Tracking Uncertainty in Knowledge Graphs Using Kalman Filtering

by

Alina Tkachenko
B.Sc., Odesa I. I. Mechnikov National University, 2021

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

We acknowledge and respect the Lək̓ʷəŋən (Songhees and Xʷsepsəm/Esquimalt) Peoples on whose territory the university stands, and the Lək̓ʷəŋən and W̱SÁNEĆ Peoples whose historical relationships with the land continue to this day.

Tracking Uncertainty in Knowledge Graphs Using Kalman Filtering

by

Alina Tkachenko
B.Sc., Odesa I. I. Mechnikov National University, 2021

Supervisory Committee

Dr. Alex Thomo, Supervisor
(Department of Computer Science)

Dr. Kevin Stanley, Departmental Member
(Department of Computer Science)

ABSTRACT

Knowledge graphs (KGs) represent structured knowledge as networks of entities and relations, forming a foundation for reasoning in artificial intelligence. To make these symbolic structures usable by machine learning systems, knowledge graph embeddings (KGEs) map entities and relations into continuous vector spaces. However, traditional KGE models are typically static and deterministic. They treat all facts as equally certain and require full retraining when new data arrive, making them unsuitable for evolving, uncertain knowledge.

This thesis introduces a new framework that reframes knowledge graph embedding as an *online state estimation* problem. By integrating the Kalman filter, a recursive algorithm that updates beliefs under uncertainty, into KGE training, the proposed approach enables continuous and uncertainty aware learning of entity and relation representations. The framework treats each embedding as a probabilistic latent state, updated incrementally as new triples arrive, blending prior knowledge with new, possibly noisy, observations.

Two models instantiate this framework. **KalmanKG2E** extends the probabilistic Gaussian embedding model KG2E with Kalman-based online updates of means and covariances. **KalmanComplEx** adapts the non-probabilistic, complex-valued ComplEx model to a dynamic, uncertainty-tracking setting. Together, these demonstrate the frameworks generality across fundamentally different embedding architectures.

Extensive experiments on six benchmark datasets show consistent improvements over static baselines. The Kalman-based models converge faster, achieve higher predictive accuracy, and exhibit greater robustness in sparse, evolving graphs. These results validate Kalman filtering as a principled and efficient mechanism for online knowledge graph learning.

Overall, this work bridges classical state estimation and modern representation learning, advancing knowledge graph embeddings from static snapshots to dynamic, continuously adaptive systems that better reflect the evolving nature of real-world knowledge.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to sincerely thank my supervisor, Dr. Alex Thomo, for his guidance, encouragement, and thoughtful feedback throughout this research. His expertise and mentorship were invaluable to the development and completion of this thesis.

I am also grateful to Dr. Kevin Stanley, a member of my supervisory committee, for his valuable input, time, and support.

I would like to acknowledge the Department of Computer Science at the University of Victoria for providing an excellent academic environment, resources, and support, which were essential for the successful completion of my graduate studies.

Research is formalized curiosity. It is poking and prying with a purpose.
Zora Neale Hurston

Chapter 1

Introduction

1.1 Motivation

Human progress has always depended on how we capture and structure knowledge. Ancient libraries cataloged scrolls, encyclopedias distilled centuries of thought, and todays digital platforms integrate information at planetary scale. In artificial intelligence, this long tradition finds its modern counterpart in the *knowledge graph* (KG): a structured network in which entities are connected by typed relations, forming a web of interlinked facts that machines can interpret, traverse, and reason over [1, 2, 3].

A knowledge graph represents information as triples (h, r, t), each expressing a factual statement connecting a *head entity* h to a *tail entity* t through a *relation* r. For example:

- `(Paris, is_capital_of, France)`
- `(France, located_in, Europe)`

Individually, these triples are simple. Collectively, they form a structured fabric of knowledge, a network in which meaning arises from connection. This representation enables machines to move from raw data toward reasoning and discovery, supporting search, analytics, and question answering at scale [4, 5].

The idea of connecting knowledge through relationships has deep roots. From early ontologies and semantic networks to todays large-scale graph infrastructures, the goal has remained constant: to make meaning explicit by modeling how things relate [6, 7]. Modern KGs like Wikidata, DBpedia, and Googles Knowledge Graph have become central to how digital systems organize and interpret information [5, 3].

In scientific domains, specialized graphs such as UniProt and BioKG link molecular entities, diseases, and experimental findings, helping researchers navigate complex relational structures [8].

Recent surveys emphasize that knowledge graphs now form a cornerstone of modern AI ecosystems, bridging structured data, symbolic reasoning, and natural language understanding [1, 2]. They provide a shared substrate on which reasoning systems, search engines, and even foundation models for knowledge reasoning are built [9, 10].

Yet a graph on its own is only a symbolic structure. To make KGs useful for computation, we need a way to translate this symbolic structure into a mathematical language that machine learning models can work with. This is the role of *knowledge graph embeddings* (KGEs). Traditional KGE models map each entity and relation into a low-dimensional vector space as point vectors, enabling algebraic operations to capture semantic relationships: distances can encode similarity, directions can model relations, and geometric structure can support reasoning tasks such as link prediction, entity classification, and recommendation [11, 12].

However, representing embeddings as static point vectors has significant limitations. Point embeddings fail to capture inherent uncertainties in knowledge graphs, such as ambiguous relations, noisy data, or varying degrees of confidence in facts. For instance, a relation like "influences" between two historical figures might be debatable or context dependent, but a point vector treats it as a fixed, deterministic link. This brittleness can lead to poor generalization, especially in incomplete or evolving graphs where not all facts are equally certain [13].

To address these shortcomings, a more advanced approach is to represent embeddings not as fixed points but as *probability distributions* in the vector space. Standard embedding models implicitly assume that all facts are equally reliable, but in real-world knowledge graphs this assumption rarely holds. Some entities appear in thousands of triples while others occur only a few times; some relations are semantically precise, others ambiguous or context dependent. Such heterogeneity requires representations that can express different degrees of confidence and variability across the graph.

Probabilistic KGE models, such as those using Gaussian distributions, encode entities and relations as densities (for example, means and covariances), explicitly modeling uncertainty [13, 14]. In this paradigm, the mean vector represents the cen-

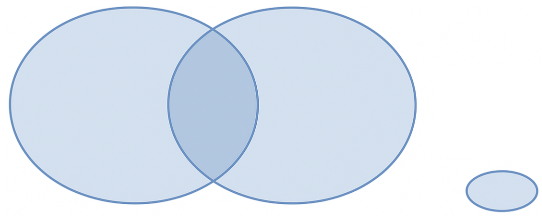

Figure 1.1: Entities represented as Gaussian distributions in the embedding space. Larger ellipses indicate higher uncertainty, while overlap reflects shared semantic context.

tral estimate of an entity or relation, while the variance or covariance describes the spread of possible positions, reflecting confidence levels. Entities with rich evidence form tight, low variance distributions, whereas newly introduced or weakly connected entities are represented with broader, high variance ones that express limited knowledge.

This distributional view provides a richer geometric and semantic framework. Overlap between distributions can measure similarity or compatibility, while probabilistic scoring functions can incorporate uncertainty directly into predictions, improving robustness under noise or incompleteness [15, 16]. For example, Figure 1.1 illustrates three entities: "CRISPR-Cas9", "mRNA vaccine, and "Polymerase," each represented as a Gaussian. The larger ellipses for CRISPR-Cas9 and mRNA vaccine reflect greater uncertainty and show partial overlap, capturing both their emerging status and shared scientific context. In contrast, "Polymerase" appears as a tightly clustered ellipse, representing an older, well-established concept with low uncertainty. Without uncertainty modeling, these ellipses would collapse to single points at their means, losing this richer view.

By moving beyond deterministic points to probabilistic densities, embedding models better mirror the nuanced, evolving nature of real-world knowledge and provide a foundation for adaptive updating in dynamic graphs.

Uncertainty and Updates Yet, even with probabilistic representations, the world described by knowledge graphs is never fixed. New discoveries are made, old facts are corrected, and previously unknown entities enter the scene. Examples abound:

- In biomedical graphs, new drugtarget interactions are published continuously [17].
- In search engines, knowledge about public figures and events shifts daily [4].

- In protein databases such as UniProtKG, annotations are revised on a near-daily basis [8].

If embeddings, whether point based or distributional, are learned once and left unchanged, they quickly become stale. Retraining the entire model from scratch is costly and often impractical for knowledge graphs with millions of triples. The central problem, then, is how to give embeddings the same adaptability and fluidity as the knowledge they represent while preserving their probabilistic nature. The key question is how to update probabilistic embeddings as new triples arrive without revisiting the entire graph. We need a principled way to merge prior distributions with new, possibly noisy or conflicting, evidence quickly enough to keep pace with evolving data.

Enter the Kalman Filter. This is precisely the problem that the Kalman filter was invented to solve [18]. Originally developed for aerospace navigation and control [19], the Kalman filter estimates the hidden state of a system that evolves over time, given noisy and partial observations [20]. It balances:

- **Prior state:** what we believed before, encoded in the embedding distribution (mean and covariance).
- **New evidence:** the latest triple, which may be noisy or incomplete.
- **Noise terms:** measurement noise R (how much to trust the new fact) and process noise Q (how much to allow the state itself to drift).

The Kalman update then blends these sources in closed form, producing an updated mean and covariance for each embedding. In the context of KGs, this means entities and relations can evolve smoothly, tracking both knowledge and uncertainty as the graph changes, all while maintaining the probabilistic representation.

