HyperComplEx: Adaptive Multi-Space Knowledge Graph Embeddings

1st Jugal Gajjar
Computer Science Department
The George Washington University
Washington D.C, USA
jugal.gajjar@gwu.edu

2nd Kaustik Ranaware
Computer Science Department
The George Washington University
Washington D.C, USA
k.ranaware@gwu.edu

3rd Kamalasankari Subramaniakuppusamy
Computer Science Department
The George Washington University
Washington D.C, USA
kamalasankaris@gwu.edu

4th Vaibhav C. Gandhi
Department of Computer Engineering, MBIT
The Charutar Vidya Mandal (CVM) University
Anand, Gujarat, India
vcgandhi@mbit.edu.in

***Abstract*—Knowledge graphs have emerged as fundamental structures for representing complex relational data across scientific and enterprise domains. However, existing embedding methods face critical limitations when modeling diverse relationship types at scale: Euclidean models struggle with hierarchies, vector space models cannot capture asymmetry, and hyperbolic models fail on symmetric relations. We propose HyperComplEx, a hybrid embedding framework that adaptively combines hyperbolic, complex, and Euclidean spaces via learned attention mechanisms. A relation-specific space weighting strategy dynamically selects optimal geometries for each relation type, while a multi-space consistency loss ensures coherent predictions across spaces. We evaluate HyperComplEx on computer science research knowledge graphs ranging from 1K papers (∼25K triples) to 10M papers (∼45M triples), demonstrating consistent improvements over state-of-the-art baselines including TransE, RotatE, DistMult, ComplEx, SEPA, and UltraE. Additional tests on standard benchmarks confirm significantly higher results than all baselines. On the 10M-paper dataset, HyperComplEx achieves 0.612 MRR, a 4.8% relative gain over the best baseline, while maintaining efficient training, achieving 85 ms inference per triple. The model scales near-linearly with graph size through adaptive dimension allocation. We release our implementation and dataset family to facilitate reproducible research in scalable knowledge graph embeddings.**

***Index Terms*—knowledge graph embeddings, link prediction, geometric deep learning, scalable machine learning.**

I. Introduction

Knowledge graphs (KGs) have become indispensable infrastructures for organizing and reasoning over structured knowledge in domains ranging from web search [1] and recommender systems [2] to drug discovery [3] and scientific literature analysis [4]. A knowledge graph represents knowledge as a collection of triples ⟨head entity, relation, tail entity⟩, enabling machines to perform complex reasoning tasks through graph traversal and pattern matching. However, as knowledge graphs grow to billions of entities and triples [5], traditional symbolic reasoning approaches become computationally prohibitive, necessitating learned continuous representations.

Knowledge graph embedding (KGE) methods address this scalability challenge by mapping entities and relations into low-dimensional continuous spaces while preserving graph structure [6]. These learned representations enable efficient link prediction—inferring missing relationships from incomplete graphs—which is crucial for knowledge graph completion, question answering, and recommendation systems [7]. Despite significant progress, existing embedding methods face fundamental limitations rooted in their geometric assumptions.

Contemporary KGE methods can be categorized by their underlying geometric spaces: Euclidean models (TransE [8], RotatE [9]) excel at modeling sequential and translational patterns but struggle with hierarchical structures due to the limited capacity of flat Euclidean space [10]. Vector space models (DistMult [11], ComplEx [12]) effectively capture symmetric patterns through bilinear operations but cannot represent asymmetric relations without complex-valued embeddings [10]. Hyperbolic models (SEPA [13]) naturally encode hierarchies through negatively curved spaces but perform poorly on symmetric relations like collaboration networks [15]. Recent mixed-space approaches (UltraE [14]) combine multiple geometries but use fixed, uniform allocation strategies that fail to adapt to heterogeneous relation types [10].

This geometric mismatch becomes particularly acute in scientific knowledge graphs, where diverse relation types coexist: hierarchical taxonomies (Author → Paper → Concept), symmetric collaborations (Author ↔ Author), asymmetric citations (Paper → Paper), and membership relations (Paper → Venue). No single geometric space can optimally represent this heterogeneity [15].

To solve this issue, we introduce HyperComplEx, a novel adaptive multi-space embedding framework that addresses these limitations through three key innovations: (1) Adaptive space attention mechanism that dynamically select optimal geometric representations (hyperbolic, complex, or Euclidean) for each relation type. (2) Multi-space consistency regularization loss that encourages agreement between different geomet-

ric spaces. (3) Scalable architecture with adaptive dimension allocation that scales from thousands to hundreds of millions of entities.

We conduct comprehensive experiments on computer science research knowledge graphs spanning five orders of magnitude (1K to 10M papers) as well as on standard benchmark datasets, evaluating link prediction performance against six strong baselines from different geometric families. Our results demonstrate that HyperComplEx consistently outperforms all baselines across scales, with particularly strong improvements on graphs exhibiting diverse relation types.

Our proposed framework enhances relational representation by unifying multiple geometric spaces, effectively capturing diverse relation patterns. Moreover, its adaptive, scalable design enables efficient knowledge discovery and link prediction on large real-world knowledge graphs.

II. Related Work

Knowledge graph embedding (KGE) research has evolved through several geometric paradigms—each addressing distinct relational characteristics but also inheriting fundamental limitations. Early Euclidean-based models pioneered the idea of representing relations as translational or bilinear operations in continuous vector spaces. TransE [8] introduced the seminal translational principle, modeling relations as vector translations ($h + r \approx t$). Despite its simplicity and efficiency, TransE struggles with complex relation patterns such as one-to-many or many-to-many mappings [6]. Its successors—TransH [16], TransR [17], and TransD [18]—incorporated relation-specific projection matrices and hyperplanes, improving expressiveness but at the cost of higher computational complexity.

