Knowledge Graph Embeddings using Neural Itô Process: From Multiple Walks to Stochastic Trajectories

Mojtaba Nayyeri[1], **Bo Xiong**[1], **Majid Mohammadi**[3]
Mst. Mahfuja Akter[6], **Mirza Mohtashim Alam**[6], **Jens Lehmann**[4,5], and **Steffen Staab**[1,2]
[1]University of Stuttgart, [2] University of Southampton, [3]Vrije Universiteit Amsterdam,
[4]TU Dresden, [5]Amazon (work done outside of Amazon), [6] University of Bonn

Abstract

Knowledge graphs mostly exhibit a mixture of branching relations, e.g., *hasFriend*, and complex structures, e.g., hierarchy and loop. Most knowledge graph embeddings have problems expressing them, because they model a specific relation r from a head h to tails by starting at the node embedding of h and transitioning deterministically to exactly one other point in the embedding space. We overcome this issue in our novel framework Itô$\hat{\text{E}}$ by modeling relations between nodes by relation-specific, stochastic transitions. Our framework is based on stochastic Itô processes, which operate on low-dimensional manifolds. Itô$\hat{\text{E}}$ is highly expressive and generic subsuming various state-of-the-art models operating on different, also non-Euclidean, manifolds. Experimental results show the superiority of Itô$\hat{\text{E}}$ over other deterministic embedding models with regard to the KG completion task.

1 Introduction

Knowledge graphs (KGs) play a central role in many AI-related tasks (Nickel et al., 2016) such as recommendation systems (Lukovnikov et al., 2017) and question answering (Zhang et al., 2016). KGs represent real-world knowledge as a set of facts in the form of triples *(entity, relation, entity)*, e.g., *(Alice, FriendOf, Bob)*. Entities are nodes of a graph and relations are the directed edges. KGs are highly incomplete which adversely affects the outcome of various KG-centered tasks. To tackle this problem, various link prediction approaches have been proposed to leverage the existing links for the prediction of new ones (a.k.a., knowledge graph completion). Among existing approaches (Wang et al., 2017; Ji et al., 2021), KG embedding (KGE) succeeded and became a popular line of work.

KGE models map entities and relations in KGs into a low dimensional geometric space and measure the plausibility of each triple *(entity, relation, entity)* by a scoring function f that uses the embedded vectors of this triple **(entity, relation, entity)** to gauge its plausibility by a real, positive value $f($**entity, relation, entity**$)$. Prominent examples of KGEs include translational/rotational families (Wang et al., 2014; Lin et al., 2015), bilinear families (Nickel et al., 2011; Yang et al., 2014) and neural network families (Dettmers et al., 2018; Nguyen et al., 2018). However, these models are all defined in Euclidean space that suffers from modeling complex structures such as hierarchies (Chami et al., 2020) in low-dimensional vector space. Therefore, KG embeddings based on non-Euclidean geometric spaces such as hyperbolic space (Chami et al., 2020; Balazevic et al., 2019) have been proposed. Both lines of work often formulate the relation-specific transitions between subsequent nodes in a deterministic way and represent them, e.g., by translations or rotations, implying that with regard to a particular relation each node is connected to a single other node.

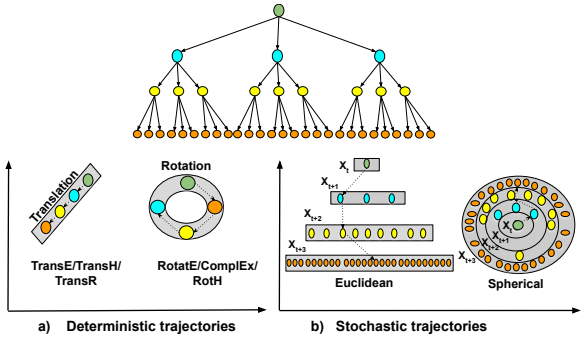


Figure 1: The embedding of a synthesized tree structure, shown in the top panel, by two approaches: a) deterministic transition; and b) stochastic transitions. Deterministic transition (translation or rotation) generates a specific path while stochastic transition generates multiple paths forming hierarchies.

In a broader view, this implies that starting from an embedded node with a given single relation type, there will be only one walk with the length m,

which is m subsequent relation-specific transitions from the starting node, and this walk will lead to another specific embedded node. However, starting at any specific node and considering a branching relation such as *hasFriend*, there may be several relation-specific walks of length m. These walks define a group of nodes in distance m. Figure 1 illustrates the limits of traditional embedding models based on translation or rotation with deterministic transitions (a) and contrasts it with the tree-like structure that unfolds in a non-deterministic model (b) able to represent a branching relationship.

In this paper, we formalize probabilistic walks of length $m-1$ between two groups of nodes $(\Omega_{t_1}, \Omega_{t_m})$ in order to represent branching relationships. Each group of nodes Ω_{t_i} from start Ω_{t_1} to end Ω_{t_m} is represented by a random variable X_{t_i}, which indicates a probabilistic distribution on the manifold $X_{t_i} : \Omega_{t_i} \rightarrow \mathcal{M}, \quad i = 1, \ldots, m$. The sequence of random variables $S = \{X_{t_1}, X_{t_2}, \ldots, X_{t_m}\}$ constitutes a stochastic process. A sample, also called realization, of the stochastic process S is a walk with $m-1$ transitions containing m entity representations $\{\boldsymbol{e}_{t_1}, \ldots, \boldsymbol{e}_{t_m}\}$ on the manifold, $\boldsymbol{e}_{t_i} \in \mathcal{M}$.

We model probabilistic transitions between groups of nodes using the Itô process, which defines transitions between nodes through an integral containing drift and diffusion. Drift models the deterministic nature of a transition, while diffusion captures stochastic variation. Drift and diffusion operations are learned separately via a relation-specific neural network.

Owing to the probabilistic perspective, we develop a KGE model, called ItôE, that uses the Itô process for stochastic transitions as well as various, also non-Euclidean, manifolds (i.e., sphere, Poincaré ball, hyperboloid) to support the modeling of heterogeneous graph structures on manifolds. We also provide an extensive theoretical investigation and prove that ItôE: a) is fully expressive; b) is capable of encoding various structures as stochastic processes; c) subsumes various state-of-the-art KGE models, including ComplEx and QuatE, d) models various relational patterns, e.g., symmetric patterns. Experimental results show the superiority of our model, especially in low-dimensional space.