Motivation and Scope. This thesis proposes to treat probabilistic embeddings as latent states and update them via Kalman filtering. We test this idea in two settings: (i) a Gaussian embedding framework inspired by KG2E [13], which natively supports distributions, and (ii) a complex-valued interaction model (ComplEx) [21], extended to a probabilistic form. Together, these cover two very different scoring functions,

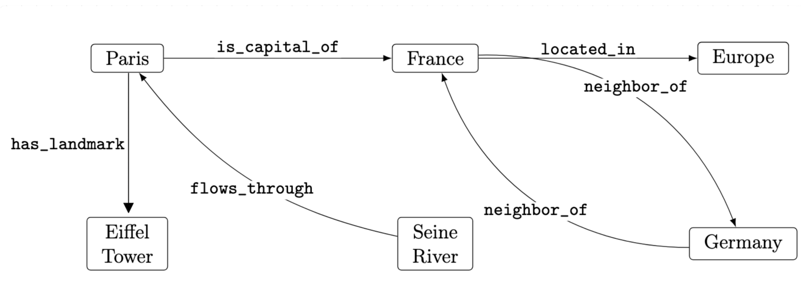

Figure 1.2: Conceptual mapping between Kalman Filter variables and probabilistic KGE embeddings, illustrated with a toy knowledge graph.

probabilistic Gaussian and complex-valued, and allow us to ask whether Kalman-based updates provide general benefits under fixed training budgets, such as limited computational resources.

Toy Knowledge Graph Example Before introducing formal notation, Figure 1.2 shows a small illustrative KG. Entities such as `Paris`, `France`, and `Europe` are connected by labeled relations. For instance, `(Paris, is_capital_of, France)` encodes that Paris is the capital of France, while `(France, located_in, Europe)` places France within Europe. Additional facts, such as the `Eiffel Tower` being a landmark of `Paris`, or the `Seine River` flowing through the city, illustrate how heterogeneous information can coexist within a single graph. This toy graph highlights two properties central to the work of this thesis:

1. Relations differ in structure: some are *asymmetric* (e.g., `is_capital_of`), while others are *symmetric* (e.g., `neighbor_of`). Effective embedding models must capture both.

2. Knowledge graphs evolve: new entities appear, relations change, and established facts are revised. Probabilistic embeddings equipped with Kalman updates can adapt gracefully to these shifts, refining their means and uncertainties as new information arrives.

The Kalman-based approach developed here equips embeddings with the ability to evolve as the knowledge graph changes, refining both their representations and associated uncertainties over time.

1.2 Related Work

Knowledge graph embeddings (KGEs) learn continuous vector representations for entities and relations to support tasks such as link prediction, reasoning, and knowledge graph completion. Early translational models established the foundation of this field. **TransE** [11] represents each relation as a translation in the embedding space between head and tail entities, while its extensions, **TransH** [22] and **TransR** [23], introduce relation-specific hyperplanes and projection matrices to better capture one-to-many and many-to-one patterns.

KG2E [13] extends this translational framework into a probabilistic one by representing entities and relations as multivariate Gaussian distributions, where the covariance encodes per-dimension uncertainty. This formulation improves robustness to sparsity and noise but remains fundamentally staticnew triples require retraining the entire model from scratch.

Modern toolkits such as **PyKEEN** [24] provide modular, reproducible pipelines for developing and evaluating KGE models. The PyKEEN implementation of KG2E serves as the current state of the art in probabilistic embedding and as the baseline for this thesis.

The **Kalman filter** [20, 25] offers a principled mechanism for recursive state estimation under uncertainty. By updating mean and covariance incrementally as new observations arrive, it allows models to adapt online without revisiting past data. Despite its success in control theory, robotics, and online neural learning, Kalman filtering remains largely unexplored in the context of KG embeddings.

Building on the uncertainty-aware foundation of KG2E, our approach applies Kalman filtering to perform online updates of entity and relation embeddings. This enables explicit uncertainty tracking, incremental adaptation to new facts, and support for nonstationary or drifting semantics through the inclusion of process noise. In doing so, it advances KGE methods toward scalable, dynamic, and uncertainty-aware learning.

Finally, other probabilistic methods, such as **UKGE** [26] and **BEUrRE** [14], associate uncertainty with individual triples rather than with entities or relations. While effective for confidence-aware reasoning, these approaches remain static and lack per-entity and per-relation uncertainty, making them complementary but orthogonal to our work.

1.3 Research Gap and Research Objectives

Based on the above discussion of related work, we see that no existing approach seamlessly integrates all three desired properties:

1. Per-entity and per-relation uncertainty modeling.
2. Efficient online updates without full retraining.
3. The ability to handle potentially nonstationary or drifting concepts in evolving KGs through adaptive uncertainty.

This gap directly motivates the research objectives of this thesis, which together define its design and evaluation goals. Our aim is to develop a knowledge graph embedding framework that:

1. **Represents uncertainty explicitly.** Each entity and relation is modeled as a probability distribution, defined by a mean vector and a diagonal covariance matrix.
2. **Supports online, incremental updates.** The model updates embeddings efficiently as new triples are processed, without retraining on the full dataset.
3. **Adapts to nonstationary knowledge.** Uncertainty updates allow the model to adjust when the semantics of entities or relations shift over time (concept drift).
4. **Maintains scalability.** The update procedure remains computationally efficient and applicable to large benchmark datasets.
5. **Improves or preserves predictive accuracy.** Under fixed computational budgets, the model achieves equal or better performance than static baselines on standard link prediction tasks.

In pursuit of these objectives, this thesis makes the following specific contributions.

1.4 Contributions

The specific contributions of this thesis are:

1. **KalmanKG2E Model:** An extension of the KG2E model [13] incorporating Kalman filtering [20] for online updates of Gaussian embedding means and covariances, designed for efficient and uncertainty-aware adaptation.

2. **KalmanComplEx Model:** A novel Kalman-filtered variant of the ComplEx model [21], demonstrating the applicability of the Kalman update mechanism to complex-valued embeddings by tracking uncertainty in real and imaginary components.

3. **Unified Online Update Framework:** A general methodology detailing how Kalman filtering can be integrated into KGE training loops, including the treatment of positive and negative triples and the role of process and measurement noise parameters.

4. **Comprehensive Empirical Evaluation:** Experiments conducted on six diverse benchmark datasets (FB15k-237, WN18RR, CoDEx-Medium, CoDEx-Large, DB100K, YAGO3-10), demonstrating performance improvements (e.g., up to 15% in Hits@10 on WN18RR for KG2E-Kalman) or favorable trade-offs compared to static baselines under fixed budgets, along with analyses of convergence and computational cost.

5. **Open-source Implementation:** A reproducible implementation based on PyKEEN [24], providing scripts for experiments, hyperparameter tuning, and analysis to facilitate future research.

Publication: This work has been published in the proceedings of the *14th International Conference on Complex Networks and Their Applications (COMPLEX NETWORKS 2025)*, Springer, under the title **"Tracking Uncertainty in Knowledge Graphs: A Kalman Filtering Approach"** (Alina Tkachenko, Alex Thomo, and Kevin Stanley, 2025).

Chapter 2

Background

2.1 Knowledge Graphs and Embeddings

A *knowledge graph* (KG) $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ is a directed multigraph where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ is the set of relation types, and $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is the set of observed triples (h, r, t). Each triple asserts that a relation $r \in \mathcal{R}$ holds between a head entity $h \in \mathcal{E}$ and a tail entity $t \in \mathcal{E}$. For example:

- $(\text{Paris}, \text{is\_capital\_of}, \text{France})$
- $(\text{Einstein}, \text{influenced}, \text{Bohr})$

Relations can be *symmetric* (e.g., married_to) or *asymmetric* (e.g., parent_of). Asymmetric relations are more general and dominate real-world KGs; we focus on this case throughout the thesis.

Knowledge graphs provide a structured and interpretable way to represent factual information, yet their symbolic nature poses serious challenges for learning and inference. Real-world KGs such as Wikidata, YAGO, or Freebase contain millions of entities and billions of triples, but are also *incomplete* (many true relations are missing) and *noisy* (some recorded facts are incorrect or uncertain). Symbolic representations, while semantically rich, lack the smoothness and generalization ability needed for machine learning methods that depend on numerical operations such as distance, similarity, and optimization.

Knowledge Graph Embeddings (KGEs) address these issues by mapping the discrete elements of a KG into a continuous vector space. Each entity $e \in \mathcal{E}$ is represented by a vector $\mathbf{e} \in \mathbb{R}^d$, and each relation $r \in \mathcal{R}$ by a vector $\mathbf{r} \in \mathbb{R}^d$, where d is a latent dimension typically much smaller than $|\mathcal{E}|$. The objective is to learn embeddings such that geometric operations in this latent space reflect the semantic and relational structure of the original graph. For example, if (h, r, t) holds in the graph, then the embedding of h combined with r should be close to that of t according to some scoring function $f(h, r, t)$.

Intuitively, embedding learning transforms symbolic reasoning into geometric reasoning: relations become translations, rotations, or interactions in a vector space; distances and angles capture semantic similarity; and neighborhoods reflect clusters of related entities. This representation enables a broad range of downstream applications, including link prediction, entity classification, relation extraction, and question answering, while also allowing efficient storage and computation.

The most common evaluation task for KGE models is *link prediction*, which tests the models ability to infer missing connections in the graph. Given a partial candidate triple such as $(h, r, ?)$ or $(?, r, t)$, the model ranks all possible entities and is evaluated by whether the correct entity receives a high score. This task captures the models ability to generalize beyond observed facts and to reason over incomplete dataa crucial capability for large-scale, evolving knowledge graphs.

2.1.1 Scoring Functions and Negative Sampling

Learning knowledge graph embeddings revolves around evaluating and comparing the plausibility of triples. To do so, a KGE model defines a *scoring function* $f(h, r, t) \in \mathbb{R}$ that assigns a real-valued score to each triple, representing how likely it is to be true. The guiding principle is simple: true triples (observed facts) should receive higher scores than false ones. This principle forms the foundation of nearly all KGE learning frameworks.

Training proceeds by contrasting positive triples $\mathcal{T}^+$, drawn from the observed KG, against *negative* (corrupted) triples $\mathcal{T}^-$. Negative samples are synthetic triples created by replacing either the head or the tail entity of a true triple with a randomly selected entity:

$$(h', r, t) \quad \text{or} \quad (h, r, t')$$

where h' or t' is drawn uniformly from $\mathcal{E}$. This process is known as the *stochastic Local Closed World Assumption* (sLCWA), since it assumes that unobserved triples are falseat least locally around the known facts.