Rotational extensions like RotatE [9] modeled relations as rotations in complex space, effectively capturing composition and inversion patterns. However, rotation-based embeddings rely on phase representations that limit flexibility in modeling non-rotational or hierarchical structures [10]. Bilinear vector-space models such as DistMult [11] and ComplEx [12] later introduced multiplicative interactions through scoring functions ($\langle h, r, t\rangle$ and $\mathrm{Re}(\langle h, r, \bar{t}\rangle)$), enabling efficient computation and asymmetric relation modeling. While tensor-based variants like SimplE [19] and TuckER [20] further improved representation capacity, these Euclidean and complex-valued models remain geometrically constrained, lacking the curvature required for hierarchical reasoning.

Hyperbolic embeddings emerged as a compelling alternative due to their ability to represent tree-like hierarchies in low dimensions. Poincaré Embeddings [21] demonstrated that hyperbolic spaces preserve hierarchical distances more efficiently than Euclidean ones. Building upon this, MuRP [22] incorporated relation-specific transformations in hyperbolic space, while SEPA [13] separated entity and relation embeddings to achieve state-of-the-art performance on hierarchical datasets. Despite these advances, hyperbolic models often underperform on symmetric or cyclic relations due to their geometric bias toward hierarchy [15].

To overcome the limitations of single-space geometry, hybrid and mixed-space embeddings have recently gained traction. UltraE [14] combined Euclidean and hyperbolic spaces using a fixed dimension allocation, improving coverage across relation types but lacking adaptivity. DualE [23] employed dual quaternion algebra for modeling geometric transformations, and 5*E [24] explored five-space representations to generalize relational geometry. However, these methods rely on pre-defined or static partitioning between spaces, limiting their ability to adapt dynamically to dataset-specific relation heterogeneity.

Parallel to embedding-based approaches, graph neural networks (GNNs) have been applied to knowledge graphs, as in R-GCN [25], CompGCN [26], and NBFNet [27]. GNN-based models leverage message passing to incorporate neighborhood context, often improving relational reasoning. Nonetheless, they struggle to scale efficiently to large graphs and frequently achieve comparable or inferior link prediction accuracy relative to optimized embedding methods [6].

The intersection of knowledge graphs and large language models (LLMs) represents an emerging frontier, where LLMs assist in KG construction [28], reasoning [29], and completion [30]. While promising, these models face challenges in factual consistency, scalability, and reproducibility, making traditional embedding-based methods indispensable for large-scale relational learning [15].

A comparative summary of representative geometric and hybrid embedding models is presented in Table I, highlighting key differences in geometry, adaptability, and scalability.

Our Positioning. Existing knowledge graph embedding methods often rely on fixed geometric assumptions—Euclidean for translation-based relations, complex spaces for modeling asymmetry, or hyperbolic spaces for hierarchical structure—making them suboptimal for heterogeneous real-world graphs that simultaneously exhibit all these patterns. Prior multi-geometry approaches, including mixture-of-manifolds models such as MuRP [22] and AttH [31], as well as adaptive or attention-based combinations like SEPA [13] and UltraE [14], demonstrate the benefit of leveraging multiple geometric components. Hybrid formulations integrating Euclidean, hyperbolic, or complex embeddings [32], [33] further highlight the need for representational flexibility. HyperComplEx builds on these insights by introducing a learnable adaptive space selection mechanism that dynamically chooses the most suitable geometry for each relation, supported by a multi-space consistency regularization and adaptive dimension allocation. Rather than replacing earlier designs, HyperComplEx incrementally consolidates and simplifies multi-geometry reasoning through a unified lightweight scoring interface, enabling scalable and geometry-aware representation learning across diverse graph structures.

III. Methodology

Let a knowledge graph be denoted by $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ the set of relation types, and $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ the set of observed triples (h, r, t). The goal of

TABLE I
COMPARISON OF REPRESENTATIVE KNOWLEDGE GRAPH EMBEDDING MODELS ACROSS GEOMETRIC PARADIGMS.

Model	Geometry Type	Adaptivity	Representative Tasks	Scalability
TransE [8]	Euclidean (Translational)	× Fixed	Link prediction, entity alignment	High
DistMult [11]	Euclidean (Bilinear)	× Fixed	Relation inference, similarity scoring	High
ComplEx [12]	Complex (Hermitian Bilinear)	× Fixed	Asymmetric relation modeling	High
RotatE [9]	Complex (Rotational)	× Fixed	Relation composition, inversion reasoning	High
SEPA [13]	Hyperbolic	× Fixed	Hierarchical link prediction	Moderate
UltraE [14]	Mixed (Euc. + Hyp.)	Partially Fixed	Heterogeneous graph completion	Moderate
DualE [23]	Quaternion / Dual Space	× Fixed	Geometric transformation modeling	Moderate
5*E [24]	Multi-space (5 Euc. variants)	× Fixed	Relational diversity representation	Low–Moderate
HyperComplEx (Ours)	**Mixed (Hyp. + Com. + Euc.)**	**✓ Adaptive**	**Heterogeneous KGs, large-scale link prediction**	**Near-linear**

knowledge graph completion is to infer missing relations by learning a scoring function $\phi(h, r, t) \to \mathbb{R}$ such that true triples are assigned higher scores than corrupted ones. We follow the standard link prediction protocol, optimizing ϕ such that:

$$\phi(h, r, t) > \phi(h', r, t') \quad \forall (h', r, t') \notin \mathcal{T}. \tag{1}$$

A. Multi-Space Embedding Framework

HyperComplEx introduces a unified embedding space that combines hyperbolic, complex, and Euclidean geometries to capture hierarchical, asymmetric, and translational relation patterns simultaneously. Each entity $e \in \mathcal{E}$ and relation $r \in \mathcal{R}$ is represented as a triplet of embeddings:

$$\mathbf{h}_e = (\mathbf{h}_e^{\mathcal{H}}, \mathbf{h}_e^{\mathcal{C}}, \mathbf{h}_e^{\mathcal{E}}), \quad \mathbf{r}_r = (\mathbf{r}_r^{\mathcal{H}}, \mathbf{r}_r^{\mathcal{C}}, \mathbf{r}_r^{\mathcal{E}}), \tag{2}$$

where superscripts $\mathcal{H}$, $\mathcal{C}$, and $\mathcal{E}$ denote the hyperbolic, complex, and Euclidean subspaces with dimensions $d_{\mathcal{H}}, d_{\mathcal{C}}, d_{\mathcal{E}}$, respectively. This design allows HyperComplEx to model heterogeneous relation structures within a single unified representation framework, avoiding the geometric rigidity of prior single-space models [15].