2 Related Work

Euclidean KGE Models The earliest KGE models employ Euclidean geometry, i.e., a d dimensional real space $\mathbb{R}^d$, with its corresponding Euclidean distance function as well as inner product. TransE represents each relation as a translation from a head to a tail embedding. Several variants of TransE have been developed to model one-to-many relations, as well as symmetric and reflexive patterns (Wang et al., 2014; Lin et al., 2015). DistMult (Yang et al., 2014) proposes a tensor factorization with a diagonal relation matrix. The model captures symmetric relations, but not anti-symmetry. ComplEx(Trouillon et al., 2016) and QuatE (Zhang et al., 2019) models have been proposed as extensions of DistMult from the real space to the (hyper)complex space. RotatE (Sun et al., 2019) models each relation as rotation in the complex space using the Euler formula, which captures symmetric, anti-symmetric, inverse, and composition patterns. Apart from the mentioned shallow embedding models, there is a thread of neural network KGE models such as Neural Tensor Network (Socher et al., 2013), ConvE (Dettmers et al., 2018), and ConvKB (Nguyen et al., 2018). In summary, these models cannot capture various graph structures, especially in low dimensional space due to the underlying Euclidean space. A series of graph embedding models have been proposed based on random walks in the graph space.

Node2vec (Grover and Leskovec, 2016; Ristoski and Paulheim, 2016; Portisch and Paulheim, 2022; Huang et al., 2021; Perozzi et al., 2014) are among the models which perform a biased random walk in graph space and compute the low dimensional representation of nodes in such a way to maximize the likelihood of preserving network neighborhoods of nodes. The random walk-based models are not among state-of-the-art KGE models in the link prediction task. In addition, in most cases, the random walk is performed in the graph space to obtain the sequences of nodes for embedding, but the notion of walk in the embedding space is neglected as there is no transition function in the embedding space to model this.

Non-Euclidean KGE Models Euclidean-based KGE models are not capable of preserving complex graph structures in a low-dimensional space. However, embedding KGs on non-Euclidean manifolds has shown promising performance in the preservation of a few structures, especially in low-dimensional spaces. (Nickel and Kiela, 2017; Chami et al., 2020; Balazevic et al., 2019; Weber and Nickel, 2018) showed the advantage of

Poincaré Ball and other geometries for embedding graphs with various structures including hierarchical structures. 5*E (Nayyeri et al., 2021a) utilized the projective geometry with the five main transitions, namely translation, rotation, inversion, reflection, and homothety, to capture heterogeneous structures such as loop-path subgraphs. (Nayyeri et al., 2021b) embedded KGs on the vector field. UltraE (Xiong et al., 2022) considers a mixture manifold–pseudo-Riemannian space that generalizes hyperbolic and spherical spaces. Other manifold-based KGEs can be found in (Suzuki et al., 2018). Overall, the mentioned models suffer from the same problem mentioned for the Euclidean counterpart, i.e., deterministic transitions between embedded nodes.

3 Preliminaries

Stochastic Process Let T be an arbitrary index set. A *stochastic process* is a collection of random variables $S = \{X_t : t \in T\}$ defined on a probability space $\mathcal{P}$ with the index set T of size $|T|$. All random variables $X_t : \Omega_t \to \mathbb{R}^d, e_t \longmapsto X_t(e_t), t \in T$ belong to the same probability space. A random experiment (realization) is a selection of an outcome $X_t(e_t) \in \mathcal{M}$ at random considering the probability measure $\mathbb{P}$. A *sample path* of a stochastic process $S = \{X_t : t \in T\}$ is a function from t to $X_t(e_t)$ using an ordered index set T, giving us a random walk. Figure 2b illustrates a random walk in the vector space with 50000 steps.

Brownian Motion A stochastic process $B = \{B(t), t \geq 0\}$ defined on a probability space $\mathcal{P} = (\Omega, \mathcal{F}, \mathbb{P})$ ($\mathcal{F}$ is a sigma algebra (events)) is a Brownian motion if a) $B(0) = 0$; b) B has independent and stationary increment, i.e., $B(t_1), B(t_2) - B(t_1), \ldots, B(t_n) - B(t_{n-1})$ are independent random variables for $0 < t_1 < \ldots < t_n$; c) B has Gaussian increments, i.e., $B(t_n + \alpha) - B(t) \sim N(0, \alpha),\ \forall t \geq 0,\ \alpha > 0$; and d) B has continuous sample paths, i.e., B is continuous in t. Figure 2a shows the evolution of two tree structures via the Brownian motion of several particles.

Brownian motion is suitable to model probabilistic branching in random walks as path diffusion when traversing the embedded KG in a vector space. In our setting, we will write interchangeably $B(e_t)$ and $B(\boldsymbol{e}_t)$ for $B(t)$.

Itô Process The stochastic process $S = \{X_t, t \geq 0\}$ that solves the following integral is called an Itô process,

$$X_t = X_0 + \int_0^t a(X_s, s)\, ds + \int_0^t b(X_s, s)\, dB_s, \quad (1)$$

where $t \geq 0$ and X_0 is a scalar starting point. $\{a(X_t, t) : t \geq 0\}, \{b(X_t, t) : t \geq 0\}$ are stochastic processes which are called *drift* and *diffusion*, respectively. B is Brownian motion, dB is normally distributed with zero mean and variance dt.

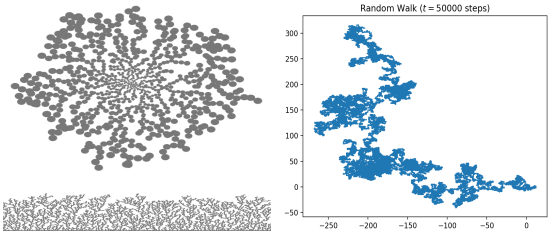
The above formulation of Itô process is approximated by the *Euler-Maruyama* approximation as follows

$$\begin{aligned} &X(t_{n+1}) = X(t_n) + a(X(t_n), t_n)\Delta t + \\ &b(X(t_n), t_n)\Delta B(t_n), \quad t_n = n\Delta t. \end{aligned} \quad (2)$$