The model is optimized using a *contrastive loss* that encourages separation between the scores of positive and negative triples. A common choice is the margin-based ranking loss:

$$\mathcal{L} = \sum_{(h,r,t)\in\mathcal{T}^+} \sum_{(h',r,t')\in\mathcal{T}^-} \max\big(0, \gamma + f(h', r, t') - f(h, r, t)\big)$$

where $\gamma > 0$ is the margin hyperparameter. Intuitively, the loss enforces that each positive triple should score higher than a negative triple by at least γ. Triples that already satisfy this margin contribute zero to the loss, focusing training on harder examples that violate the constraint. This ranking-based approach encourages relative ordering rather than absolute calibration, aligning naturally with the ranking-based evaluation metrics used in link prediction (e.g., Hits@10, MRR).

Negative sampling serves two critical roles: it provides the contrastive signal necessary for discriminative learning and it regularizes the embedding space. Without it, models would trivially assign high scores to all triples, collapsing the representation. The balance between positive and negative examples, and the quality of generated negatives, directly affects training stability and generalization.

2.1.2 Translational Models: TransE

Among early KGE methods, **TransE** [11] remains one of the most influential and conceptually intuitive. It models relations as simple *translations* in the embedding space, interpreting a relation vector $\mathbf{r}$ as an operation that moves the head embedding $\mathbf{h}$ close to the tail embedding $\mathbf{t}$. The scoring function is defined as the negative distance between the translated head and the tail:

$$f_{\text{TransE}}(h, r, t) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_p \tag{2.1}$$

where $\|\cdot\|_p$ is typically the L_1 or L_2 norm.

Training encourages valid triples to satisfy $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, while invalid ones do not. This geometric simplicity captures directional and hierarchical relations effectively, making TransE well-suited for *asymmetric* and *one-to-one* patterns—cases where each head corresponds to a single tail and vice versa (e.g., `(Paris, is_capital_of,` `France)` or `(Earth, has_moon, Moon)`). In such settings, the translation constraint

$$\mathbf{h} + \mathbf{r} \approx \mathbf{t}$$

aligns neatly with the graph structure, allowing the model to represent relations as clear, consistent vector offsets without interference from multiple possible targets.

While TransE provides an elegant geometric foundation, its simplicity limits expressiveness. It represents every entity and relation as a *deterministic point vector*, assuming complete certainty in all observed facts. Real-world knowledge graphs, however, are noisy, incomplete, and context dependent. Deterministic embeddings cannot express confidence or disagreement, leading to brittle generalization under uncertainty or data drift. This limitation motivates probabilistic approaches such as **KG2E** [13], which extend TransE by representing entities and relations as probability distributionsan idea explored in the next subsection.

2.1.3 From Points to Distributions: KG2E

Real-world knowledge graphs are far from perfect representations of truth. They are inherently uncertain: some facts are noisy, others are ambiguous, and most are only sparsely supported by evidence. Traditional point embeddings ignore this variability, assigning the same confidence to all facts regardless of how well they are substantiated. For example, the triple $(\text{Einstein}, \text{influenced}, \text{Bohr})$well established through historical recordsshould not be treated with the same certainty as a newly discovered or weakly documented relation. When all triples are represented deterministically, such distinctions vanish, leading to brittle and overconfident representations.

To address this limitation, He et al. [13] proposed **KG2E**, which models each entity e and relation r not as a single point in space, but as a *probability distribution*. Specifically, every entity and relation is represented as a multivariate Gaussian:

$$e \sim \mathcal{N}(\boldsymbol{\mu}_e, \boldsymbol{\Sigma}_e), \qquad r \sim \mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r),$$

where the mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ captures the central or expected position in the embedding space, and the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ quantifies uncertainty, encoding how dispersed or confident the model is about that representation. For computational tractability, the covariance is typically restricted to a diagonal form, $\boldsymbol{\Sigma}_e = \text{diag}(\boldsymbol{\sigma}_e^2)$, allowing independent uncertainty estimation along each dimension.

This probabilistic representation introduces a new degree of realism. Entities that appear in few triples or whose relationships are inconsistent will naturally learn larger variances, reflecting weaker evidence. Conversely, entities with abundant, coherent connectionssuch as "France" or "Earth"converge to tight, low-variance distributions, indicating strong certainty. In this way, the model internalizes data quality: it does not merely encode what it knows, but how confidently it knows it.

The translation principle of TransE extends naturally into this probabilistic framework. The composition of the head and relation, $(\mathbf{h}, \mathbf{r})$, is modeled as the sum of two independent random variables:

$$\mathcal{N}(\boldsymbol{\mu}_h + \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_h + \boldsymbol{\Sigma}_r),$$

which represents the predicted distribution for the tail entity. The scoring function then measures the similarity between this predicted distribution and the actual tail distribution $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$. Unlike deterministic models, where similarity is defined by Euclidean distance, KG2E compares entire probability distributionscapturing both positional alignment and uncertainty overlap.

The most common scoring function used in KG2E is the **asymmetric KullbackLeibler (KL) divergence:**

$$f_{\mathrm{KG2E}}^{\mathrm{KL}}(h, r, t) = -D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{\mu}_h + \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_h + \boldsymbol{\Sigma}_r) \parallel \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t))\,.$$

It measures how much information is lost when approximating the true tail distribution with the predicted one. By minimizing this divergence, the model encourages the predicted and true distributions to overlap, rewarding triples whose uncertainty regions intersect and naturally tolerating noise or sparsity in the graph.

In practice, KG2E is trained end-to-end with stochastic gradient methods such as Adam, using the same margin-based negative sampling as TransE. The **PyKEEN** framework [24] provides a robust implementation of KG2E, making it accessible for large-scale experimentation. However, despite its probabilistic nature, KG2E remains a *static* model. Once trained, both the means and covariances are fixed. Any new triple or entity requires retraining the entire model from scratcha process that is computationally costly and fundamentally inconsistent with the evolving nature of real-world knowledge graphs.

This limitation points directly to the motivation of the present work. If distributions provide a principled way to represent uncertainty, then the natural next step is to allow these distributions to evolve as new information arrives. Rather than relearning embeddings in batches, we seek a recursive mechanism to update means and covariances online, balancing prior belief and new evidence. This is precisely the role of the *Kalman filter*, introduced in Section 2.2 and applied in Chapter 3. It transforms the probabilistic embedding paradigm into a dynamic framework, one that updates continuously, adapts in real time, and learns as the knowledge graph evolves.

2.1.4 Complex-Valued Embeddings: ComplEx

Many real-world relations are *many-to-one* in nature, where multiple head entities share the same tail. For example, in the triples `(Paris, located_in, France)`, `(Lyon, located_in, France)`, and `(Marseille, located_in, France)`, all distinct cities map to the same country. Translational models such as TransE struggle with such many-to-one patterns. Because TransE represents each relation as a simple vector translation, every head entity connected by the same relation is pushed toward nearly the same region in the embedding space. In the example above, the vectors for `Paris`, `Lyon`, and `Marseille` all must satisfy $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, causing their embeddings to collapse around the vector for `France`. This geometric compression reduces the models capacity to distinguish between entities that play similar relational roles but differ semantically.

Addressing this limitation requires a richer representational space that can preserve directionality while maintaining distinct representations for entities participating in many-to-one relationships. This motivates the use of **ComplEx** [21], which embeds entities and relations in a complex vector space where real and imaginary components interact multiplicatively, allowing the model to encode directional and relational asymmetries more effectively.

Trouillon et al. [21] proposed **ComplEx**, a highly expressive embedding model that introduces *complex-valued representations* for entities and relations. Each entity e and relation r is represented as a vector in the complex space $\mathbb{C}^d$:

$$\mathbf{e} = \mathbf{e}_r + i\mathbf{e}_i, \quad \mathbf{r} = \mathbf{r}_r + i\mathbf{r}_i,$$

where $\mathbf{e}_r, \mathbf{r}_r$ denote the real parts, $\mathbf{e}_i, \mathbf{r}_i$ denote the imaginary parts, and $i = \sqrt{-1}$ is the imaginary unit. This extension effectively doubles the representational capacity of the model while keeping the same dimensionality, since real and imaginary parts are learned jointly.

The scoring function of ComplEx is defined as the real part of a trilinear Hermitian product:

$$f_{\text{ComplEx}}(h, r, t) = \text{Re}\big(\langle \mathbf{h}, \mathbf{r}, \overline{\mathbf{t}} \rangle\big) = \text{Re}\left(\sum_{k=1}^{d} h_k r_k \bar{t}_k\right),$$

where $\overline{\mathbf{t}} = \mathbf{t}_r - i\mathbf{t}_i$ is the complex conjugate of the tail embedding. This ensures that scores remain real-valued, while the use of complex conjugation allows the model to encode directional, non-reversible interactions between entities.

Expanding the Hermitian product reveals where this asymmetry originates:

$$f_{\text{ComplEx}}(h, r, t) = (\mathbf{h}_r \odot \mathbf{r}_r - \mathbf{h}_i \odot \mathbf{r}_i) \cdot \mathbf{t}_r + (\mathbf{h}_r \odot \mathbf{r}_i + \mathbf{h}_i \odot \mathbf{r}_r) \cdot \mathbf{t}_i,$$

where $\odot$ denotes elementwise multiplication. Because the tail vector appears in conjugated form, reversing the triple (t, r, h) changes the sign of the imaginary component, giving $f(h, r, t) \neq f(t, r, h)$. This asymmetry is crucial for modeling many-to-one or one-to-many relationspatterns where multiple heads can link to the same tail or vice versa. In TransE, all triples satisfying $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ collapse toward a single region, forcing distinct entities such as `Paris`, `Lyon`, and `Marseille` to cluster around `France`. ComplEx avoids this collapse by letting each head differ by its phase in the complex plane, so that multiple entities can still align with the same tail without losing their individuality.

Intuitively, while TransE represents relations as translationspushing all connected heads toward one tailComplEx represents them as rotations and phase shifts. Each relation defines an angular transformation that can rotate several distinct heads into approximate alignment with the same tail, yet keep their embeddings distinct through different phases. This geometric flexibility lets ComplEx represent many-to-one and one-to-many structures naturally, capturing relational multiplicity that TransE cannot express through linear translation alone.