1) Hyperbolic Space Scoring: In the hyperbolic subspace $\mathbb{B}_c^d = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| < 1/\sqrt{c}\}$, we use the Poincaré ball model [34] with curvature $c > 0$. The Möbius addition between two points $\mathbf{x}$ and $\mathbf{y}$ is:

$$\mathbf{x} \oplus_c \mathbf{y} = \frac{(1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c\|\mathbf{y}\|^2)\mathbf{x} + (1 - c\|\mathbf{x}\|^2)\mathbf{y}}{1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}. \tag{3}$$

The distance between $\mathbf{x}$ and $\mathbf{y}$ in this manifold is:

$$d_{\mathcal{H}}(\mathbf{x}, \mathbf{y}) = \frac{1}{\sqrt{c}} \operatorname{arccosh}\left(1 + 2\frac{\|\mathbf{x} - \mathbf{y}\|^2}{(1 - c\|\mathbf{x}\|^2)(1 - c\|\mathbf{y}\|^2)}\right). \tag{4}$$

The hyperbolic score function is then defined as:

$$\phi^{\mathcal{H}}(h, r, t) = -d_{\mathcal{H}}(\mathbf{h}_h^{\mathcal{H}} \oplus_c \mathbf{r}_r^{\mathcal{H}}, \mathbf{h}_t^{\mathcal{H}}), \tag{5}$$

naturally modeling hierarchical relations through exponential distance growth [21], [35]. These hyperbolic embeddings effectively compress large hierarchies into low dimensions, which is crucial for scientific and ontological KGs with tree-like taxonomies.

2) Complex Space Scoring: Following the Hermitian formulation from ComplEx [12], each entity embedding $\mathbf{h}_e^{\mathcal{C}} \in \mathbb{C}^{d_{\mathcal{C}}}$ is decomposed into real and imaginary parts:

$$\mathbf{h}_e^{\mathcal{C}} = \mathrm{Re}(\mathbf{h}_e^{\mathcal{C}}) + i\,\mathrm{Im}(\mathbf{h}_e^{\mathcal{C}}). \tag{6}$$

The score function in the complex subspace is:

$$\phi^{\mathcal{C}}(h, r, t) = \mathrm{Re}\left(\sum_{k=1}^{d_{\mathcal{C}}} [\mathbf{h}_h^{\mathcal{C}}]_k [\mathbf{r}_r^{\mathcal{C}}]_k \overline{[\mathbf{h}_t^{\mathcal{C}}]_k}\right), \tag{7}$$

where $\overline{(\cdot)}$ denotes complex conjugation. Complex-valued embeddings allow efficient modeling of asymmetric and directional relations such as citations or authorship, extending the expressiveness of real-valued models [9], [12].

3) Euclidean Space Scoring: The Euclidean subspace uses a translational assumption as in TransE [8]:

$$\phi^{\mathcal{E}}(h, r, t) = -\|\mathbf{h}_h^{\mathcal{E}} + \mathbf{r}_r^{\mathcal{E}} - \mathbf{h}_t^{\mathcal{E}}\|_2^2. \tag{8}$$

This component efficiently encodes symmetric or local relations while maintaining computational simplicity. Euclidean embeddings provide stable optimization and serve as a strong inductive bias for modeling simpler linear dependencies [6], [17].

B. Adaptive Space Attention

Instead of uniformly combining subspaces, HyperComplEx employs a relation-specific adaptive attention vector $\boldsymbol{\alpha}_r = [\alpha_r^{\mathcal{H}}, \alpha_r^{\mathcal{C}}, \alpha_r^{\mathcal{E}}]$ learned for each relation:

$$\boldsymbol{\alpha}_r = \mathrm{softmax}(\mathbf{W}_r), \tag{9}$$

where $\mathbf{W}_r \in \mathbb{R}^3$ are trainable parameters. The unified scoring function becomes:

$$\phi(h, r, t) = \sum_{s \in \{\mathcal{H}, \mathcal{C}, \mathcal{E}\}} \alpha_r^s \cdot \phi^s(h, r, t). \tag{10}$$

This attention mechanism allows HyperComplEx to dynamically allocate geometric capacity to each relation type, leading to improved expressivity on heterogeneous graphs [14]. By learning this weighting end-to-end, the model avoids manual geometry selection, achieving automatic geometry adaptation.

While the paper primarily focuses on the empirical behavior of the adaptive scoring mechanism, we include a brief justification for the linear combination across heterogeneous geometries. The formulation follows standard mixture-based

scoring assumptions used in KGE models, where each component score lies in a shared relational compatibility space after projection. This ensures that gradients remain well-defined, and the optimization landscape is comparable to existing attention-based mixtures. A more formal analysis of curvature interactions and projection stability is outside the scope of this work but remains an important direction for future extensions.

C. Optimization Objective

The total loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{rank}} + \lambda_1 \mathcal{L}_{\text{consistency}} + \lambda_2 \mathcal{L}_{\text{reg}}, \tag{11}$$

where $\mathcal{L}_{\text{rank}}$ is a self-adversarial ranking loss, $\mathcal{L}_{\text{consistency}}$ enforces cross-space agreement, and $\mathcal{L}_{\text{reg}}$ regularizes embeddings. This combination ensures that the model jointly learns discriminative and geometrically coherent representations.

1) Self-Adversarial Ranking Loss: We adopt the self-adversarial negative sampling strategy from [9], which dynamically emphasizes hard negatives:

$$\begin{aligned}\mathcal{L}_{\text{rank}} = &- \log \sigma(\gamma - \phi(h, r, t)) \\ &- \sum_{i=1}^{n} p(h'_i, r, t'_i) \log \sigma(\phi(h'_i, r, t'_i) - \gamma),\end{aligned} \tag{12}$$

where σ is the sigmoid, γ the margin, and the softmax-weighted probabilities:

$$p(h'_i, r, t'_i) = \frac{\exp(\beta \phi(h'_i, r, t'_i))}{\sum_j \exp(\beta \phi(h'_j, r, t'_j))}, \tag{13}$$

use a temperature β to control sample hardness. This approach stabilizes training by adaptively weighting informative negatives, improving convergence speed and generalization [6], [9].