Non-Euclidean Geometry In this part, we introduce three popular non-Euclidean manifolds namely spherical $\mathbb{S} = \{\boldsymbol{x} \in \mathbb{R}^{d+1} | \langle \boldsymbol{x}, \boldsymbol{x} \rangle = 1\}$, Hyperboloid $\mathbb{H} = \{\boldsymbol{x} \in \mathbb{R}^{d+1} | \langle \boldsymbol{x}, \boldsymbol{x} \rangle = \frac{1}{K}\}$, and Poincaré ball $\mathbb{B} = \{\boldsymbol{x} \in \mathbb{R}^d \mid \|\boldsymbol{x}\| < -\frac{1}{K}\}$. $K > 0$ is curvature. The tangent space $\mathcal{T}_{\boldsymbol{x}}^K$ at a point $\boldsymbol{x}$ on manifold is a d-dimensional vector space. This space covers all possible directions of paths on a manifold starting from $\boldsymbol{x}$. Each point on the tangent space is mapped to the manifold via an exponential map. Given $\boldsymbol{v}$ as a tangent vector at point $\boldsymbol{x}$ on the manifold, the exponential map for sphere, Hyperboloid and Poincaré ball are

$$\begin{aligned} &exp_{\boldsymbol{x}}(\boldsymbol{v}) = cos(\|\boldsymbol{v}\|)\boldsymbol{x} + sin(\|\boldsymbol{v}\|)\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}, \\ &cosh(\sqrt{|K|}\|\boldsymbol{v}\|)\boldsymbol{x} + \boldsymbol{v}\frac{sinh(\sqrt{|K|}\|\boldsymbol{v}\|)}{\sqrt{|K|}\|\boldsymbol{v}\|}, \\ &\text{and } \boldsymbol{x} \oplus_K (tanh(\sqrt{|K|}\frac{\lambda_{\boldsymbol{x}^K}\|\boldsymbol{v}\|}{2})\frac{\boldsymbol{v}}{\sqrt{|K|}\|\boldsymbol{v}\|}), \end{aligned} \quad (3)$$

respectively. The exponential map projects the tangent vector at a point $\boldsymbol{x}$ on a manifold to another point laying on the geodesic curve, i.e, a curve with



(a) Brownian Tree (b) Random Walk

Figure 2: a) Brownian Tree: The evolution of a tree using Brownian motions over particles, b) a random walk in the vector space associated to a stochastic process.

the shortest distance on the manifold. K is curvature and $\oplus_K$ is the Möbius addition (Balazevic et al., 2019). Note that the tangent vector is defined as $\boldsymbol{v} = \frac{d\boldsymbol{x}}{dt}$, which is orthogonal to the manifold at the point $\boldsymbol{x}$. Later in the paper, we show that by using the Itô process, as stochastic differential equations, we can derive a stochastic process evolving on the manifold.

4 ItôE: Neural Itô Process Embedding

We introduce ItôE, a novel KGE model that utilizes the stochastic processes on manifolds for KG embedding. ItôE is capable of preserving various graph structures and capturing branching relations by modeling the evolution of graph structures in the embedding space as a stochastic process.

KGE models have four essential components: entity and relation representation, score function, and loss function. In the following, these four components of ItôE are explained.

Entity Representation Let's suppose that $\mathcal{E}$ represents the collection of all entities present in the knowledge graph (KG). Each symbolic entity $e \in \mathcal{E}$ in a KG is embedded on a d-dimensional manifold $\mathcal{M}$, i.e., $\boldsymbol{e} \in \mathcal{M}$. Therefore, entity embeddings are points on the manifold. In the proposed model, we use Poincaré Ball, Hyperboloid, Euclidean, and Sphere manifolds.

Relation Representation Each fact in a KG is represented by a triple (e_t, r, e_{t+1}). Because the entities in a triple are subsequent but with arbitrary indexes, we use the notation $(e_{t_n}, r, e_{t_{n+1}})$ to represent the triple, where n is between 1 and $|\mathcal{E}|$ (number of entities in the KG). Most KGE methods model the transition from $\boldsymbol{e}_{t_n}$ to $\boldsymbol{e}_{t_{n+1}}$ via a relation-specific transition (e.g., translation, or rotation). Therefore, each relation is modeled as a point-wise deterministic transition.

However, in a broader view, the relational dependencies between nodes are stochastic processes, i.e., a transition to a tail given a head node and a relation happens with a probability so that the sequence of such probabilistic transitions constitutes a stochastic process. Each random variable in the stochastic process includes all the entities $\mathcal{E}$, a few of which has a non-zero likelihood since the transition from a given entity is only possible to its neighbors. The group of neighbors at each step together with their likelihood in a probability space is considered a random variable. As a result, there

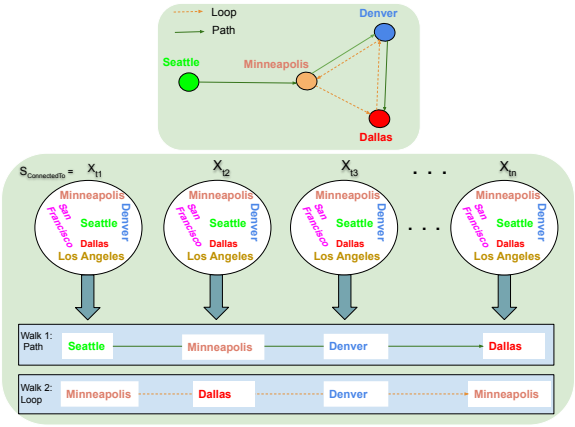


Figure 3: Relation representation: stochastic process forming path-loop on the connected airports.

is a relational mapping between two random variables $X_{t_n}, X_{t_{n+1}}$, associated with the two groups of nodes $\Omega_{t_n}, \Omega_{t_{n+1}}$ at each step. In the following, we model a relation r (defining transitions between nodes) as a stochastic process S_r.

To explain this, let us view the notion of walks formed by a relation r from a stochastic processes angle. A sequence of symbolic entities, connected by a relation r forms a walk with a particular length. A walk of length $n-1$ includes n entities and is shown by $\mathcal{P}_r = \{e_{t_1}, e_{t_2}, \ldots, e_{t_n}\}$.

Assume that there are multiple walks from a set of given entities Ω_{t_1} associated to X_{t_1}, to a set of entities in Ω_{t_n} associated to X_{t_n}. The transition between the nodes in a walk is done randomly according to a distribution.

During traversing the graph from a starting node taken from X_{t_1} to a target node, taken from X_{t_n}, and at each step t_i, there are n_i possible options for selecting the next node. Therefore, traversing a graph with walks of length $n-1$ (with relation r) leads to a stochastic process $S_r = \{X_{t_1}, X_{t_2}, \ldots, X_{t_n}\}$. Consequently, each relation in the KG is represented by a stochastic process.