Empirically, ComplEx achieves strong results on link prediction tasks with such relational multiplicity, including those with hierarchical or asymmetric structures. Its ability to preserve both directionality and diversity of entities, while maintaining computational efficiency, has made it a widely adopted baseline in modern KGE research.

However, despite its expressiveness, ComplEx remains fundamentally *deterministic*. All facts are treated with equal certainty, and embeddings are fixed after offline training. The model cannot quantify or propagate uncertainty, nor can it adapt incrementally to new information without retraining from scratch. Unlike TransE, which has a direct probabilistic extension through KG2E, ComplEx lacks a native distributional formulation. This absence of a probabilistic, uncertainty-aware mechanism forms a critical research gap.

In this thesis, we address this limitation by introducing **KalmanComplEx**, an online, uncertainty-aware adaptation of the complex-valued embedding framework. By coupling the expressive power of complex embeddings with the recursive, probabilistic updating of the Kalman filter, KalmanComplEx enables dynamic, distributional learning in the complex domain. It represents a step toward continuously updated, uncertainty-calibrated representations of asymmetric relational knowledge.

Taken together, **KG2E** and **ComplEx** resolve complementary aspects of the classical embedding problem. KG2E introduces uncertainty modeling through probabilistic distributions but remains confined to static, real-valued spaces. ComplEx, in contrast, captures asymmetric relational structure in a richer complex space but lacks any notion of uncertainty or adaptability. The goal of this thesis is to unify these strengths: to make probabilistic embeddings dynamic through Kalman filtering and to extend complex embeddings into an uncertainty-aware, continuously updating framework.

2.2 The Kalman Filter

The Kalman filter [20, 27] is a recursive, optimal algorithm for estimating the hidden state of a dynamic system from a sequence of noisy observations. It is the gold standard in fields like aerospace, robotics, and navigation. It maintains a probabilistic belief over the system's state, represented as a Gaussian distribution, and updates this belief optimally by combining prior knowledge with incoming evidence.

Unlike batch estimation methods (like standard KGE training) that require access to all data simultaneously, the Kalman filter operates *sequentially*. It updates its estimate each time a new observation arrives. This property makes it ideally suited for problems involving streaming data, online learning, and real-time adaptation, precisely the conditions under which knowledge graphs evolve.

At a conceptual level, the Kalman filter embodies a simple but powerful idea: *each new observation should adjust our belief about the world, but only in proportion to how much we trust it.* The filter does not replace old knowledge with new information; instead, it fuses them, balancing prior certainty against observational noise. The result is an adaptive, self-correcting estimator that learns continuously.

2.2.1 Intuition: Precision-Weighted Averaging

The essence of the Kalman filter can be illustrated through a simple one-dimensional example. Suppose we wish to estimate a scalar quantity x.

- Our **prior belief** (before a new measurement) is $\hat{x}_{\text{prior}}$, with a variance σ^2_{prior}.
- We then receive a new **measurement** z, which is noisy and has a measurement noise variance R.

Our goal is to produce an updated estimate $\hat{x}_{\text{update}}$ that best integrates these two pieces of information. The optimal update, derived from minimizing the posterior variance, takes the form of a correction:

$$\hat{x}_{\text{update}} = \hat{x}_{\text{prior}} + K(z - \hat{x}_{\text{prior}}),$$

where K is the **Kalman gain**. This gain K is the central component that balances the two sources of information:

$$K = \frac{\sigma^2_{\text{prior}}}{\sigma^2_{\text{prior}} + R}.$$

The gain determines how much weight to assign to the new measurement (the "innovation" $z - \hat{x}_{\text{prior}}$) relative to the prior belief:

- If the prior is very uncertain (large σ^2_{prior}), then $K \to 1$, and $\hat{x}_{\text{update}} \to z$. We mostly trust the new measurement.
- If the measurement is very noisy (large R), then $K \to 0$, and $\hat{x}_{\text{update}} \to \hat{x}_{\text{prior}}$. We mostly trust our prior.

The updated uncertainty (variance) also has a clean, closed-form solution:

$$\sigma^2_{\text{update}} = (1-K)\sigma^2_{\text{prior}} = \frac{\sigma^2_{\text{prior}} R}{\sigma^2_{\text{prior}} + R}.$$

This reveals the key insight: the new estimate is a *precision-weighted average*, where precision is the inverse of variance. In Bayesian terms, the Kalman filter performs a continuous posterior update under Gaussian assumptions, yielding a sharper (lower variance) belief at each iteration.

2.2.2 Prediction: Modeling Change Over Time

While the scalar example captures the update, most real systems also change between observations. The Kalman filter explicitly models this temporal evolution using a **linear dynamical system**. The state $\mathbf{x}$ at step n evolves from step $n-1$:

$$\mathbf{x}_n = F\mathbf{x}_{n-1} + \mathbf{w}_{n-1}, \quad \mathbf{w}_{n-1} \sim \mathcal{N}(\mathbf{0}, Q),$$

where $\mathbf{x}_n$ is the latent state vector, $F \in \mathbb{R}^{d\times d}$ is the state transition matrix (how the state evolves on its own), and Q is the **process noise covariance**. This Q term is crucial: it accounts for unmodeled dynamics, gradual drift, or general uncertainty about the system's evolution.

The **Prediction Step** propagates our belief (mean $\hat{\mathbf{x}}$ and covariance P) forward in time, *before* seeing the new measurement:

$$\hat{\mathbf{x}}_{n|n-1} = F\hat{\mathbf{x}}_{n-1|n-1}$$

$$P_{n|n-1} = FP_{n-1|n-1}F^\top + Q$$

The covariance $P_{n|n-1}$ always expands due to the addition of Q. This represents a natural increase in uncertainty as time passes without a new observation. This process noise is essentialit allows the model to remain "flexible" and prevents the covariance from collapsing to zero, which would stop the model from learning.

2.2.3 Update: Incorporating New Evidence

When a new observation $\mathbf{z}_n$ becomes available, the filter integrates it through the **measurement model**:

$$\mathbf{z}_n = H\mathbf{x}_n + \mathbf{v}_n, \quad \mathbf{v}_n \sim \mathcal{N}(\mathbf{0}, R),$$

where $H \in \mathbb{R}^{m \times d}$ is the observation matrix (mapping the latent state to the observation space) and R is the measurement noise covariance (how much we trust the sensor/observation).

The **Kalman gain** now generalizes to a matrix form that solves the precision-weighting problem in high dimensions:

$$K_n = P_{n|n-1}H^\top(HP_{n|n-1}H^\top + R)^{-1}.$$

The term $(HP_{n|n-1}H^\top + R)$ is the total uncertainty in the *observation space*. The gain K_n thus quantifies how much the "surprise" (the innovation $\mathbf{z}_n - H\hat{\mathbf{x}}_{n|n-1}$) should influence our belief about the latent state.

The **Update Step** then corrects the predicted state and shrinks its covariance:

$$\hat{\mathbf{x}}_{n|n} = \hat{\mathbf{x}}_{n|n-1} + K_n(\mathbf{z}_n - H\hat{\mathbf{x}}_{n|n-1})$$

$$P_{n|n} = (I - K_nH)P_{n|n-1}$$

The covariance $P_{n|n}$ contracts, reflecting that we have gained information and our belief about the state $\mathbf{x}_n$ is now more certain.

2.2.4 Geometric and Probabilistic View

Geometrically, the Kalman filter maintains an evolving Gaussian ellipsoid in the d-dimensional state space. The **Prediction** step (adding Q) inflates and warps this ellipsoid. The **Update** step "intersects" this prior ellipsoid with the new information from the measurement (another, often larger, ellipsoid), resulting in a new, smaller ellipsoid that represents the updated posterior belief.

Probabilistically, this corresponds to Bayesian conditioning. The prior $p(\mathbf{x}_n|\mathbf{z}_{1:n-1})$ is multiplied by the likelihood $p(\mathbf{z}_n|\mathbf{x}_n)$ to produce the posterior $p(\mathbf{x}_n|\mathbf{z}_{1:n})$. Due to the conjugacy of linear-Gaussian models, this posterior is also a Gaussian, allowing the process to repeat indefinitely. This dual interpretation provides a perfect bridge to distributional KGEs, where means are entity locations and covariances encode their uncertainty.

2.2.5 Key Properties

The Kalman filter is defined by several properties that make it uniquely suited to our problem:

- **Closed-form:** All updates are exact analytic solutions, not approximations. They are computed via matrix operations, requiring no iterative gradient descent, sampling, or numerical optimization. This is computationally efficient.

- **Recursive:** The filter is "memoryless." It maintains only the current state estimate ($\hat{\mathbf{x}}_{n|n}$) and its covariance ($P_{n|n}$). All information from past observations is compressed into this belief. It does not need to store or revisit past data, making it ideal for unbounded data streams.

- **Optimal:** Under the linear-Gaussian assumptions, the Kalman filter is provably optimal. It is the *minimum mean-squared error (MMSE)* estimator. No other linear or non-linear estimator can produce a more accurate estimate on average.

- **Adaptive:** The Kalman gain K_n acts as a dynamically adjusting, "principled" learning rate. When the model is uncertain (large P) or the measurement is reliable (small R), the gain is large, and the model learns quickly. When the model is confident (small P) or the measurement is noisy (large R), the gain is small, and the model resists change.

- **Interpretable:** Every component has a clear probabilistic meaning. P is our model's uncertainty. Q is our assumption about how much the world "drifts." R is our trust in the data.

In the next chapter, we will leverage these properties, reinterpreting knowledge graph embeddings as latent states within this framework. Triples will act as noisy observations, embedding distributions as our prior beliefs, and the Kalman filter will become the core mechanism for continuous, uncertainty-aware knowledge updates as graphs evolve over time.

Chapter 3

From Static to Adaptive: A Kalman Filtering Approach to Knowledge Graph Embedding

This chapter introduces a unified, uncertainty-aware framework for *online knowledge graph embedding* using Kalman filtering. We treat entity and relation embeddings as latent states in a linear dynamical system, with means representing central estimates and diagonal covariances capturing per-dimension uncertainty. As triples arrive sequentially, the filter performs closed-form prediction and update steps to refine embeddings incrementally, eliminating the need for full retraining. We first present **KalmanKG2E**, which extends the Gaussian-based KG2E model with contrastive Kalman updates for positive and negative triples. We then generalize the approach to **KalmanComplEx**, applying the same filtering mechanism to real and imaginary components of complex-valued embeddings. In both cases, process noise enables adaptation to concept drift, measurement noise modulates trust in observed facts, and the Kalman gain dynamically balances prior belief with new evidence. The resulting models support scalable, adaptive, and uncertainty-aware link prediction in evolving knowledge graphs.