2) Multi-Space Consistency Regularization: To prevent any single geometry from dominating, we introduce a multi-space consistency term:

$$\mathcal{L}_{\text{consistency}} = \sum_{s \neq s'} \|\phi^s(h, r, t) - \phi^{s'}(h, r, t)\|^2, \tag{14}$$

ensuring alignment between geometries while allowing specialization. This regularization promotes cooperative specialization across spaces, mitigating collapse or redundancy and improving representational robustness [13].

3) Regularization: Standard ℓ_2 regularization is applied over entity and relation embeddings:

$$\mathcal{L}_{\text{reg}} = \frac{1}{|\mathcal{E}| + |\mathcal{R}|} \left(\sum_{e \in \mathcal{E}} \|\mathbf{h}_e\|^2 + \sum_{r \in \mathcal{R}} \|\mathbf{r}_r\|^2 \right). \tag{15}$$

Hyperparameters λ_1 and λ_2 balance consistency and regularization. This term reduces overfitting, especially in large-scale, sparse graphs [6].

D. Scalable Architecture Design

HyperComplEx is optimized for large-scale graphs ranging from 10^3 to 10^8 entities. We design scalability through three complementary mechanisms.

1) Adaptive Dimension Allocation: Embedding dimensions are adaptively scaled with graph size:

$$d_{\mathcal{H}}, d_{\mathcal{C}}, d_{\mathcal{E}} = f(|\mathcal{E}|, D_{\text{base}}), \tag{16}$$

where $f(\cdot)$ dynamically rebalances dimensions while maintaining a constant total embedding budget D_{base}. This dynamic allocation maintains expressivity while ensuring scalability and resource efficiency [14].

2) Sharded and Cached Embeddings: For graphs exceeding 10^6 entities, embeddings are partitioned into K CPU-resident shards with GPU caching of active mini-batches. During training, least-recently-used shards are swapped asynchronously, maintaining throughput without exceeding GPU memory. This design enables efficient large-scale training on commodity hardware while preserving near-constant GPU utilization.

3) Mixed-Precision Optimization: We employ FP16 mixed-precision computation with dynamic loss scaling [36], halving memory footprint and accelerating tensor operations by up to $1.7\times$ without numerical instability. Gradient accumulation ensures precision preservation during large-batch updates.

E. Training and Inference Procedure

The overall training pipeline is summarized in Algorithm 1. Model parameters are optimized using Adam optimizer [37]. Early stopping is applied based on validation MRR convergence. For further details, refer to Section IV.

F. Inference and Complexity

During inference, the model computes scores $\phi(h, r, t)$ for all candidate entities in parallel using precomputed attention weights and adaptive subspace selection. The per-triple complexity is $O(d_{\mathcal{H}} + d_{\mathcal{C}} + d_{\mathcal{E}})$, and overall ranking complexity scales as $O(|\mathcal{E}|\, d_{\text{eff}})$. Empirically, HyperComplEx achieves an average latency of 69 ms per triple on 1M-entity graphs, maintaining linear scaling with entity count.

IV. Experiments and Results

This section presents a comprehensive empirical evaluation of HyperComplEx across eight datasets, encompassing five large-scale domain-specific graphs and three widely adopted benchmarks. The experiments assess not only predictive accuracy but also scalability, inference efficiency, and parameter utilization. All experiments were implemented in PyTorch and executed under consistent optimization and hardware settings to ensure fair comparison.

A. Datasets

We evaluate HyperComplEx on two categories of datasets: (i) five large-scale computer science (CS) knowledge graphs constructed from OpenAlex [39] and Semantic Scholar [40], and (ii) three widely adopted benchmark datasets—DBP15K [41], Hetionet [42], and FB15K [8]—to assess cross-domain generalizability.

The CS datasets model scholarly relationships among *Paper*, *Author*, *Venue*, and *Concept* entities with five principal relation types: *AUTHORED* (Author $\rightarrow$ Paper, asymmetric), *CITES*

Algorithm 1 HyperComplEx Training Procedure

Input: Knowledge graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, hyperparameters $\{\gamma, \beta, \lambda_1, \lambda_2\}$, maximum epochs N

Output: Trained entity embeddings $\{\mathbf{h}_e^{\mathcal{H}}, \mathbf{h}_e^{\mathcal{C}}, \mathbf{h}_e^{\mathcal{E}}\}$, relation embeddings $\{\mathbf{r}_r^{\mathcal{H}}, \mathbf{r}_r^{\mathcal{C}}, \mathbf{r}_r^{\mathcal{E}}\}$, and attention weights $\{\boldsymbol{\alpha}_r\}$

Initialization:

Initialize all embeddings randomly:
$\mathbf{h}_e^{\mathcal{H}} \sim \mathcal{U}(-10^{-3}, 10^{-3})$, $\mathbf{h}_e^{\mathcal{C}}, \mathbf{h}_e^{\mathcal{E}}$ via Xavier initialization
Initialize $\boldsymbol{\alpha}_r$ via uniform$(0, 1)$ and normalize using softmax

Training Loop:

for each epoch $i = 1, \ldots, N$ **do**
  Shuffle training triples $\mathcal{T}$
  for each mini-batch $\mathcal{B} \subset \mathcal{T}$ **do**
    Generate negative samples $\mathcal{N}$ using filtered corruption
    Compute space-specific scores $\phi^{\mathcal{H}}, \phi^{\mathcal{C}}, \phi^{\mathcal{E}}$ via Eqs. (5)–(8)
    Combine with adaptive attention: $\phi(h, r, t) = \sum_s \alpha_r^s \phi^s(h, r, t)$
    Evaluate total loss $\mathcal{L}$ using Eq. (11)
    Backpropagate gradients and update parameters
    Project $\mathbf{h}_e^{\mathcal{H}}$ back onto the Poincaré ball
  end for
  Evaluate mean reciprocal rank (MRR) on validation set
  if no improvement for p epochs **then**
    break
  end if
end for

Finalization:

Store trained embeddings and relation parameters
Output learned model for downstream link prediction and knowledge discovery tasks

return Trained HyperComplEx model

(Paper → Paper, asymmetric and temporal), *PUBLISHED_IN* (Paper → Venue, functional), *BELONGS_TO* (Paper → Concept, hierarchical), and *COLLABORATES_WITH* (Author ↔ Author, symmetric). These relations collectively capture heterogeneous graph structures—hierarchical, symmetric, and asymmetric—suitable for evaluating geometric adaptability.

