graph Embedding. In *Proceedings of the ACM on Web Conference 2024 (WWW'24)*. 2116–2127.

[36] Richong Zhang, Junpeng Li, Jiajie Mei, and Yongyi Mao. 2018. Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In *Proceedings of the 2018 World Wide Web Conference (WWW'18)*. 1185–1194.

[37] Xue Zhou, Bei Hui, Ilana Zeira, Hao Wu, and Ling Tian. 2023. Dynamic relation learning for link prediction in knowledge hypergraphs. *Applied Intelligence* 53, 22 (2023), 26580–26591.

A MODEL DETAILS

Without a special notation introduction, this paper uses lowercase letters for scalars, bold lowercase letters for vectors, and bold uppercase letters for matrices (or 3D tensors). Algorithm 1 summarizes the training procedure for HySAE.

Algorithm 1: Training procedure for HySAE

```
Input: The n-ary knowledge hypergraph H = (E, R, T),
       the negative sampling rate N, the maximum number
       of iterations n_iter = 500
Output: The score of each knowledge tuple
Init: Entity embedding matrix E for e_i ∈ E, Relation
      embedding matrix R for r_i ∈ R, Unified entity position
      embedding matrix P
for i = 1, 2, ··· , n_iter do
    Sample a mini-batch T_batch ∈ T;
    for each fact x := {r, e_1, e_2, ..., e_n} ∈ T_batch do
        Construct negative samples for fact x;
        e_i^rol ← knowledge hypergraph position-aware
        entity role semantic embedding using (2);
        C_n ← construct n-ary knowledge tuple 3D feature
        tensor cube using (4);
        F_i ← get the feature maps using the 3D scalable
        end-to-end embedding architecture (7);
        ISE (F_i) ← HySAE-I get the feature maps using the
        internal semantic enhancement learning strategy (8);
        ESE (F_i) ← HySAE-E get the feature maps using
        the external semantic enhancement learning
        strategy (9);
        φ(x) ← get the final score of each knowledge tuple
        using 1-N multilinear scoring (12);
    end
    Update learnable parameters w.r.t. gradients based on
    the whole objective in (14);
end
```

B EXPERIMENTAL DETAILS

B.1 Datasets

The seven datasets used in the experimental section need some additional details. The WikiPeople dataset contains abundant n-ary semantic information, and it has the most significant number of relations and entities, i.e., high relation-specific and entity-specific. Also, the WikiPeople dataset has the highest number of relation-arity, which facilitates the performance release of role-aware or position-aware knowledge hypergraph representation learning models. In contrast, the FB-AUTO dataset has the least relation-specific and entity-specific, as well as the lowest number of relation-arity. The n-ary semantic information inherent in the JF17K dataset lies between FB-AUTO and WikiPeople. Since JF17K dataset lacks a validset, we randomly select 20% of the train set as validation [20]. Other datasets follow the split of the corresponding original papers. In addition, WikiPeople-3, JF17K-4, and WikiPeople-4 are from GETD [19], and JF17K-5 is extracted directly from the source data (JF17K dataset) by us.

B.2 Baseline Models

All local experiments were obtained on 4 NVIDIA GeForce RTX 3090 GPUs and PyTorch 1.12.0, ensuring a consistent hardware and software environment. The specific hyperparameters for HySAE are given in our code link, and the training iteration is terminated during the model training process if the MRR metric does not improve for 50 epochs continuously. Additionally, the baseline models used for the local experiments all use the open-source code given in the original paper. For the hyperparameters of the baseline methods, we follow the notation from the original papers.

- **RAM [20].** Details of the main hyperparameters: $m = 2$, $K = 10$, the batch size is set to 64, the embedding size is selected from $\{25, 50\}$, the decay rate is selected from $\{0.995, 0.990\}$, dropout is selected from $\{0.0, 0.2, 0.4\}$, and the learning rate is selected from $\{0.005, 0.003, 0.002, 0.001\}$.
- **HSimplE [10].** The original paper does not give hyperparameter information, and we use the hyperparameters given in the open-source code: the batch size is set to 128, the embedding size is set to 200, the negative ratio is set to 10, and the learning rate is set to 0.01.
- **PosKHG [4].** The original paper does not give hyperparameter information, and we use the hyperparameters given in the open-source code: $m = 2$, $K = 10$, the batch size is set to 64, the embedding size is selected from $\{25, 50\}$, the decay rate is set to 0.995, dropout is selected from $\{0.0, 0.2, 0.4\}$, and the learning rate is selected from $\{0.005, 0.003\}$.
- **ReAlE [11].** Details of the main hyperparameters: the learning rate is set to 0.08, the window size is set to 2, the negative ratio is selected from $\{10, 100\}$, and the batch size is selected from $\{128, 512\}$.
- **HypE [10].** The original paper does not give hyperparameter information, and we use the hyperparameters given in the open-source code: $filt\_w = 1$, $out\_channels = 6$, $stride = 2$, the batch size is set to 128, the embedding size is set to 200, the negative ratio is set to 10, and the learning rate is set to 0.1.
- **HyConvE [29].** Details of the main hyperparameters: the batch size is set to 128, the embedding size is set to 400, dropout is selected from $\{0.0, 0.2, 0.3, 0.4\}$, and the learning rate is selected from $\{0.01, 0.005, 0.003, 0.001, 0.0005, 0.0001\}$.