3.1 Kalman Filter for KG Embeddings (KalmanKG2E)

The first proposed model, **KalmanKG2E**, applies the Kalman filter to the problem of learning knowledge graph embeddings in an online, uncertainty-aware setting. The central idea is to treat each entity and relation as a latent state that evolves over time as new triples are observed. At any given step, the mean vector represents the current estimate of the embedding, while the covariance quantifies uncertainty about that estimate. When a new triple arrives, the Kalman filter combines prior knowledge (the previous mean and covariance) with the new observation (the triples relational constraint) to produce a refined, probabilistically grounded update. This process provides a mathematically principled mechanism for incremental learning, replacing gradient-based batch optimization with recursive Bayesian estimation.

In this formulation, both entities and relations are equipped with dynamic uncertainty estimates that evolve with each update. This allows the model to express different confidence levels across the graph: entities seen frequently have low-variance embeddings, while sparsely observed or newly introduced entities retain higher uncertainty. By maintaining these uncertainty estimates, the model can moderate how strongly each new triple influences the embeddings, preventing overconfident updates when data are scarce or noisy. We also incorporate contrastive learning by using both positive and negative triples, allowing the Kalman updates to push compatible entities closer together and incompatible ones apart in a unified probabilistic framework.

Clarification on Training Dynamics: It is important to clarify that the training process in this framework differs fundamentally from traditional Knowledge Graph Embedding (KGE) training. Unlike standard models that optimize a global loss function via stochastic gradient descent (SGD), our framework performs **recursive Kalman updates**.

In this context, the scoring function (e.g., the translational distance in KG2E) serves two distinct roles:

1. **Constraint Definition (Training):** The scoring function defines the *Measurement Model*. It establishes the geometric constraint (e.g., $h + r \approx t$) that the Kalman filter uses to calculate the "innovation" or error residual. The filter then minimizes this error locally for each triple.

2. **Evaluation (Testing):** During link prediction, the scoring function is used to rank candidate entities and compute metrics such as MRR and Hits@10, exactly as it is used in static baselines.

Consequently, the scoring function itself is not "optimized" iteratively; rather, it defines the observation space for the filter.

3.1.1 Initialization

Each entity e_i and relation r_j in the knowledge graph is initialized with:

- an initial embedding vector, which may be sampled from a small Gaussian distribution or derived from pre-trained embeddings if available, and
- a diagonal covariance matrix $P = \sigma^2 I$, where I is the identity matrix and σ^2 represents the initial uncertainty.

This initialization sets the prior belief for each latent state. A larger σ^2 indicates lower initial confidence, allowing the model to adjust quickly in early updates. Conversely, smaller variances slow adaptation, favoring stability over flexibility. These priors therefore control how aggressively the model learns in its first iterations and can be tuned to reflect domain knowledge about the expected variability of entities and relations.

3.1.2 Kalman Filter Update for Each Triple

For each incoming triple (h, r, t), the Kalman filter performs a two-stage update for the corresponding embeddings of the head h, relation r, and tail t. The first stage, *prediction*, propagates prior knowledge forward in time, estimating what the embeddings are expected to be before observing the new triple. The second stage, *update*, corrects these estimates based on the information provided by the triple itself. Together, these steps constitute a recursive belief-update process that continuously refines both the embeddings and their associated uncertainty.

Prediction Step

In the prediction step, each embedding vector is projected from its previous state to the next:

$$\hat{x}_{n,\text{prior}} = F\hat{x}_{n-1,\text{update}}$$

where F is the state transition matrix. Since there is no independent temporal dynamic driving the embeddings in our formulation, F is set to the identity matrix. Thus, the predicted state simply inherits the most recent updated state:

$$\hat{x}_{n,\text{prior}} = \hat{x}_{n-1,\text{update}}.$$

The associated covariance is also propagated forward, with the addition of process noise to account for possible drift:

$$P_{n,\text{prior}} = P_{n-1,\text{update}} + Q,$$

where Q is a diagonal matrix whose small positive entries represent process noise.

Intuitively, this step maintains continuity in the embeddings while acknowledging the possibility of gradual change. The use of an identity transition matrix reflects the assumption that embeddings do not evolve autonomously between updatesonly when informed by new relational evidence. However, adding Q reintroduces a small degree of uncertainty into each dimension, ensuring that the model remains flexible enough to adapt when the next observation arrives. This prevents the covariance from collapsing to zero and allows the embeddings to retain the capacity for future movement, effectively keeping the system "open to learning."

Measurement Model for Positive and Negative Triples

For each incoming triple (h, r, t), the Kalman filter requires a measurement model that expresses how the observed data (triples) relate to the latent states (embeddings). This measurement encodes the relational structure implied by the triple and serves as the observation against which the current embeddings are compared. We define two complementary cases: positive triples, representing true facts, and negative triples, representing corrupted or false ones.

- **Positive Triple:** For a valid triple (h, r, t), the measurements correspond to the expected geometric relationships between the embeddings:

  $$z_{h,\text{measure}} = t - r, \quad z_{t,\text{measure}} = h + r, \quad z_{r,\text{measure}} = t - h$$

  and the measurement noise covariance R_{positive} is applied.

  These equations align directly with the translational principle of TransE, ensuring that the embeddings are updated toward satisfying $h + r \approx t$. The covariance R_{positive} controls the degree of trust placed in each positive observation, allowing the model to remain flexible while still prioritizing consistency with verified triples.

- **Negative Triple:** For a corrupted or negative triple (e.g., (h, r, t') where $t' \neq$

t), the goal is to discourage the model from aligning $h + r$ with t'. To achieve this, we define a shifted measurement:

$$z_{t',\text{measure}} = t' + \text{offset},$$

where *offset* is a small random vector drawn from a normal distribution with a modest standard deviation relative to the embedding scale. This offset introduces controlled separation between the predicted and corrupted embeddings, providing contrastive information without destabilizing the embedding space.

The measurement noise for negative triples is set higher:

$$R_{\text{negative}} = \alpha \cdot R_{\text{positive}}, \quad \alpha > 1,$$

reflecting lower confidence in these artificial examples. The larger covariance downweights the influence of each negative triple, allowing it to act as a "soft constraint"pushing the model away from false associations without forcing sharp adjustments to h or r.

This asymmetric treatment of positive and negative triples mirrors their epistemic roles. Positive triples provide direct evidence about valid relationships and should therefore produce confident updates. Negative triples, by contrast, serve as mild repulsive signals that shape the geometry of the embedding space, ensuring separation between true and false relations. Combining both within the same Kalman filtering framework, the model learns not only to align compatible entities but also to maintain meaningful margins between incompatible ones. This dual mechanism is key to stable and uncertainty-aware contrastive learning in evolving knowledge graphs.

Kalman Gain and Update Equations

Once the prediction and measurement steps have been defined, the Kalman filter fuses them through a set of recursive update equations that refine both the mean and covariance of each embedding. For every entity and relation involved in a triple, the following three updates are applied.

Kalman Gain:

$$K_n = P_{n,\text{prior}}(P_{n,\text{prior}} + R)^{-1}$$

where $R = R_{\text{positive}}$ for positive triples and $R = R_{\text{negative}}$ for negative triples.

The Kalman gain K_n determines how much the model should trust the new measurement relative to its prior belief. If the prior uncertainty $P_{n,\text{prior}}$ is large, the gain increases, allowing the measurement to have greater influence. Conversely, if the measurement noise R is large (as in the case of negative triples), the gain decreases, preserving confidence in the prior estimate.

Dynamic Nature of the Gain. A key property of this update mechanism is that the Kalman gain is not constant, even when the noise matrices Q and R are fixed. The gain is computed directly from the evolving error covariance:

$$K_n = P_{n,\text{prior}}(P_{n,\text{prior}} + R)^{-1}.$$

Because the covariance P is updated after every triple, the gain adapts automatically over time:

- **Early in training:** Uncertainty is high, leading to a large gain and substantial embedding updates.
- **Later in training:** As uncertainty decreases, the gain shrinks, stabilizing the embeddings and reducing sensitivity to noise.

This behavior acts as an uncertainty-driven learning rate that decays naturally as evidence accumulates.

State Update:

$$\begin{aligned}
\hat{x}_{n,h,\text{update}} &= \hat{x}_{n,h,\text{prior}} + K_{n,h}(z_{h,\text{measure}} - \hat{x}_{n,h,\text{prior}}) \\
\hat{x}_{n,t,\text{update}} &= \hat{x}_{n,t,\text{prior}} + K_{n,t}(z_{t,\text{measure}} - \hat{x}_{n,t,\text{prior}}) \\
\hat{x}_{n,r,\text{update}} &= \hat{x}_{n,r,\text{prior}} + K_{n,r}(z_{r,\text{measure}} - \hat{x}_{n,r,\text{prior}})
\end{aligned}$$

Here, the residual term $(z_{\text{measure}} - \hat{x}_{\text{prior}})$ represents the discrepancy between what the model currently believes and what the triple implies. The gain K_n scales this residual, producing an embedding shift proportional to both the reliability of the new evidence and the models confidence in its prior. Since the observation matrix H is assumed to be the identity ($H = I$), the measurement directly informs the embeddings in their native vector space, requiring no transformation.

Covariance Update:

$$
\begin{aligned}
P_{n,h,\text{update}} &= (I - K_{n,h})P_{n,h,\text{prior}} \\
P_{n,t,\text{update}} &= (I - K_{n,t})P_{n,t,\text{prior}} \\
P_{n,r,\text{update}} &= (I - K_{n,r})P_{n,r,\text{prior}}
\end{aligned}
$$

The covariance update reflects the models increased confidence after processing the new triple. Dimensions in which the measurement provided strong evidence see their uncertainty reduced, while uncertain dimensions retain higher variance. By applying the correction term $(I - K_n)$, the filter ensures that embeddings become progressively more stable in well-learned regions while remaining adaptable where uncertainty persists. This self-regulating uncertainty mechanism allows the model to retain flexibility without overfitting.