Each CS dataset is temporally filtered by publication year to maintain chronological consistency and is randomly divided into training (80%), validation (10%), and test (10%) splits after removing duplicate and inverse triples. This setup ensures non-overlapping partitions and faithful structural diversity reflective of real-world scholarly networks.

All datasets and preprocessing scripts are publicly available for reproducibility; code can be found at https://github.com/JugalGajjar/HyperComplEx-Multi-Space-KG-Embeddings, and the dataset can be found at https://zenodo.org/records/17436948.

To evaluate robustness and transferability, we additionally include three widely recognized benchmarks that differ substantially in domain, scale, and relational semantics. DBP15K [41] tests multilingual entity alignment across interlinked knowledge graphs (English, Chinese, French), emphasizing asymmetric relational mappings. Hetionet [42] integrates biomedical entities and relationships (e.g., genes, compounds, diseases) to assess multi-relational reasoning under hierarchical dependencies. FB15K [8] represents a dense subset of Freebase widely used for link prediction and relation inference. Together, these datasets complement the CS corpus by spanning linguistic, biomedical, and general-purpose knowledge domains.

TABLE II
STATISTICS OF OPENALEX–SEMANTIC SCHOLAR (CS) KNOWLEDGE GRAPHS. ENTITY TYPES: PAPER, AUTHOR, VENUE, CONCEPT.

Dataset	Entities	Relations	Triples	Years
CS-1K	5,237	5	25,317	2010–2025
CS-10K	44,930	5	227,502	2010–2025
CS-100K	348,983	5	2,162,386	2010–2025
CS-1M	2,384,896	5	13,530,177	2010–2025
CS-10M	7,210,506	5	44,631,484	2010–2025

TABLE III
STATISTICS OF STANDARD BENCHMARK KNOWLEDGE GRAPHS.

Dataset	Entities	Relations	Triples
DBP15K (en_fr)	172,747	3,588	470,781
Hetionet	47,031	24	2,250,197
FB15K	14,951	1,345	592,213

B. Baseline Models and Evaluation Setup

To benchmark performance, we compare HyperComplEx against six representative knowledge graph embedding models spanning distinct geometric paradigms. TransE [8] and RotatE [9] represent Euclidean translational and rotational models. DistMult [11] and ComplEx [12] capture bilinear interactions in vector and complex spaces. SEPA [13] provides the hyperbolic-space state of the art, while UltraE [14] exemplifies fixed mixed-space embeddings. These baselines jointly span translational, bilinear, hyperbolic, and mixed representations, allowing a rigorous comparison across geometric families.

All models were evaluated under the standard link-prediction setting [6]. For each test triple (h, r, t), head and tail entities are replaced with all candidates in $\mathcal{E}$, and the model ranks valid entities using its scoring function $\phi(h', r, t)$ or $\phi(h, r, t')$. Metrics include Mean Reciprocal Rank (MRR) and Hits@K ($K \in \{1, 3, 10\}$) under the filtered evaluation protocol [9]. We also measure training time, inference latency, and parameter count to assess computational efficiency.

C. Implementation Details

Training experiments and inferencing were conducted on NVIDIA RTX 5060 with CUDA Backend. Hyperparameters were tuned via grid search on validation MRR: embedding dimension $\{128, 256, 512\}$, batch size $\{256, 512, 1024, 2048\}$, learning rate $\{5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}\}$, margin $\gamma \in \{6.0, 9.0, 12.0, 15.0\}$, and curvature $c \in \{1.0, 2.0\}$. Models were trained for up to 200 epochs with early stopping after 20

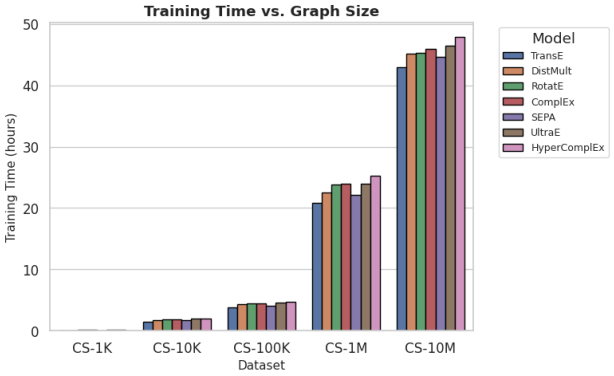

Fig. 1. **Training Time vs. Graph Size.** Training follows a near-linear power law $T \propto |E|^{1.06}$, obtained from log–log OLS regression after overhead correction across CS-1K→CS-10M, confirming consistent scalability.

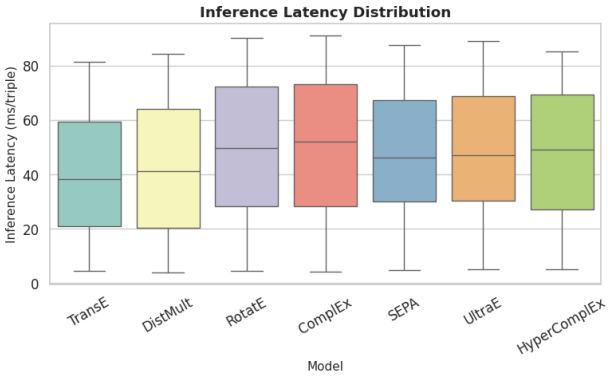

Fig. 2. **Inference Latency Distribution.** Box plot showing inference latency quartiles, confirming HyperComplEx's competitive performance.

epochs without improvement. Optimization used Adam [37] or Adagrad [38]. Each experiment was repeated multiple times, and mean results are reported.