In this paper, among various stochastic processes, we employ the Itô process for modeling each relation r due to the simplicity of implementation and controlling drift and diffusion. The Itô process is defined as

$$X_t = X_0 + \int_0^t a_r(X_s, s)\, ds + \int_0^t b_r(X_s, s)\, dB_s,\ t \geq 0, \tag{4}$$

where $a_r(.,.)$ and $b_r(.,.)$ are relation-specific drift and diffusion in the Itô integral, respectively. The drift part captures the deterministic transitions, while the diffusion part captures the stochastic tran-

sitions. In this equation, we consider the random variables corresponding to the group of entities that can be seen at each step as a continuous representation. For implementation, we provide the Euler-Maruyama approximation (Jahnke et al., 2012) as a well-known approximation for discrete space:

$$X(t_{n+1}) = X(t_n) + a_r(X(t_n), t_n)\Delta t + b_r(X(t_n), t_n)\Delta B(t_n),\ t_n = n\Delta t, \quad (5)$$

where $X(t_i)$ is the random variable at step i. For simplicity, we set $\Delta t = 1$ from now onward. Note that this approximation is used for the Euclidean manifold. The drift $a_r(X(t_n), t_n)$ and diffusion $b_r(X(t_n), t_n)$ parts are parameterized by two separate multi-layer neural networks. In this way, both drift and diffusion at each step are learned by the model. To enable the Itô process acting on a manifold, it is essential to obtain the tangent vector $\boldsymbol{v}$. The tangent vector determines the direction of movement on manifold $\mathcal{M}$. The exponential map (Equation 3) is used to map the two subsequent random variables on the manifold while considering drift and diffusion as follows

$$X_{t_{n+1}} = exp_{X_{t_n}}(a_r(X(t_n), t_n)\Delta t + b_r(X(t_n), t_n)\Delta B(t_n)),\ t_n = n\Delta t. \quad (6)$$

Therefore, each two subsequent sampled entities lie on a geodesic curve (shortest path). Because the above equation is a stochastic process on a manifold, at each iteration of batch learning, a set of entities in batch triples are observed randomly to hold this equation (realization) as follows:

$$X_{t_{n+1}}(e_{t_{n+1}}) = exp_{X_{t_n}(e_{t_n})}(a_r(X_{t_n}(e_{t_n}, t_n))\ \Delta t + b_r(X_{t_n}(e_{t_n}, t_n))\Delta B(t_n)),\ t_n = n\Delta t,$$

where the equation is held for each sample walk. Note that $X_{t_i}(e_{t_i}) = \boldsymbol{e}_{t_i} \in \mathcal{M},\ \ i = 1, \ldots, n$ is entity embeddings.

Scoring Function For a given triple $(e_{t_n}, r, e_{t_{n+1}})$, the score is as follows

$$f(e_{t_n}, r, e_{t_{n+1}}) = -\|\boldsymbol{e}_{t_{n+1}} - exp_{\boldsymbol{e}_{t_n}}(a_r(\boldsymbol{e}_{t_n}, t_n)\ \Delta t + b_r(\boldsymbol{e}_{t_n}, t_n)\Delta B(t_n))\|,\quad t_n = n\Delta t.$$

For positive (negative) triples, $f(e_{t_n}, r, e_{t_{n+1}})$ is a high (low) value. That is, the two sampled entities $e_{t_n}, e_{t_{n+1}}$ lie on a geodesic curve on a manifold.

Loss Function For training the model, we use the following loss function (Chami et al., 2020)

$$\mathcal{L} = \sum_{e' \in \mathcal{E}} log(1 + exp(y_{e'}(f(e_{t_n}, r, e') + \delta_{e_{t_n}} + \delta_{e'}))),$$

where $y_{e'} = 1$ if $e' = e_{t_{n+1}}$, and $y_{e'} = -1$ if $e' \neq e_{t_{n+1}}$, and $\delta_{e_{t_n}}$ and $\delta_{e'}$ are trainable entity biases.

In the next section, we present important insights about our formulation, followed by theoretical justification for the core formulation of our model.

5 Insights and Theoretical Analysis

Memory Complexity In Itô*E*, the number of relation parameters grows linearly with the relation's dimensionality. Hence, Itô*E*'s space complexity is $O(N_e \times d_e + N_r \times d_r)$, where N_e and N_r are the numbers of entities and relations, d_e and d_r are the embedding dimensionality of entities and relations, respectively. The additional parameters come from the neural network that approximates the Itô*E*' process, which is in our case, shared across all entities.

Relational Sub-structures Here we show the capability of the stochastic process in equation (5) for modeling various graph sub-structures. To this end, let us have a sample walk from the stochastic process S_r, which is $\{X_{t_1}(e_{t_1}), \ldots, X_{t_n}(e_{t_n})\}$ = $\{\boldsymbol{e_{t_1}}, \ldots, \boldsymbol{e_{t_n}}\}$. A stochastic process S_r inherently covers various graph structures such as hierarchy, loop, and path. This is due to the fact that various sample walks over a stochastic process generate different parts of a subgraph. Figure 3 shows an example of a stochastic process for the relation $ConnectedTo$ which forms several relational and structural patterns such as loop and path by sampling from the process in the airport example (airports are connected with various shapes). Another example is a tree-like structure. For tree-like structures (see Figure 1), X_{t_1} contains only a root node, and X_{t_n} contains all the leaves. $X_{t_i}, i = 2, \ldots, n-1$ generates the intermediate nodes from the root to the leaves. Therefore, any walk from the root node to each of the leaves is a sample taken from the stochastic process. The relation-specific stochastic process generates various sample walks at random covering various parts of a sub-graph.

Subsumption of Other KGE Models Itô*E* provides a general framework that covers various baselines and state-of-the-art KGE models. We prove,

in this part, that our model subsumes TransE, RotatE, QuatE, 5*E, ComplEx, and DistMult. That is, given any set of triples with arbitrary true/false labeling, any score value represented by each of the mentioned models for the triples in the set is also represented by the score function of Itô*E*. In this regard, the following theorem holds:

Theorem 1. *Itô*E subsumes TransE, RotatE, DistMult, ComplEx, QuatE, and 5*E.*

As a consequence of Theorem 1, Itô*E* is fully expressive and capable of capturing various graph structures and relational patterns that each of the mentioned models is capable of. Therefore, the following corollaries hold:

Corollary 1. *Itô*E is fully expressive, i.e., for every ground truth over an arbitrary KG, there are assignments to the entities and relations embedding to capture the ground truth.*

Corollary 2. *Itô*E models symmetric, anti-symmetric, composition, inversion, transitive, and reflection patterns.*

Corollary 3. *Itô*E models one-to-many relation.*

Corollary 4. *a) Let $\mathcal{L}^n_{r_1}$ be a loop structure with a single relation r_1 and n nodes. Itô*E models the loop structure $\mathcal{L}^n_{r_1}$.*

*b) Let $\mathcal{P}^n_{r_2}$ be a path structure with relation r_2 and n nodes. In addition, each nodes of $\mathcal{L}^n_{r_1}$ is connected to one node in $\mathcal{P}^n_{r_1}$. The combined structure is denoted by $\mathcal{LP}^n_{r_1r_2r_3}$. Itô*E models $\mathcal{LP}^n_{r_1r_2r_3}$.*