It is important to note that our research problem is fundamentally different from hyper-relational KGs [11, 20, 29]. Hyper-relational KGs are still based on the primary triple (h, r, t) as the core semantic structure, and their model expression ability and computational efficiency are greatly limited [5, 12, 13, 22, 25, 30, 33]. Knowledge hypergraphs abandon the traditional triple structure and have more substantial semantic expression capability. In addition, the format and details of knowledge hypergraph datasets are significantly different from hyper-relational KG datasets. Consequently, the hyper-relational KG methods are not the baseline model for this paper.

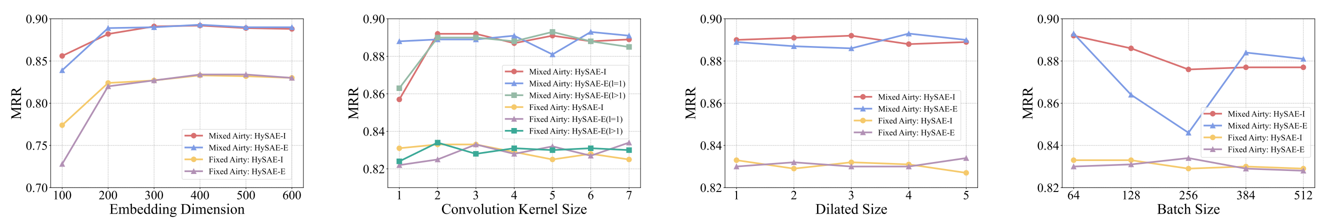
C EXPERIMENTAL SUPPLEMENT

C.1 Performance Breakdown Supplement

The performance breakdown experiment trains the model using all knowledge tuples in the *mixed arity* knowledge hypergraph, and

Table 6: Results of performance breakdown. The best results are in boldface and the second-best results are underlined.