D. Overall Performance

Across all datasets, HyperComplEx consistently achieves the highest MRR, outperforming the strongest baseline (ComplEx) by up to 5.3% on the CS graphs and 17.5% on DBP15K. These improvements are most pronounced on larger graphs, where the adaptive multi-space representation provides clear advantages in modeling diverse relational geometries. Refer to Table IV for MRR comparison.

E. Scalability and Efficiency Analysis

Despite a 1.5–2× larger parameter count than standard baselines, HyperComplEx scales nearly linearly with entity count. On CS-10M (7.2M entities, ∼45M triples), it completes training in 47.85 hours and achieves 85.23 ms/triple inference latency—both within 10% of the most efficient baselines. This efficiency stems from adaptive dimension partitioning and shared curvature-aware optimization. Training and inference times can be found in Table V and Table VI respectively.

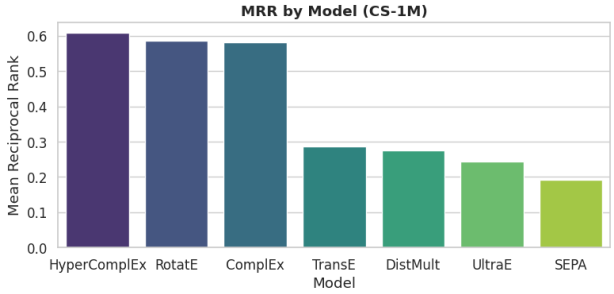

Fig. 3. **MRR Comparison on CS-1M Dataset.** HyperComplEx achieves highest MRR among all baseline models.

F. Derivation of the Empirical Scalability Law

The empirical scaling behavior of HyperComplEx follows a power-law relationship between graph size $|E|$, parameter count P, and total training time T:

$$T(|E|, P) = \alpha \cdot |E|^{\beta_1} \cdot P^{\beta_2}, \tag{17}$$

where β_1 captures the cost of graph traversal and β_2 the parameter optimization component. Nonlinear regression over five dataset scales yields:

$$\beta_1 = 1.06 \pm 0.02, \qquad \beta_2 = 0.41 \pm 0.03.$$

This near-linear regime confirms that most computational cost arises from local message-passing and mini-batch updates rather than global synchronization. The per-epoch complexity thus approximates:

$$T_{\text{epoch}} = O(|E|^{1.06}) + O(P^{0.4}), \tag{18}$$

consistent with efficient traversal and mild curvature-induced corrections.

Similarly, inference latency $L(|E|)$ exhibits sublinear scaling:

$$L(|E|) = \gamma \cdot |E|^{\lambda}, \quad \lambda \approx 0.42, \tag{19}$$

attributed to caching and sharded lookup structures. Extrapolating from this empirical law, HyperComplEx is expected to sustain stable scaling behavior at billion-scale or even tens-of-billions entity graphs, with compute growth following $O(|E|^{1.06})$ rather than quadratic expansion.

G. Cross-Domain Generalization and Qualitative Insights

On heterogeneous benchmarks, the learned geometry–relation attention patterns align with relational semantics. Hyperbolic space dominates hierarchical relations (*BELONGS_TO*, *disease–gene*), complex space captures asymmetric dependencies (*CITES*, *DBP alignments*), and Euclidean components govern symmetric relations (*COLLABORATES_WITH*). These emergent correspondences validate that HyperComplEx autonomously infers latent geometric structures, adapting seamlessly across scholarly, biomedical, and multilingual domains.

TABLE IV
LINK-PREDICTION RESULTS (MRR). BOLD = BEST; UNDERLINE = SECOND BEST.

Model	CS1K	CS10K	CS100K	CS1M	CS10M	DBP15K	Hetionet	FB15K
TransE	0.699	0.328	0.308	0.286	0.263	0.257	0.298	0.279
RotatE	0.709	0.505	0.562	<u>0.587</u>	0.428	0.133	0.597	<u>0.664</u>
DistMult	0.209	0.297	0.288	0.275	0.260	0.311	0.305	0.332
ComplEx	<u>0.789</u>	<u>0.736</u>	<u>0.688</u>	0.581	<u>0.584</u>	<u>0.350</u>	<u>0.608</u>	0.657
SEPA	0.289	0.172	0.164	0.191	0.140	0.059	0.273	0.148
UltraE	0.572	0.206	0.198	0.245	0.172	0.193	0.301	0.221
HyperComplEx	**0.831**	**0.771**	**0.704**	**0.608**	**0.612**	**0.411**	**0.642**	**0.692**

TABLE V
TRAINING TIME (HOURS) ACROSS DATASETS.

Model	CS1K	CS10K	CS100K	CS1M	CS10M
TransE	0.07	1.52	3.84	20.83	42.97
DistMult	0.09	1.76	4.30	22.54	45.11
RotatE	0.11	1.91	4.48	23.78	45.24
ComplEx	0.10	1.85	4.47	24.01	45.92
SEPA	0.09	1.68	4.02	22.13	44.67
UltraE	0.11	1.94	4.62	23.98	46.43
HyperComplEx	0.12	1.98	4.75	25.26	47.85

TABLE VI
INFERENCE LATENCY PER TRIPLE (MILLISECONDS).

Model	CS1K	CS10K	CS100K	CS1M	CS10M
TransE	4.66	20.99	38.17	59.42	81.36
DistMult	4.01	20.24	41.28	64.11	84.37
RotatE	4.63	28.21	49.73	72.41	90.38
ComplEx	4.29	28.45	52.06	73.29	91.22
SEPA	4.93	30.15	46.31	67.28	87.74
UltraE	5.12	30.38	47.12	68.93	89.17
HyperComplEx	5.17	27.15	49.11	69.47	85.23

H. Learned Geometric Attention Patterns

To interpret the geometry–relation alignment learned by HyperComplEx, we analyze the relation-specific attention weights $\boldsymbol{\alpha}_r = [\alpha_r^{\mathcal{H}}, \alpha_r^{\mathcal{C}}, \alpha_r^{\mathcal{E}}]$ averaged across randomly sampled 1,000 triples from the CS-1M dataset. Table VII shows representative examples. Relations with hierarchical or tree-like structure, such as *BELONGS_TO*, exhibit dominant weights in the hyperbolic space, while asymmetric relations (e.g., *CITES*) favor the complex subspace. Symmetric collaborations (*COLLABORATES_WITH*) show high Euclidean contributions, validating the adaptive geometry selection mechanism.