*c) Itô*E models loop-path structure with single relation r_1, i.e., $\mathcal{LP}^n_{r_1r_1r_1}$.*

6 Experiments and Results

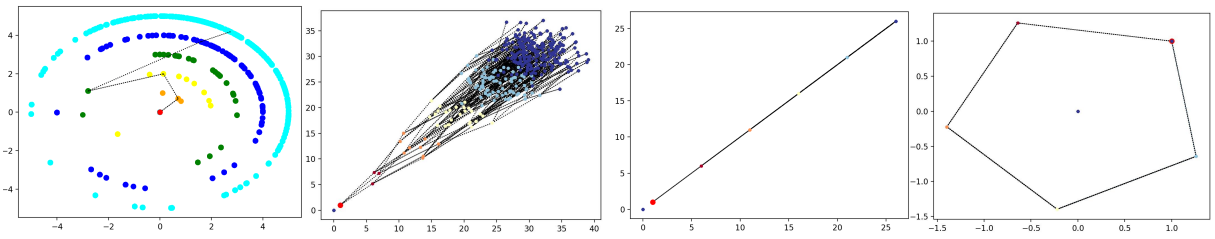
Experimental Setup In this section, we evaluate the performance of Itô*E* against various state-of-the-art KGE models in the link prediction task. Our evaluation in this section includes link prediction in low-dimensional space, analysis on capturing complex structures, analysis of capturing hierarchical structures, and time and memory complexity. Further evaluations including results per manifold, variance of results of our model, influence of embedding dimension, and discussion on the effect of loss function on deterministic models can be found in appendix.

Evaluation Metrics We use four standard metrics for link prediction namely Mean Reciprocal Rank (MRR), Hits@k (k=1,3,10). To compute each of the metrics, we use the procedure in (Bordes et al., 2013; Lacroix et al., 2018). For each test triple $(e_{t_n}, r, e_{t_{n+1}})$, we first replace the head entity e_{t_n} by each of the entities in the dictionary, i.e., $e' \in \mathcal{E} - \{e_{t_n}\}$. This results in n_e corrupted triples $\{(e', r, e_{t_{n+1}})\}$, where n_e is the number of entities in the KG. We filtered this set by removing all triples that are already appeared the dataset as well as self loop. We then compute the scores of the original test triple $(e_{t_n}, r, e_{t_{n+1}})$ and the corrupted triples $\{(e', r, e_{t_{n+1}})\}$, sort them based on scores and rank them.

The resulted rank of the original triple is the left rank r_l. The same procedure is performed to compute the right rank by the corruption of the tail entities. r_r denotes the right rank. The average of the left and the right ranks is denoted by r_a. The mean rank of all testing triples is MR. The percentage of the test triples ranked lower than $k = 1, 3, 10$ denotes Hits@k. MRR is the average reciprocal of rank for all testing triples.

Environment and Hyperparameters We implemented[1] our model using Python and PyTorch library. We added reciprocal relations to the training samples as a standard technique used in (Kazemi and Poole, 2018; Lacroix et al., 2018). We added $N3$ regularization for training the models (Lacroix et al., 2018). Because one of the main goal of this paper is modeling graphs in low dimensional space, we follow the common practice of existing works in low dimensional embedding (Chami et al., 2020) and trained the models in a low dimension ($d = 32$). The other dimensions have been done as further analysis in the appendix. We split data into several batches and used the Adagrad/Adam as an optimizer. An early stopping technique based on validation MRR has been used to terminate the running and perform testing. Batch size b, learning rate lr, $N3$ regularization coefficient α are among the hyperparameters used in this paper. In addition, we set the number of hidden layers for drift and diffusion neural networks to two. l denotes the number of neurons in the hidden layer of each neural networks. The used distribution for Brownian motion ΔB is a normal distribution with zero mean and σ variance. For simplicity, we set $\sigma = 1$. Due to randomness, for our model, we perform experiments 10 times and report the average results in Table 1. Because the variances were low, we did not report them in the main table. The manifold $\mathcal{M}$ is selected from *Poincaré ball* $\mathbb{B}$, *Hyperboloid*

[1] https://github.com/ColdMist/ItoE

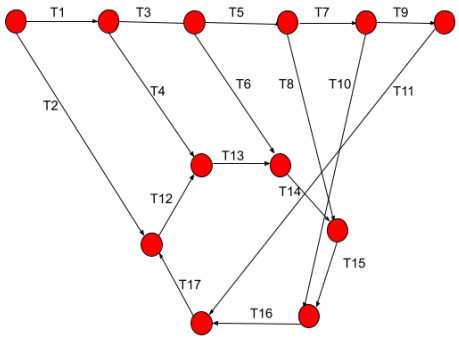


(a) Itô process Sphere Tree (b) Itô Process Tree (c) Translation Tree (d) Rotation Tree

Figure 4: Generation of tree in the embedding space by using the transiting functions of a) Itô process on Sphere. One trajectory from root to a leaf is plotted with a dashed line. b) Itô process. Several trajectories form tree structures in the embedding space, c) translation. A single trajectory, d) rotation. a single trajectory. In each level l, the tree contains n^l nodes.

model $\mathbb{H}$, *Euclidean real manifold* $\mathbb{R}$ and *spherical* manifold $\mathbb{S}$. The optimal hyperparameters per each dataset are reported in a separate table in appendix.

Dataset We used the two standard benchmark datasets namely FB15k-237 (Toutanova and Chen, 2015) and WN18RR (Dettmers et al., 2018) for evaluating ItôE on static KGs. Both datasets contain structural and relational patterns including symmetric/anti-symmetric and composition patterns. In addition, WN18RR contains hierarchical structures associated with *hypernym* and *part-of*. Furthermore, both datasets include various types of relationships, including one-to-one relationships, one-to-many relationships (where a subject can have multiple objects but each object has only one subject), many-to-one relationships, and many-to-many relationships (where a subject can have multiple objects and an object can have multiple subjects).



(a) Loop-path

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17
ItoE	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
TransE	5	5	4	5	5	5	2	3	6	5	4	3	6	5	3	2	5
RotatE	5	4	5	5	10	1	4	9	8	6	5	7	6	6	6	2	5
ComplEx	2	3	8	5	1	3	7	11	2	10	7	8	5	6	2	2	5
QuatE	2	1	2	1	2	5	5	3	5	2	1	5	2	1	1	4	4
RotH	1	1	2	5	2	2	1	2	1	3	5	4	5	1	2	2	4
RefH	3	3	2	2	2	1	1	1	1	2	2	4	1	4	3	2	2
AttH	3	2	2	6	2	1	1	5	1	2	4	5	3	2	1	1	1

(b) Heatmap of ranking

Figure 5: Ranking results of loop connected to a path.Ti refers to the ith triple in the graph. The ith cell contains the rank of the ith triple Ti.