Summary. Together, these equations allow the Kalman filter to act as a probabilistic optimizer for knowledge graph embeddings. Positive triples drive the embeddings toward semantically consistent configurations, while negative tripleswith their higher measurement noiseprovide soft repulsive forces that maintain contrastive structure. Unlike fixed learning rates or static update rules, the Kalman gain evolves dynamically with uncertainty, giving the model a form of *learned adaptivity*. Each triple thus contributes not just new information but also a refined sense of how confident the model should be in what it has learned.

3.1.3 Example

Assume a Knowledge Graph with a positive triple $(h, r, t) = (\text{Paris}, \text{capital\_of}, \text{France})$ and a negative triple $(h, r, t') = (\text{Paris}, \text{capital\_of}, \text{Germany})$. We initialize the embeddings for each entity and relation randomly and define their initial covariance matrices as follows.

- **Embeddings:**

$$
\hat{x}_{\text{Paris}} = \begin{bmatrix} 0.5 \\ 0.3 \end{bmatrix}, \quad \hat{x}_{\text{capital\_of}} = \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix},
$$

$$
\hat{x}_{\text{France}} = \begin{bmatrix} 1.0 \\ 0.8 \end{bmatrix}, \quad \hat{x}_{\text{Germany}} = \begin{bmatrix} 0.7 \\ 0.9 \end{bmatrix}
$$

- **Initial Covariance Matrices:** Each embedding is associated with an initial diagonal covariance matrix to represent uncertainty in each dimension.

$$P_{\text{Paris}} = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}, \quad P_{\text{capital\_of}} = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix},$$
$$P_{\text{France}} = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}, \quad P_{\text{Germany}} = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}$$

Kalman Filter Update for Positive Triple

For positive triple $(\text{Paris}, \text{capital\_of}, \text{France})$, we compute the measurements as:

$$z_{\text{Paris, measure}} = \hat{x}_{\text{France}} - \hat{x}_{\text{capital\_of}} = \begin{bmatrix} 0.8 \\ 0.7 \end{bmatrix},$$

$$z_{\text{France, measure}} = \hat{x}_{\text{Paris}} + \hat{x}_{\text{capital\_of}} = \begin{bmatrix} 0.7 \\ 0.4 \end{bmatrix}$$

and use a measurement noise covariance

$$R_{\text{positive}} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

for both Paris and France.

Kalman Gain Calculation

For Paris, the Kalman gain K_{Paris} is calculated as:

$$K_{\text{Paris}} = P_{\text{Paris, prior}}(P_{\text{Paris, prior}} + R_{\text{positive}})^{-1}$$

$$= \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix} \begin{bmatrix} 0.15 & 0 \\ 0 & 0.15 \end{bmatrix}^{-1} = \begin{bmatrix} 0.333 & 0 \\ 0 & 0.333 \end{bmatrix}$$

State Update for Paris

The updated embedding for Paris is:

$$\hat{x}_{\text{Paris, update}} = \hat{x}_{\text{Paris, prior}} + K_{\text{Paris}}(z_{\text{Paris, measure}} - \hat{x}_{\text{Paris, prior}}) =$$

$$\begin{bmatrix}0.5\\0.3\end{bmatrix}+\begin{bmatrix}0.333&0\\0&0.333\end{bmatrix}\left(\begin{bmatrix}0.8\\0.7\end{bmatrix}-\begin{bmatrix}0.5\\0.3\end{bmatrix}\right)=\begin{bmatrix}0.6\\0.433\end{bmatrix}$$

Covariance Update for Paris

The updated covariance for Paris is:

$$P_{\text{Paris, update}}=(I-K_{\text{Paris}})P_{\text{Paris, prior}}=$$

$$\left(\begin{bmatrix}1&0\\0&1\end{bmatrix}-\begin{bmatrix}0.333&0\\0&0.333\end{bmatrix}\right)\begin{bmatrix}0.05&0\\0&0.05\end{bmatrix}=\begin{bmatrix}0.033&0\\0&0.033\end{bmatrix}$$

This completes the Kalman filter update for Paris. A similar process would apply for France and capital_of.

Kalman Filter Update for Negative Triple

For the negative triple $(\text{Paris}, \text{capital\_of}, \text{Germany})$, we aim to discourage the alignment $h+r\approx t'$. To achieve this, we introduce an offset to the measurement for Germany, pushing it away from the predicted position of Paris + capital_of. We set:

$$z_{\text{Germany, measure}}=\hat{x}_{\text{Germany}}+\text{offset}=\begin{bmatrix}0.7\\0.9\end{bmatrix}+\begin{bmatrix}0.1\\-0.1\end{bmatrix}=\begin{bmatrix}0.8\\0.8\end{bmatrix}$$

where the offset is chosen to shift Germanys embedding slightly in a different direction. For negative triples, we use a higher measurement noise covariance $R_{\text{negative}}=\alpha\cdot R_{\text{positive}}=\begin{bmatrix}0.3&0\\0&0.3\end{bmatrix}$, with $\alpha=3$.

Kalman Gain Calculation for Germany

For Germany, the Kalman gain K_{Germany} is calculated as:

$$K_{\text{Germany}}=P_{\text{Germany, prior}}(P_{\text{Germany, prior}}+R_{\text{negative}})^{-1}=$$

$$\begin{bmatrix}0.05&0\\0&0.05\end{bmatrix}\begin{bmatrix}0.35&0\\0&0.35\end{bmatrix}^{-1}=\begin{bmatrix}0.143&0\\0&0.143\end{bmatrix}$$

State Update for Germany

The updated embedding for Germany is:

$$\hat{x}_{\text{Germany, update}} =$$

$$\hat{x}_{\text{Germany, prior}} + K_{\text{Germany}}(z_{\text{Germany, measure}} - \hat{x}_{\text{Germany, prior}}) =$$
$$\begin{bmatrix} 0.7 \\ 0.9 \end{bmatrix} + \begin{bmatrix} 0.143 & 0 \\ 0 & 0.143 \end{bmatrix} \left(\begin{bmatrix} 0.8 \\ 0.8 \end{bmatrix} - \begin{bmatrix} 0.7 \\ 0.9 \end{bmatrix} \right) = \begin{bmatrix} 0.714 \\ 0.886 \end{bmatrix}$$

Covariance Update for Germany

The updated covariance for Germany is:

$$P_{\text{Germany, update}} = (I - K_{\text{Germany}})P_{\text{Germany, prior}} =$$
$$\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.143 & 0 \\ 0 & 0.143 \end{bmatrix} \right) \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix} = \begin{bmatrix} 0.043 & 0 \\ 0 & 0.043 \end{bmatrix}$$

Takeaway

This toy example highlights how the Kalman-based update behaves very differently for positive and negative evidence. For the positive triple $(\text{Paris}, \text{capital\_of}, \text{France})$, the Kalman gain is relatively large and the embedding of Paris moves noticeably toward the measurement $z_{\text{Paris, measure}}$, while its covariance shrinks from 0.05 to 0.033 in each dimension. This reflects that the model both *adjusts its belief* and becomes *more confident* when a trusted, positive fact is observed.

For the negative triple $(\text{Paris}, \text{capital\_of}, \text{Germany})$, the larger measurement noise R_{negative} yields a smaller gain for Germany, producing only a mild shift in its mean (from $(0.7, 0.9)$ to $(0.714, 0.886)$) and a much smaller reduction in covariance (from 0.05 to 0.043). Negative evidence thus acts as a soft, contrastive signal: it gently pushes embeddings away from implausible configurations without overriding strong prior beliefs.

3.2 Kalman Filter for ComplEx Embeddings (KalmanComplEx)

The second component of this framework extends Kalman filtering to the complex-valued embedding model **ComplEx**. ComplEx represents entities and relations as vectors with real and imaginary parts, allowing it to capture asymmetric and many-to-one relational patterns that simple translational models cannot. While standard ComplEx embeddings are deterministic, we make them *probabilistic* by assigning each real and imaginary component its own Gaussian distribution, characterized by a mean and diagonal covariance. Applying the Kalman filter separately to these components enables continuous, uncertainty-aware updates of complex-valued embeddings as new triples arrive. This way, **KalmanComplEx** preserves the expressive relational structure of ComplEx while adding a principled mechanism for tracking and adapting uncertainty over time.

3.2.1 Probabilistic Representation

Each entity or relation is represented by both a mean and a variance for its real and imaginary parts:

$$(\underbrace{h_r^{\mu}}_{\text{real mean}}, \underbrace{h_i^{\mu}}_{\text{imag mean}}, \underbrace{h_r^{\sigma}}_{\text{real var}}, \underbrace{h_i^{\sigma}}_{\text{imag var}}),$$

so that every embedding now maintains a distributional characterization in the complex domain. This design effectively yields four learned vectors per entity or relation, allowing the model to express directional asymmetry while also quantifying uncertainty within each subspace.

3.2.2 Initialization

As in KalmanKG2E, each entity e_i and relation r_j begins with:

- an initial mean vector (split into real and imaginary parts), and
- a diagonal covariance vector for each part, i.e., $\sigma^2 I$ for the real and $\sigma^2 I$ for the imaginary components.

This initialization yields an effective embedding of dimension $4 \times d$: $\left(h_r^{\mu}, h_i^{\mu}, h_r^{\sigma}, h_i^{\sigma}\right)$.

3.2.3 Kalman Filtering for Real and Imaginary Components

For each triple (h, r, t), the filter updates the real and imaginary parts independently:

$$(h_r^\mu, h_r^\sigma), \quad (h_i^\mu, h_i^\sigma), \quad (r_r^\mu, r_r^\sigma), \quad (r_i^\mu, r_i^\sigma), \quad (t_r^\mu, t_r^\sigma), \quad (t_i^\mu, t_i^\sigma).$$

Each update follows the same prediction and correction logic introduced for KalmanKG2E, now applied dimension-wise to both the real and imaginary subspaces.