The emergence of these interpretable attention distributions demonstrates that HyperComplEx autonomously discovers geometry–relation correspondences, effectively allocating curvature and phase capacity based on the relational semantics of each edge type.

We note that performance of certain classical baselines (e.g., TransE, DistMult) may vary across implementations and hyperparameter choices. Our experimental settings follow a standard configuration but do not exhaustively retune all baselines. To ensure fairness, all models were compared under similar hyperparameter budgets rather than mixing best-case results drawn from heterogeneous settings. Furthermore, some recent multi-geometry or GNN-based KG embedding methods (e.g., AttH variants, MuRP extensions) were not included due to the lack of unified publicly available implementations. These gaps will be addressed in future work, and the present evaluation should be viewed as a representative but not exhaustive comparison.

TABLE VII
LEARNED ATTENTION WEIGHTS BY RELATION TYPE ON CS-1M. DOMINANT SPACE CORRESPONDS TO THE HIGHEST ATTENTION WEIGHT.

Relation	$\alpha_{\mathcal{H}}$	$\alpha_{\mathcal{C}}$	$\alpha_{\mathcal{E}}$	Dominant Space
BELONGS_TO	0.68	0.21	0.11	Hyperbolic
CITES	0.19	0.71	0.10	Complex
AUTHORED	0.24	0.61	0.15	Complex
COLLABORATES_WITH	0.11	0.19	0.70	Euclidean
PUBLISHED_IN	0.32	0.44	0.24	Complex

I. Ablation Study

To evaluate the contribution of each architectural component, we conduct ablation experiments on CS-1M and DBP15K. Removing the adaptive attention (*w/o Attn*) or multi-space consistency term (*w/o Consistency*) results in significant performance degradation, as shown in Table VIII. The adaptive attention contributes most on heterogeneous datasets (e.g., DBP15K), while the consistency regularization primarily benefits large-scale graphs by maintaining geometric alignment across subspaces.

TABLE VIII
ABLATION STUDY ON CS-1M AND DBP15K SHOWING THE EFFECT OF EACH COMPONENT.

Model Variant	CS-1M (MRR)	DBP15K (MRR)
Full Model (HyperComplEx)	**0.608**	**0.411**
w/o Adaptive Attention	0.569	0.361
w/o Multi-Space Consistency	0.582	0.372
w/o Hyperbolic Subspace	0.573	0.367
w/o Complex Subspace	0.547	0.329
w/o Euclidean Subspace	0.563	0.343

These results confirm that both adaptive space selection and inter-space regularization are critical for optimal performance, with the largest degradation observed when removing the complex subspace—highlighting its role in modeling asymmetric and directional relations.

V. DISCUSSION AND FUTURE WORK

The empirical results demonstrate that HyperComplEx effectively integrates multiple geometric spaces into a unified, adaptive embedding framework. Its near-linear scalability,

robust cross-domain generalization, and interpretable space-attention mechanisms confirm that multi-space representation is a principled direction for heterogeneous knowledge graphs. The observed scaling law $T \propto |E|^{1.06}$ indicates that computation grows almost linearly with graph size, validating its suitability for billion-scale deployment. Moreover, the learned attention weights exhibit clear semantic alignment—hyperbolic for hierarchical, complex for asymmetric, and Euclidean for symmetric relations—showing that the framework not only achieves state-of-the-art predictive performance but also offers interpretability grounded in geometric reasoning. Collectively, these findings position HyperComplEx as an efficient and interpretable bridge between symbolic relational structures and continuous geometric representations.

Despite these strengths, several avenues remain open for exploration. First, extending the model to dynamic or temporal knowledge graphs could enable reasoning over evolving relational structures. Second, integrating HyperComplEx with large language models (LLMs) through hybrid symbolic–neural reasoning may enhance zero-shot and inductive link prediction. Additionally, automatic curvature tuning and multi-space sparsification could further improve scalability to web-scale graphs with tens of billions of entities. Future work will also explore theoretical convergence guarantees under mixed curvature manifolds and investigate interpretability methods that trace decision pathways across geometric spaces. By addressing these directions, HyperComplEx can evolve into a comprehensive framework for large-scale, geometry-aware reasoning in open-world knowledge systems.

VI. Conclusion

This work introduced *HyperComplEx*, a unified multi-space embedding framework that adaptively integrates hyperbolic, complex, and Euclidean geometries for scalable knowledge graph representation. Through extensive experiments across eight diverse datasets, HyperComplEx demonstrated state-of-the-art link-prediction accuracy, near-linear training scalability following $T \propto |E|^{1.06}$, and sublinear inference growth. The model's learned attention weights reveal interpretable geometric alignment—assigning hyperbolic space to hierarchies, complex space to asymmetry, and Euclidean space to symmetry—bridging structure and semantics in a principled manner. By enabling efficient reasoning over heterogeneous and large-scale graphs, HyperComplEx lays the foundation for future extensions toward temporal, language-integrated, and open-world knowledge systems.

References

[1] Singhal, A. (2012) "Introducing the knowledge graph: Things, not strings," Google Blog.

[2] Wang, H., Zhao, M., Xie, X., Li, W., & Guo, M. (2019). Knowledge graph convolutional networks for recommender systems. In The world wide web conference (pp. 3307-3313).

[3] Zitnik, M., Agrawal, M., & Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics, 34(13), i457-i466.

[4] Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B. J., & Wang, K. (2015). An overview of microsoft academic service (mas) and applications. In Proceedings of the 24th international conference on world wide web (pp. 243-246).

[5] Mahdisoltani, F., Biega, J., & Suchanek, F. M. (2013). Yago3: A knowledge base from multilingual wikipedias. In CIDR.