Breakdown Arity	Model	JF17K				WikiPeople				FB-AUTO			
		MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
2-ary	RAM [20]	0.342	0.253	0.374	0.523	0.450	0.372	0.472	0.577	0.542	0.496	0.587	0.621
	PosKHG [4]	0.337	0.250	0.368	0.509	0.346	0.238	0.428	0.505	0.517	0.445	0.566	0.630
	HyConvE [29]	0.362	0.269	0.395	0.550	0.374	0.304	0.457	0.510	0.400	0.329	0.430	0.528
	HySAE-I (Ours)	**0.389**	**0.295**	**0.419**	**0.583**	**0.473**	**0.396**	**0.516**	**0.608**	0.594	0.522	0.631	**0.722**
	HySAE-E (Ours)	0.372	0.279	0.404	0.565	0.472	**0.396**	0.513	0.604	**0.606**	**0.545**	**0.641**	0.717
3-ary	RAM [20]	0.596	0.522	0.617	0.743	0.289	0.219	0.334	0.431	-	-	-	-
	PosKHG [4]	0.591	0.516	0.604	0.741	0.268	0.189	0.317	0.433	-	-	-	-
	HyConvE [29]	0.594	0.526	0.615	0.723	0.336	0.252	0.350	0.500	-	-	-	-
	HySAE-I (Ours)	0.626	**0.556**	0.655	0.763	**0.352**	**0.263**	0.379	0.533	-	-	-	-
	HySAE-E (Ours)	**0.628**	0.553	**0.662**	**0.773**	0.350	0.256	**0.380**	**0.535**	-	-	-	-
4-ary	RAM [20]	0.729	0.679	0.759	0.817	0.261	0.170	0.280	0.447	0.453	0.367	0.501	0.617
	PosKHG [4]	0.757	0.698	0.774	0.864	0.200	0.110	0.217	0.413	0.461	0.379	0.517	0.622
	HyConvE [29]	0.764	0.712	0.798	0.858	0.309	0.208	0.322	0.526	0.457	0.369	0.500	0.619
	HySAE-I (Ours)	**0.809**	**0.756**	**0.842**	**0.903**	0.373	0.257	0.430	0.605	0.507	0.426	**0.534**	**0.688**
	HySAE-E (Ours)	0.802	0.749	0.837	0.901	**0.386**	**0.265**	**0.449**	**0.626**	**0.509**	**0.449**	0.528	0.625
5-ary	RAM [20]	0.787	0.721	0.862	0.900	0.113	0.043	0.145	0.221	0.387	0.290	0.531	0.568
	PosKHG [4]	0.792	0.687	0.860	0.909	0.155	0.100	0.217	0.233	0.463	0.409	0.561	0.574
	HyConvE [29]	0.786	0.701	0.861	0.896	0.391	0.301	0.429	0.562	0.953	0.937	0.967	0.976
	HySAE-I (Ours)	0.866	0.793	**0.931**	**0.982**	0.435	0.338	0.490	0.620	**0.968**	**0.962**	0.971	0.977
	HySAE-E (Ours)	**0.881**	**0.823**	0.928	0.978	**0.472**	**0.379**	**0.527**	**0.635**	0.967	0.960	**0.972**	**0.978**
6-ary	RAM [20]	0.777	0.698	0.881	0.917	0.070	0.042	0.075	0.130	-	-	-	-
	PosKHG [4]	0.866	0.854	0.878	0.896	0.023	0.015	0.034	0.048	-	-	-	-
	HyConvE [29]	0.873	0.855	0.886	0.907	0.275	0.187	0.281	0.470	-	-	-	-
	HySAE-I (Ours)	**0.956**	**0.938**	**0.969**	**0.979**	0.278	0.191	0.294	0.485	-	-	-	-
	HySAE-E (Ours)	0.908	0.875	0.927	0.958	**0.313**	**0.200**	**0.345**	**0.558**	-	-	-	-



(a) Embedding Dimension (b) Convolution Kernel Size (c) Dilated Size (d) Batch Size

Figure 5: Effects of (a) embedding dimension, (b) convolution kernel size, (c) dilated size, and (d) batch size on knowledge hypergraph datasets.

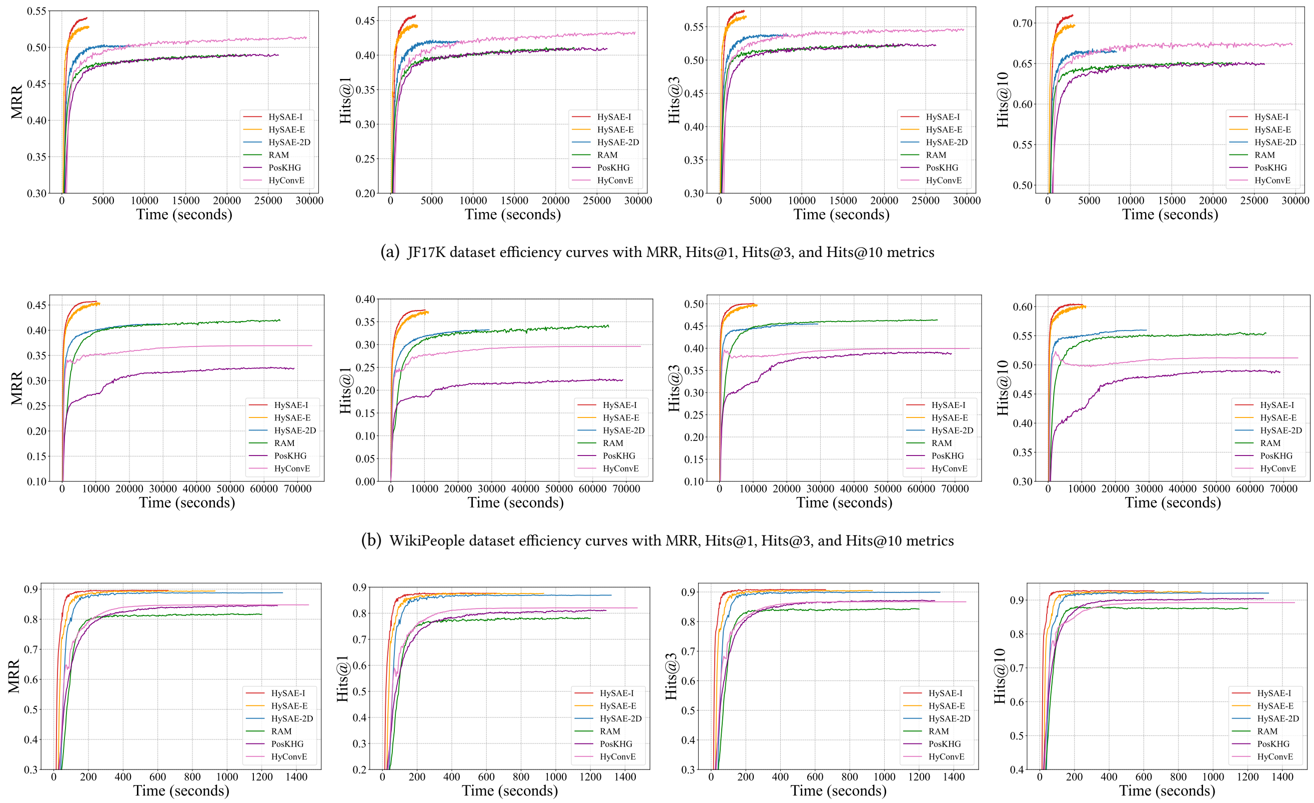
then evaluates the breakdown performance on knowledge tuples of different arities separately, so it can only be performed on the *mixed arity* knowledge hypergraph datasets [20, 29]. For a more comprehensive comparison of the performance breakdown task of the knowledge hypergraph models, we give the experimental results for all metrics in Table 6. HySAE consistently outperforms all baseline methods and has good model scalability.