Prediction Step

$$\hat{x}_{n,\text{prior}} = \hat{x}_{n-1,\text{update}}, \qquad P_{n,\text{prior}} = P_{n-1,\text{update}} + Q.$$

Here, x may correspond to any mean component $(h_r^\mu, h_i^\mu, r_r^\mu, \ldots)$, and P denotes its diagonal covariance. The process noise Q injects mild stochasticity into each dimension, preserving flexibility as the embeddings evolve through successive updates.

Measurement Model for Complex Embeddings

The measurement logic mirrors that of KalmanKG2E, but applied independently to real and imaginary components. For a positive triple (h, r, t),

$$z_{h_r,\text{measure}} = t_r^\mu - r_r^\mu, \quad z_{h_i,\text{measure}} = t_i^\mu - r_i^\mu,$$

and similarly for the corresponding relation and tail components. Negative triples are treated analogously, using higher measurement noise and optional offsets to create contrastive separation between incorrect and correct associations.

Although each embedding now has both real and imaginary parts, the fundamental idea remains unchanged: the measurement vectors z guide each component toward or away from the relationships implied by the triples, with noise controlling the trust level of these corrections.

Kalman Gain and State Update

The Kalman gain and update equations remain identical in form:

$$K_n = P_{n,\text{prior}}\big(P_{n,\text{prior}} + R\big)^{-1}, \tag{3.1}$$

$$\hat{x}_{n,\text{update}} = \hat{x}_{n,\text{prior}} + K_n\left(z - \hat{x}_{n,\text{prior}}\right), \tag{3.2}$$

$$P_{n,\text{update}} = (I - K_n)\, P_{n,\text{prior}}. \tag{3.3}$$

All computations are performed element-wise, ensuring that each real and imaginary dimension adjusts independently according to its uncertainty. This per-dimension scaling gives KalmanComplEx the ability to differentially weight information across embedding components, something not achievable in standard gradient-based updates.

ComplEx Scoring with Updated Means

After the updates, the real and imaginary means of each embedding are used to compute the standard ComplEx scoring function:

$$\operatorname{Re}\Big((h_r^\mu + i\, h_i^\mu) \odot (r_r^\mu + i\, r_i^\mu) \odot \overline{(t_r^\mu + i\, t_i^\mu)}\Big),$$

where $\bar{\cdot}$ denotes complex conjugation. This ensures full compatibility with the original ComplEx framework while benefiting from probabilistic adaptivity.

Variance Penalty (Optional). To regularize overly uncertain embeddings, a KG2E-style variance penalty can be applied by aggregating variances across all components $\big(h_r^\sigma, h_i^\sigma, r_r^\sigma, \dots\big)$. Large variances, which indicate low confidence, can down-weight the ComplEx score using a simple scaling factor such as $\big(\sum \sigma\big)^{1/2}$, or equivalently be interpreted under a negative log-likelihood formulation. This optional adjustment allows the uncertainty modeled by the filter to directly influence prediction confidence.

In summary, KalmanComplEx fuses the expressive geometry of complex embeddings with the dynamic, uncertainty-aware refinement of Kalman filtering. Maintaining per-dimension covariances for both real and imaginary parts, the model can continuously adjust to new evidence while preserving the asymmetric relational structure that makes ComplEx effective in the first place. The result is a generalizable and flexible mechanism for evolving probabilistic embeddings in complex-valued spaces.

3.3 Summary of the Unified Framework

The two models introduced in this chapter, **KalmanKG2E** and **KalmanComplEx**, demonstrate how Kalman filtering provides a general, Gaussian-state framework for learning uncertainty-aware knowledge graph embeddings. While the models differ in representation geometry (real-valued translational vs. complex-valued semantic matching), both maintain Gaussian beliefs over embeddings and apply the same recursive update mechanism.

1. **Embeddings as latent Gaussian states.** Each entity and relation is represented by a mean vector and a diagonal covariance matrix, jointly capturing both its estimated position in the embedding space and the models uncertainty about it.

2. **Sequential Bayesian updates.** Each new triple acts as an observation that partially reveals these latent states. The Kalman filter fuses prior beliefs with this new evidence through closed-form updates, adjusting means and covariances according to relative uncertainty.

3. **Dynamic uncertainty calibration.** Process noise Q allows embeddings to drift as knowledge evolves, while measurement noise R controls the degree of trust in each observation. Positive triples are treated as low-noise, high-confidence evidence, whereas negative triples serve as high-noise, contrastive feedback that shapes relational boundaries.

In **KalmanKG2E**, these Gaussian-state updates operate within a translational geometry, directly extending the probabilistic framework of KG2E to an online, continuously adaptive setting. In **KalmanComplEx**, the same filtering principles are applied to the real and imaginary parts of complex embeddings, endowing the deterministic ComplEx model with dynamic uncertainty tracking and online adaptability.

These models unify two major KGE paradigms under a single probabilistic interpretation. They show that Kalman filtering is not limited to a particular embedding family but serves as a general principle for evolving, uncertainty-calibrated representations of knowledge.

Chapter 4

Experimental Setup

4.1 Introduction and Experimental Objectives

The preceding chapter introduced two online, uncertainty-aware knowledge graph embedding models: **KalmanKG2E** (Section 3.1) and **KalmanComplEx** (Section 3.2). This chapter details the empirical setup used to evaluate these models and assess their effectiveness relative to their static, batch-trained baselines. The overarching aim is to determine whether the integration of Kalman filteringthrough per-triple probabilistic updatesenhances predictive performance, adaptability, and generalization in knowledge graph embedding.

The experiments are designed around two central research questions derived from the thesis objectives:

1. **Efficacy.** Does the proposed online, per-triple update mechanismbased on Kalman filtering and explicit uncertainty trackingachieve higher predictive accuracy than traditional static models such as KG2E and ComplEx, which rely on batch-based optimization?

2. **Generalizability.** Does Kalman filtering constitute a general-purpose framework for online embedding learning, or is its benefit specific to certain model architectures? This is tested by applying the same update formulation to two distinct embedding paradigms: the Gaussian-translational family (**KalmanKG2E**) and the complex-valued semantic matching family (**KalmanComplEx**).

The evaluation task for all experiments is **link prediction**, the standard benchmark for knowledge graph embedding. Link prediction tests a models ability to infer missing facts by predicting the correct entity in an incomplete triple, such as $(h, r, ?)$ for tail prediction or $(?, r, t)$ for head prediction. For each query, the model computes a plausibility score for all candidate entities, ranks them, and measures how highly the true entity appears in this ranking.

This protocol provides a direct measure of how well a model captures the graphs semantic structure. For every test triple (h, r, t), the model ranks the true tail entity t against all other entities $e \in \mathcal{E}$ for the query $(h, r, ?)$, and similarly ranks the true head h for $(?, r, t)$. A model that has effectively learned relational patterns and uncertainty will consistently assign higher scores to the correct entities, even in the presence of noise, sparsity, or evolving graph structure.

4.2 Evaluation Metrics

To evaluate model performance on the link prediction task, we employ two widely used ranking-based metrics: **Mean Reciprocal Rank (MRR)** and **Hits@10**. Together, these measures quantify both the precision and the coverage of a models predictions, offering a complementary view of its ranking quality. They are the de facto standards in knowledge graph embedding research, enabling direct comparison with prior work.

4.2.1 Mean Reciprocal Rank (MRR)

The *Mean Reciprocal Rank* measures the average inverse rank of the correct entity across all test queries. It is defined as:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \tag{4.1}$$

where $|Q|$ is the total number of evaluated queries (both $(h, r, ?)$ and $(?, r, t)$), and rank_i denotes the rank position of the correct entity for the i-th query.

MRR emphasizes precision at the very top of the ranked list. A perfect prediction yields a score of 1.0, while a rank of 2 or 10 yields 0.5 and 0.1, respectively. Because it averages reciprocal ranks, MRR penalizes lower-ranked correct answers more strongly than metrics based on top-k inclusion alone. In the context of this thesis, MRR reflects how effectively the proposed Kalman-based models refine uncertainty to produce sharper, more confident rankings of correct entities.

4.2.2 Hits@10

The *Hits@10* metric captures the proportion of queries in which the correct entity appears among the top ten ranked predictions:

$$\text{Hits@10} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \mathbb{I}(\text{rank}_i \leq 10) \tag{4.2}$$

where $\mathbb{I}(\cdot)$ is the indicator function, equal to 1 when the condition is true and 0 otherwise. Hits@10 measures retrieval effectivenessanalogous to recall in information retrievalanswering the practical question: *How often does the model place the correct answer within the first page of results?*

While MRR focuses on ranking sharpness, Hits@10 highlights overall recall. Evaluating both allows us to understand not only whether the model identifies the right entity, but also how decisively it ranks that entity above competing candidates. For uncertainty-aware models such as KalmanKG2E and KalmanComplEx, improvements in these metrics indicate that probabilistic updates and dynamic covariance adjustments lead to more accurate and confident inference across diverse relational contexts.

4.3 Benchmark Datasets and Baselines

This section describes the datasets and baseline models used to evaluate the proposed Kalman-based knowledge graph embeddings. The experiments span six widely used benchmark datasets that vary substantially in domain, scale, and structural complexity, ensuring a comprehensive assessment of both performance and generalizability. Dataset statistics are summarized in Table 4.1.

Table 4.1: Statistics of benchmark datasets used for evaluation.