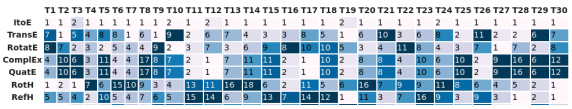


Figure 6: The heatmap of ranking results given by our model and the other competitors on the graph in Figure 7. For real models the embedding dimension is set to 4, and for complex models, the embedding dimension is 2. Ti refers to the ith triple in the graph. The ith cell contains the rank of the ith triple Ti.

KGE Models Two classes of baselines and state-of-the-art KGE models have been selected as competitors: a) baselines KGEs: TransE, DistMult, and ComplEx, b) state-of-the-art KGEs: RotatE, QuatE, 5*E, MurP/MurE, and RotH/RefH/AttH. We trained these models using entity bias, and cross-entropy loss, regularization, and also we added reverse triples as in (Chami et al., 2020). Such techniques improves the performance of the models including TransE comparing to their original results.

MurP, RotH/RefH/AttH employ Poincaré ball as a non-Euclidean manifold to preserve hierarchical structures. RotH additionally takes the advantage of rotation in hyperbolic space to model various relational patterns such as symmetry, anti-symmetry, inversion, and composition. To train ComplEx, QuatE, RotH/RefH/AttH, MurP, DistMult, and our model, we enriched the data with reverse triples, as the standard technique employed in (Kazemi and Poole, 2018), and used $N3$ regularization (Lacroix et al., 2018).

	Model	FB15k-237				WN18RR			
		MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Euc	TransE	.295	.210	.322	.466	.366	.274	.433	.515
	RotatE	.051	.029	.051	.091	.309	.293	.317	.336
	ComplEx	.287	.203	.316	.456	.421	.391	.434	.476
	QuatE	.293	.212	.320	.460	.421	.396	.430	.467
non-Euc	MuRP	.321	.239	.352	.495	.473	.421	.484	.546
	5*E	.323	.240	.355	.501	.449	.418	.462	.510
	REFH	.316	.229	.345	.490	.449	.418	.462	.510
	ROTH	.315	.226	.348	.491	.477	.426	.490	.548
	ATTH	.321	.240	.355	.501	.465	.426	.481	.540
	Itô E ($\mathbb{R}$)	.330	.242	.361	.508	.455	.404	.480	.548
	Itô E ($\mathbb{S}-\mathbb{P}$)	.334	.245	.361	.511	.474	.426	.499	.574

Table 1: Link prediction results on FB15k-237 and WN18RR (low dimension 32) for Euclidean (*Euc*) and non-Euclidean (*non-Euc*) models.

Results Table 1 shows the results of Itô E (Poincare ($\mathbb{P}$), Euclidean ($\mathbb{R}$), Sphere ($\mathbb{S}$)) and other models in low-dimensional embedding $d = 32$ on FB15K-237 and WN18RR. Note that $d = 32$ is a common practice of KGE literature for evaluation of the models in low dimensional embedding (Chami et al., 2020). According to our experiments, Itô E with the Poincaré ball outperforms all models on WN18RR dataset which contains mainly hierarchical relations such as *hypernym* and *part-of*. This dataset also contains relations forming loop structures such as *similar-to*. MuRP, REFH, ROTH, and ATTH utilize Poincaré ball with deterministic transitions (e.g., rotation, reflection, translation, and Affine mapping). Itô E with the stochastic transition on Poincaré ball outperforms all of these models on WN18RR across all metrics. This is especially visible by looking at Hits@3 and Hits@10. The results show the superiority of stochastic transitions over deterministic transitions to model various structures such as Hierarchical and loop structures in a low-dimensional space. Using FB15k-237, Itô E with Spherical manifold outperforms other competitors including the Hyperbolic models. Figure 8 presents performance per dimension. As shown in the figure, our model outperforms other KGE models in low dimensions. In high dimension, our model get competitive performance to other models on WN18RR dataset.

Capturing hierarchical structure In this part, we generate tree structure in the embedding space by using the transition functions of the Itô process, translation, and rotation. Starting from a point on the embedding space (i.e., root shown in red in Figure 4), a model generates n child for each node and traverses the vector space to generate the whole tree structure. As shown in Figure 4b, the Itô process generates the tree-like structure by generating stochastic trajectories in the vector space. However, translation (Figure 4c) and rotation (Figure 4d) generate a single trajectory which consequently cannot traverse the vector space to generate a tree structure. Figure 4a illustrates the evolution of tree structure in a sphere. The root is located on the center and the leaves are distributed on the border of the sphere. All the intermediate nodes on the lth level of the tree are distributed on a sphere with radius of r_l, inside the main sphere where $r_{l1} < r_{l2}$, $l_1 < l_2$. A trajectory (dashed line) is shown from root to leaf.

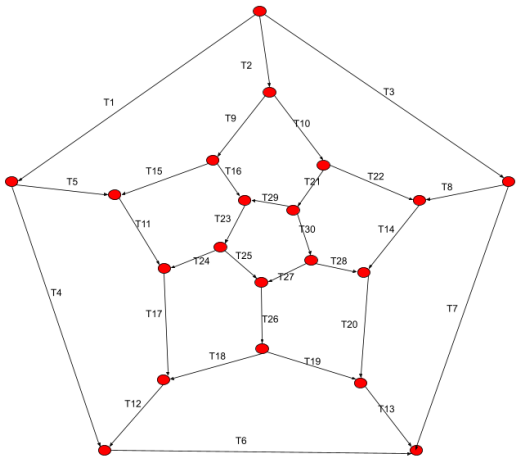


Figure 7: Complex graph structure with 30 triples.

Table 2: Number of model parameters and training time (s) on WN18RR with dimension 32.

Model	N-Parameters	Time
TransE	1392766	40s
MurE	1393470	55s
RefH	1394196	120s
AttH	1395604	240s
RotH	1394196	120s
ItôE	1395100	74s

Capturing Complex structures Here we examine ItôE and several other Euclidean and non-Euclidean models to preserve various complex structures. We train the models on the substructures and then present the ranking results given by the models in a heatmap. For the graph in Figure 7, the heatmap is presented in Figure 6. As shown in the figure, ItôE gets a very low rank (mostly 1 which is ideal) to the graph edges. This shows that the model learns the structure. For other models including manifold-based models, e.g., AttH and RotH, the ranking for the edges are high, i.e., these models do not learn the structure. Figure 5 shows the ranking results of modeling heterogeneous structure on the example of a loop connected to a path.