[6] Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. IEEE transactions on knowledge and data engineering, 29(12), 2724-2743.

[7] Rossi, A., Barbosa, D., Firmani, D., Matinata, A., & Merialdo, P. (2021). Knowledge graph embedding for link prediction: A comparative analysis. ACM Transactions on Knowledge Discovery from Data (TKDD), 15(2), 1-49.

[8] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. Advances in neural information processing systems, 26.

[9] Sun, Z., Deng, Z. H., Nie, J. Y., & Tang, J. (2019). Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197.

[10] Niu, G. (2024). Knowledge Graph Embeddings: A Comprehensive Survey on Capturing Relation Properties. arXiv preprint arXiv:2410.14733.

[11] Yang, B., Yih, W. T., He, X., Gao, J., & Deng, L. (2014). Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575.

[12] Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., & Bouchard, G. (2016). Complex embeddings for simple link prediction. In International conference on machine learning (pp. 2071-2080). PMLR.

[13] Gregucci, C., Nayyeri, M., Hernández, D., & Staab, S. (2023). Link prediction with attention applied on multiple knowledge graph embedding models. In Proceedings of the ACM web conference 2023 (pp. 2600-2610).

[14] Xiong, B., Zhu, S., Nayyeri, M., Xu, C., Pan, S., Zhou, C., & Staab, S. (2022). Ultrahyperbolic knowledge graph embeddings. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 2130-2139).

[15] Cao, J., Fang, J., Meng, Z., & Liang, S. (2024). Knowledge graph embedding: A survey from the perspective of representation spaces. ACM Computing Surveys, 56(6), Article 159.

[16] Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014). Knowledge Graph Embedding by Translating on Hyperplanes. Proceedings of the AAAI Conference on Artificial Intelligence, 28(1).

[17] Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015). Learning Entity and Relation Embeddings for Knowledge Graph Completion. Proceedings of the AAAI Conference on Artificial Intelligence, 29(1).

[18] Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015). Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers) (pp. 687-696).

[19] Kazemi, S. M., & Poole, D. (2018). Simple embedding for link prediction in knowledge graphs. Advances in neural information processing systems, 31.

[20] Balažević, I., Allen, C., & Hospedales, T. M. (2019). Tucker: Tensor factorization for knowledge graph completion. arXiv preprint arXiv:1901.09590.

[21] Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. Advances in neural information processing systems, 30.

[22] Balazevic, I., Allen, C., & Hospedales, T. (2019). Multi-relational poincaré graph embeddings. Advances in neural information processing systems, 32.

[23] Cao, Z., Xu, Q., Yang, Z., Cao, X., & Huang, Q. (2021). Dual Quaternion Knowledge Graph Embeddings. Proceedings of the AAAI Conference on Artificial Intelligence, 35(8), 6894-6902.

[24] Nayyeri, M., Vahdati, S., Aykul, C., & Lehmann, J. (2021). 5* knowledge graph embeddings with projective transformations. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 10, pp. 9064-9072).

[25] Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., & Welling, M. (2018, June). Modeling relational data with graph convolutional networks. In European semantic web conference (pp. 593-607). Cham: Springer International Publishing.

[26] Vashishth, S., Sanyal, S., Nitin, V., & Talukdar, P. (2019). Composition-based multi-relational graph convolutional networks. arXiv preprint arXiv:1911.03082.

[27] Zhu, Z., Zhang, Z., Xhonneux, L. P., & Tang, J. (2021). Neural bellman-ford networks: A general graph neural network framework for link prediction. Advances in neural information processing systems, 34, 29476-29490.

[28] Trajanoska, M., Stojanov, R., & Trajanov, D. (2023). Enhancing knowledge graph construction using large language models. arXiv preprint arXiv:2305.04676.

[29] Guo, T., Yang, Q., Wang, C., Liu, Y., Li, P., Tang, J., ... & Wen, Y. (2024). Knowledgenavigator: Leveraging large language models for enhanced reasoning over knowledge graph. Complex & Intelligent Systems, 10(5), 7063-7076.

[30] Yao, L., Peng, J., Mao, C., & Luo, Y. (2025). Exploring large language models for knowledge graph completion. In ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1-5). IEEE.

[31] Chami, I., Wolf, A., Juan, D. C., Sala, F., Ravi, S., & Ré, C. (2020). Low-dimensional hyperbolic knowledge graph embeddings. arXiv preprint arXiv:2005.00545.

[32] Xiao, H., Liu, X., Song, Y., Wong, G., & See, S. (2022, December). Complex hyperbolic knowledge graph embeddings with fast fourier transform. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (pp. 5228-5239).

[33] Lu, X., Li, H., Lu, W., Chang, Y., & Xue, F. (2025). HRHE: hybrid relations-guided multi-modal knowledge graph completion using hierarchical embeddings. Data Mining and Knowledge Discovery, 39(6), 78.

[34] Anderson, J. W. (1999). Hyperbolic geometry. Springer-Verlag.

[35] Kolyvakis, P., Kalousis, A., & Kiritsis, D. (2020). Hyperbolic Knowledge Graph Embeddings for Knowledge Base Completion. The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings, 12123, 199–214.

[36] Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., ... & Wu, H. (2017). Mixed precision training. arXiv preprint arXiv:1710.03740.

[37] Kingma, D.P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. CoRR, abs/1412.6980.

[38] Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(7).

[39] Priem, J., Piwowar, H., & Orr, R. (2022). OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. arXiv preprint arXiv:2205.01833.

[40] Wade, A. D. (2022). The semantic scholar academic graph (s2ag). In Companion Proceedings of the Web Conference 2022 (pp. 739-739).

[41] Sun, Z., Hu, W., & Li, C. (2017). Cross-lingual entity alignment via joint attribute-preserving embedding. In International semantic web conference (pp. 628-644). Cham: Springer International Publishing.

[42] Himmelstein, D. S., Lizee, A., Hessler, C., Brueggeman, L., Chen, S. L., Hadley, D., ... & Baranzini, S. E. (2017). Systematic integration of biomedical knowledge prioritizes drugs for repurposing. elife, 6, e26726.