Combined with the experimental results of *mixed arity* and *fixed arity* knowledge hypergraph link prediction tasks, the multi-scale feature learning and integration of HySAE-E is beneficial in enhancing the performance of knowledge hypergraph tasks with less n-ary semantic information. However, a higher number of scales in the ESE learning strategy will result in a more complex model and a higher number of parameters, impairing the model efficiency. Therefore, we set the ESE learning strategy as dual scales feature learning and integration, i.e., $l = 1$ and l>1. Also, we can reasonably choose the number of scales for feature learning and integration according to the needs of knowledge hypergraph tasks with different data and semantic information sizes in the real world.

C.2 Sensitivity of Hyperparameters

To further investigate the influence of critical hyperparameters on *mixed arity* and *fixed arity* knowledge hypergraphs, we conduct sensitivity analysis experiments on FB-AUTO and JF17K-4 datasets, including embedding dimension, convolution kernel size, dilated size, and batch size. Except for the hyperparameter of sensitivity analysis, the other hyperparameters are consistent with the knowledge hypergraph link prediction task.

Figure 5(a) shows the effect of embedding dimension on model performance, and it can be seen that when the embedding dimension is small, the model performance is generally lower. When the



(a) JF17K dataset efficiency curves with MRR, Hits@1, Hits@3, and Hits@10 metrics

(b) WikiPeople dataset efficiency curves with MRR, Hits@1, Hits@3, and Hits@10 metrics

(c) FB-AUTO dataset efficiency curves with MRR, Hits@1, Hits@3, and Hits@10 metrics

Figure 6: The model training efficiency curves on knowledge hypergraph datasets.

embedding dimension increases to a threshold, the model performance will not significantly improve, and even performance decay will occur. Figure 5(b) and Figure 5(c) show that the convolution kernel size and dilated size have an impact on the model performance, which proves that the size of the semantic-aware region of the model can affect model performance. Figure 5(d) illustrates the effect of batch size on model performance, which is a very important hyperparameter during model training. The inherent n-ary semantic information of *fixed arity* knowledge hypergraph is relatively simple, so the performance of HySAE on *fixed arity* knowledge hypergraph dataset is more robust.

C.3 Model Efficiency Curves Supplement

For a more comprehensive comparison of the model training efficiency curves, the experimental results for all metrics are given in Figure 6. It is evident from the experimental results that our proposed models can achieve higher performance in a shorter time on all metrics. Additionally, our proposed models are significantly faster than other baseline models and can complete 300 epochs iteration in the shortest time. In addition, the training efficiency curve of HySAE-I using the ISE learning strategy is better than that of HySAE-E using the ESE learning strategy. This further validates our idea that multi-scale feature learning and integration degrades the model efficiency.