Memory and Time Complexity Table 2 shows the training time (per epoch) and the number of model parameters for ItôE, TransE, MurE, RefH, RotH and AttH. According to the table, ItôE has a close number of parameters to other state-of-the-art models. Among the models, TransE is the most efficient model in terms of the number of parameters. In addition, our model is competitive with other models in terms of training time.

7 Conclusion

This paper presented ItôE, a knowledge graph embedding model that considers the stochastic transitions between nodes of a knowledge graph on a manifold. For doing so, ItôE modeled the relations in a KG as stochastic processes so that the transitions between two nodes could only happen with an associated likelihood. Such stochastic transitions allowed ItôE to present multiple stochastic trajectories between any two embedded nodes and to capture more sophisticated structures in KGs, including loops connected to paths and is mathematically proved to be a generalization of several state-of-the-art models. Experiments on the synthesized datasets showed that the proposed model can capture heterogeneous complex structures and patterns.

Limitations

In this section, we discuss the limitation of the proposed model. Currently, the hidden layer of the two neural networks for drift and diffusion are shared between all entities and relations. This might cause over-fitting on relations that show simple structures if the neural networks are set to be very deep. On the other hand, if the neural networks are set to be shallow it might negatively influence modeling complex relations as the direction of trajectories will be limited. One possible solution is to cluster relations based on complexities and use separate neural networks for each cluster depending on the complexity of the corresponding relation. This requires, however, prior knowledge about the structure of different relations which we leave as future work.

Ethics Statement

The authors declare that we have no conflicts of interest. This article does not contain any studies involving business data and personal information.

Acknowledgements

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Bo Xiong. Bo Xiong is funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No: 860801. Mojtaba Nayyeri is funded by the German Federal Ministry for Economic Affairs and Climate Action under Grant Agreement Number 01MK20008F (Service-Meister). This research was partially funded by the Ministry of Science, Research, and the Arts (MWK) Baden-Württemberg, Germany, within the Artificial Intelligence Software Academy (AISA) and the German Research Foundation (DFG) via grant agreement number STA 572/18-1 (Open Argument Mining). We acknowledge the support of the Stuttgart Center for Simulation Science (SimTech). The authors would like to thank the reviewers for their constructive comments and suggestions.

References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. Multi-relational poincaré graph embeddings. *Advances in Neural Information Processing Systems*, 32:4463–4473.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI conference on artificial intelligence*.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Zexi Huang, Arlei Silva, and Ambuj Singh. 2021. A broader picture of random-walk based graph embedding. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 685–695.

Tobias Jahnke, Tudor Udrescu, and Marcel Mikl. 2012. Numerical methods in mathematical finance.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

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*, pages 4289–4300.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, pages 2863–2872. PMLR.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th international conference on World Wide Web*, pages 1211–1220.

Mojtaba Nayyeri, Sahar Vahdati, Can Aykul, and Jens Lehmann. 2021a. 5* knowledge graph embeddings with projective transformations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9064–9072.

Mojtaba Nayyeri, Chengjin Xu, Franca Hoffmann, Mirza Mohtashim Alam, Jens Lehmann, and Sahar Vahdati. 2021b. Knowledge graph representation learning using ordinary differential equations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9529–9548.

Tu Dinh Nguyen, Dat Quoc Nguyen, Dinh Phung, et al. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 327–333.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 1(104):11–33.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Icml*.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30:6338–6347.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Jan Portisch and Heiko Paulheim. 2022. Walk this way! entity walks and property walks for rdf2vec. *arXiv preprint arXiv:2204.02777*.

Petar Ristoski and Heiko Paulheim. 2016. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*, pages 498–514. Springer.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

Atsushi Suzuki, Yosuke Enokida, and Kenji Yamanishi. 2018. Riemannian transe: Multi-relational graph embedding in non-euclidean space.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pages 57–66.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Melanie Weber and Maximilian Nickel. 2018. Curvature and representation learning: Identifying embedding spaces for relational data. *NeurIPS Relational Representation Learning*.

Bo Xiong, Shichao Zhu, Mojtaba Nayyeri, Chengjin Xu, Shirui Pan, Chuan Zhou, and Steffen Staab. 2022. Ultrahyperbolic knowledge graph embeddings. In *KDD*, pages 2130–2139. ACM.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. *arXiv preprint arXiv:1904.10281*.

A Appendix

We organize the appendix as follows: We first present the proof of Theorem 1, and corollaries 1-4. We then discuss the effect of loss functions on deterministic models, followed by the results per manifold and hyperparameters specification.

Proof of Theorem 1

Proof. Here we prove that Itô*E subsumes 5*E (Nayyeri et al., 2021a). We start with the formulation of 5*E. For simplicity, we remove the index of relation from the relation matrix in 5*E and consider one-dimensional complex projective line. The following equation is modeling triple in the vector space by 5*E

$$\begin{aligned} & e^p_{t_{n+1}} \approx \eta e^p_{t_n}, \\ & \eta = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \\ & e^p_{t_n} = \begin{pmatrix} e_{t_n} \\ 1 \end{pmatrix}, \\ & a, b, c, d, z \in \mathcal{C}. \end{aligned} \tag{7}$$

Let $\eta = \begin{pmatrix} a_r + a_I i & b_r + b_I i \\ c_r + c_I i & d_r + d_I i \end{pmatrix} = \begin{pmatrix} a_r & b_r \\ c_r & d_r \end{pmatrix} + \begin{pmatrix} a_I & b_I \\ c_I & d_I \end{pmatrix} i = \eta_r + \eta_I i$, and $e^p_{t_n} = \begin{pmatrix} e^r_{t_n} \\ 1 \end{pmatrix} + \begin{pmatrix} e^I_{t_n} \\ 0 \end{pmatrix} = e^{pr}_{t_n} + e^{pI}_{t_n}$. We have

$$e^p_{t_{n+1}} = \eta e^p_{t_n} = (\eta_r e^{pr}_{t_n} - \eta_I e^{pI}_{t_n}) + (\eta_r e^{pI}_{t_n} - \eta_I e^{pr}_{t_n}) i. \tag{8}$$

We merge the matrices and rewrite the formulation as follows:

$$e^p_{t_{n+1}} = \begin{pmatrix} \eta_r & \eta_I \end{pmatrix} \begin{pmatrix} e^{pr}_{t_n} \\ -e^{pI}_{t_n} \end{pmatrix} + \begin{pmatrix} \eta_r & \eta_I \end{pmatrix} \begin{pmatrix} e^{pI}_{t_n} \\ e^{pr}_{t_n} \end{pmatrix} i. \tag{9}$$

Let $\eta_{rI} = \begin{pmatrix} \eta_r & \eta_I \end{pmatrix}, e^{rI}_{t_n} = \begin{pmatrix} e^{pr}_{t_n} \\ -e^{pI}_{t_n} \end{pmatrix}, e^{Ir}_{t_n} = \begin{pmatrix} e^{pI}_{t_n} \\ e^{pr}_{t_n} \end{pmatrix}$.

Therefore, we have $e^p_{t_{n+1}} = \eta_{rI} e^{Ir}_{t_n} + \eta_{rI} e^{Ir}_{t_n} i$.

The above formulation can be rewritten in the vectored form as follows

$e^p_{t_{n+1}} = \eta_{rI} \begin{pmatrix} e^{rI}_{t_n} \\ e^{Ir}_{t_n} \end{pmatrix} = \eta_{rI} e^{IrI}_{t_n}$.

The formulation of Itô*E is

$\mathbf{e}_{t_{n+1}} \approx \boldsymbol{e}_{t_n} + a_r(\boldsymbol{e}_{t_n}, t_n)\Delta t + b_r(\boldsymbol{e}_{t_n}, t_n)\Delta B(t_n)$.

If $\Delta t = 1, b_r(\boldsymbol{e}_{t_n}, t_n) = 0$, there is a neural network $a_r(\boldsymbol{e}_{t_n}, t_n)$ that approximates the multivariate function $(\eta_{rI} - I)e^{IrI}_{t_n}$ with the error as close as zero due to universal approximation ability of the NNs. Therefore, Itô*E can approximate the score of 5*E with an arbitrary small error. Consequently, Itô*E subsumes 5*E. Because 5*E subsumes TransE, RotatE, ComplEx, Itô*E subsumes these models as well.

We now prove that Itô*E subsumes QuatE. Considering the formulation of Ito*E,

$\mathbf{e}_{t_{n+1}} = \boldsymbol{e}_{t_n} + a_r(\boldsymbol{e}_{t_n}, t_n)\Delta t + b_r(\boldsymbol{e}_{t_n}, t_n)\Delta B(t_n)$,

and setting $\Delta t = 1, b_r(\boldsymbol{e}_{t_n}, t_n) = 0, a_r(\boldsymbol{e}_{t_n}, t_n) = (\mathbf{R} - I)\mathbf{e}^v$, where $\mathbf{e}^v$ is the vector representation of a Quaternion number, the assumption of QuatE is fulfilled by Itô*E

$$\mathbf{e}^v_{t_{n+1}} \approx \mathbf{R}\mathbf{e}^v_{t_n}. \tag{10}$$

Therefore, Itô*E subsumes QuatE (Zhang et al., 2019). □

Proof of Corollaries

Proof. Here we present the proof of Corollaries 1-4. Because Itô*E subsumes 5*E, ComplEx, QuatE, RotatE, and TransE, it can encode all relational and structural patterns (symmetric, anti-symmetric, reflexive, transitive, inverse, and combination of loop and path) modeled by these models. Moreover, the Itô*E model is fully expressive because it subsumes ComplEx which is fully expressive. □

Effect of Loss Function on Deterministic Models As mentioned in the paper, the models based on deterministic transitions such as TransE with $\mathbf{e}_{t_n} + \mathbf{r} = \mathbf{e}_{t_{n+1}}$ provide a single trajectory between any two nodes. However, by using a loss function forcing an upper-bound for the score of positive samples we have $\mathbf{e}_{t_n} + \mathbf{r} = \mathbf{e}_{t_{n+1}} + \epsilon$. This allows the models based on deterministic transitions to mitigate the problem of a single trajectory. However, the problem is not fully solved because the nodes after each transition are embedded very closely. In this way, there is a single trajectory that is retrievable by transition function in which the other embedded nodes are not reachable by using the transition function in the embedding space. In contrast, using the stochastic transitions, the model can learn at each embedded node $\mathbf{e}_{t_n}$ via diffusion NN, $b_r(\mathbf{e}_{t_n}, t_n)$, the degree of branching, i.e., degree of diffusion. Therefore, different trajectories

Model	Dataset	Neg Samp.	batch_size	l_rate	reg_co	Epochs
Itô̂E	WN18RR	500	500	0.001	0	300
Itô̂E	FB15k-237	500	50	0.05	0	300

Table 3: Best hyperparameters found for Itô̂E

Table 4: Variance of MRR for Itô̂E

WN18RR	FB15k-237
.000004	.000009

are learned between any two nodes which can be either very close or far.

Variance in Itô̂E Performance Table 4 shows the variance of Itô̂E with dimension 32 on WN18RR. As shown in the table, the variance of 10 times running on the model is very low. Therefore, the model obtains stable performance on different runs.

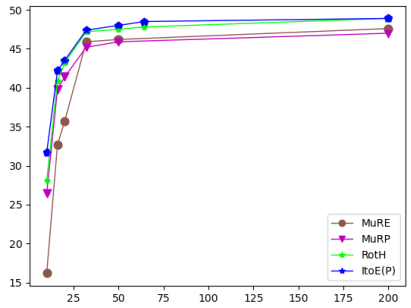


Figure 8: Performance of embedding models based on embedding dimension on WN18RR. MuRE(Balazevic et al., 2019) uses Euclidean manifold. MuRP (Balazevic et al., 2019), RotH (Chami et al., 2020) and Itô̂E (our model) uses Poincare ball. The x-axis is embedding dimension and the y axis is performance based on MRR.

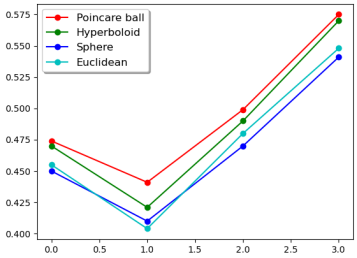


Figure 9: Performance (MRR, Hits@(1,3,10)) of Itô̂E using Poincare ball, Hyperboloid, Spherical and Euclidean manifolds.

Results On WN18RR Per Manifolds In this part, we analyze the performance of Itô̂E using various manifolds (Poincare ball, Hyperboloid, Euclidean, and Sphere) on WN18RR. Figure 9 illustrates the performance comparison according to different metrics namely MRR, Hits@1, Hits@3, Hits@10. The experiments have been done in a very low dimension of 32. According to the figure, Itô̂E with Poincare ball and Hyperboloid outperformed Itô̂E with Sphere and Euclidean manifold. This is consistent with the nature of the used KG where most relations are hierarchical in WN18RR.