Dataset	**Entities**	**Relations**	**Train**	**Test**
WN18RR	40,943	11	86,835	3,134
FB15k237	14,541	237	272,115	20,466
CoDExMedium	17,050	51	185,584	10,311
DB100K	99,604	470	597,572	50,000
CoDExLarge	77,951	69	551,193	30,622
YAGO310	123,182	37	1,073,112	4,076

4.3.1 Dataset Overview

The selected datasets cover a broad range of knowledge domains, from linguistic and encyclopedic sources to structured factual graphs:

- **WN18RR:** A subset of *WordNet*, a lexical database of English. It features strong hierarchical and symmetric relations but only 11 relation types, emphasizing reasoning over structural patterns (such as reverses and transitivity) rather than relation diversity.
- **FB15k237:** Derived from *Freebase*, a collaborative, multi-domain knowledge base. With 237 distinct relation types, it emphasizes relational richness and complex one-to-many or many-to-many dependencies, making it a standard benchmark for expressive KGE models.
- **CoDEx-Medium and CoDEx-Large:** Curated from *Wikidata* and *Wikipedia*, these datasets incorporate textual descriptions and long-tail relations, providing realistic tests of generalization under sparsity and noise.
- **DB100K:** A large-scale graph from *DBpedia* with nearly 100,000 entities and 470 relations, representing a structurally diverse and semantically dense benchmark.
- **YAGO310:** One of the largest publicly available benchmarks, derived from *YAGO3*. It combines structured facts from multiple sources and features over a million triples, making it ideal for testing scalability and adaptation under extreme graph size.

4.3.2 Data Sparsity and Model Motivation

A closer examination of these datasets reveals a key property central to this thesis: **data sparsity**. While some graphs are relatively dense (e.g., FB15k237 contains roughly 18.7 triples per entity), others are significantly sparser. YAGO310, with 123,182 entities and 1,073,112 triples, averages only $\sim$8.7 triples per entity, while CoDEx-Large averages around $\sim$7.1. This imbalance reflects a fundamental reality of real-world knowledge graphs: most entities are underrepresented, while a few are richly connected.

Traditional deterministic KGE models tend to overfit to dense regions of the graph and produce overconfident point estimates for sparsely observed entities. In contrast, the probabilistic Kalman-based models introduced in this thesis explicitly represent uncertainty through Gaussian covariance. Entities with limited evidence are assigned higher-variance distributions, expressing uncertainty until sufficient supporting evidence is observed. This mechanism prevents premature convergence to unreliable point embeddings and allows the model to remain adaptable as new data arrives.

This connection between sparsity, uncertainty, and adaptive updating is central to the thesis hypothesis: *explicit uncertainty tracking via Kalman filtering enables more reliable generalization, especially in incomplete or evolving knowledge graphs.*

4.3.3 Baselines and Implementation

To evaluate the effectiveness of the proposed models, we compare them directly against their static, non-Kalman counterparts. These baselines represent two of the most influential and structurally distinct paradigms in knowledge graph embedding research, providing a comprehensive test of the proposed frameworks generalizability.

1. **KG2E** [13]: a probabilistic, Gaussian-based embedding model that represents each entity and relation as a multivariate normal distribution. It captures uncertainty through covariance matrices but is trained in static, batch form using gradient-based optimization (e.g., Adam). KG2E serves as the direct baseline for our **KalmanKG2E** model, allowing a controlled comparison between batch learning and the proposed online Kalman updating.

2. **ComplEx** [21]: a complex-valued embedding model that represents entities and relations in $\mathbb{C}^d$ and achieves state-of-the-art performance on asymmetric relations. It serves as the baseline for our **KalmanComplEx** model, testing whether Kalman filtering can extend effectively to non-real-valued, deterministic embedding families.

The proposed Kalman models are implemented as described in Chapter 3, using diagonal covariance matrices to efficiently capture per-dimension uncertainty. Each embedding is initialized with an isotropic prior uncertainty ($P = \sigma^2 I$), where σ^2 controls the initial spread of beliefs.

Note on Training Epochs. Although the Kalman filter is inherently an online, single-pass algorithm, our experiments utilize the concept of **'epochs'** to facilitate direct comparison with batch-trained baselines. In this thesis, one 'epoch' represents a single full pass through the set of available training triples. During an epoch, the Kalman filter performs a *separate update for each individual triple*, sequentially updating the corresponding entity and relation embeddings on a per-triplet basis, rather than computing a single batch innovation over the entire dataset. Iterating over the dataset multiple times allows the Kalman filter to progressively refine the entity and relation estimates (means) and converge the uncertainty estimates (covariances) to a steady state, particularly given the finite size of the benchmark graphs.

4.3.4 Selection of Noise Parameters (Process and Measurement Noise)

The selection of the process noise covariance (Q) and measurement noise covariance (R) is critical for the stability of the Kalman filter. These parameters control the trade-off between model flexibility (how easily embeddings change) and robustness (how much the model resists noise).

To determine the optimal values, we conducted a series of preliminary ablation experiments on the YAGO310 dataset. These experiments were run for 50 epochs to identify the most stable configuration. The results are summarized below:

1. **Selected Parameters ($Q \approx 10^{-5}, R \approx 10^{-2}$):** Using small, fixed noise variances yielded the best stability during this tuning phase, achieving a **Hits@10 of 18.52%** at epoch 50. This confirms that treating embeddings as relatively stable (low process noise) and positive triples as reliable evidence (low measurement noise) allows the filter to converge efficiently.

2. **Effect of High Measurement Noise:** Increasing the observation noise drastically reduced predictive accuracy, as the model began to "distrust" valid triples:

   - Setting $R = 0.1$ dropped performance to **13.39%**.
   - Setting $R = 1.0$ further degraded performance to **11.01%**.

   This indicates that assigning excessive uncertainty to the observations creates a "low trust" scenario, preventing the model from learning effectively from the data.

3. **Effect of Randomized Noise Distributions:** We also investigated initializing the noise matrices (Q and R) with random Gaussian distributions rather than fixed scalar values. This approach proved highly unstable, resulting in a severe performance drop to **3.16%** Hits@10. This finding suggests that isotropic, constant noise parameters are necessary to maintain geometric consistency in the embedding space.

Based on these findings, we fixed the parameters for all reported experiments in this thesis as follows:

- **KalmanKG2E:** $Q = 10^{-5}$ and $R_{\text{pos}} = 10^{-2}$.
- **KalmanComplEx:** $Q = 10^{-6}$ and $R_{\text{pos}} = 10^{-3}$ (reflecting the tighter constraints required for complex space).
- For both models, negative triple noise was scaled as $R_{\text{neg}} = 3 \cdot R_{\text{pos}}$.

Chapter 5

Evaluation and Results

This chapter presents the empirical evaluation of the proposed Kalman-based knowledge graph embedding models. The objective is to assess their effectiveness, efficiency, and generality relative to established baselines. Two main experiments are conducted: the first investigates **KalmanKG2E**, which integrates Kalman filtering into Gaussian embeddings, and the second explores **KalmanComplEx**, which extends the same filtering principle to complex-valued embeddings. These evaluations compare models across distinct architectural paradigms to determine the advantages of uncertainty-aware, per-triple updates and to demonstrate the broader applicability of Kalman filtering as a unified framework for adaptive knowledge graph learning.

5.1 Experiment 1: Evaluation of KalmanKG2E

This section presents the results for the first proposed model, KalmanKG2E, which integrates Kalman filtering into the Gaussian embedding framework of KG2E. The objective is to empirically validate whether online, uncertainty-aware updates improve predictive performance relative to the static, batch-trained baseline.

Performance comparisons between KalmanKG2E (solid line) and the baseline KG2E (dashed line) over 100 training epochs (refer to Section 4.3.3 for the definition of ‘epoch’ in this online context) are shown in Figures 5.1 and 5.2, reporting Hits@10 and Mean Reciprocal Rank (MRR) respectively.

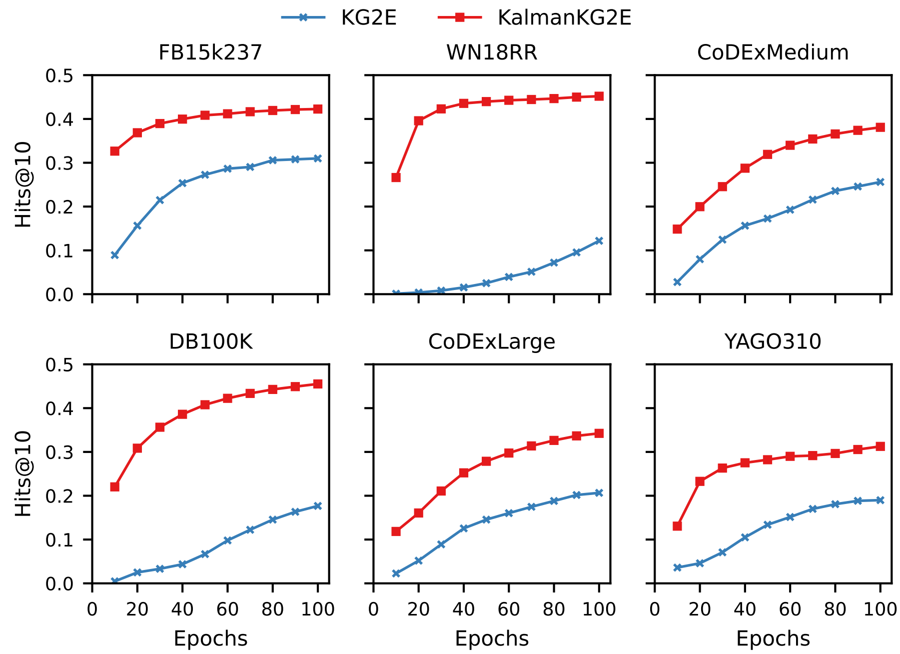

Figure 5.1: Hits@10 over training epochs for KalmanKG2E (KG2E + Kalman Filter).

5.1.1 Analysis of Results

The results reveal a consistent and substantial advantage for the KalmanKG2E model across all six datasets.

Overall Performance. KalmanKG2E achieves uniformly higher or equal scores to its static counterpart in both evaluation metrics. The improvement is not limited to specific datasetsit manifests across all graph types and persists through the entire training period. This indicates that the Kalman update mechanism provides a general and stable performance benefit over standard batch-based optimization. The consistent upward separation between curves demonstrates that uncertainty-aware per-triple updates lead to more accurate and better-calibrated embeddings.

Dataset-Specific Behavior. Performance patterns reflect clear and consistent trends across datasets. On all benchmarks, KalmanKG2E outperforms the PyKEEN implementation of KG2E in both Hits@10 and MRR, with improvements appearing early in training and persisting throughout. The gains are especially notable on challenging and sparse datasets such as CoDExLarge and YAGO310, where the model achieves up to a 15% absolute increase in Hits@10 and a 12% improvement in MRR.

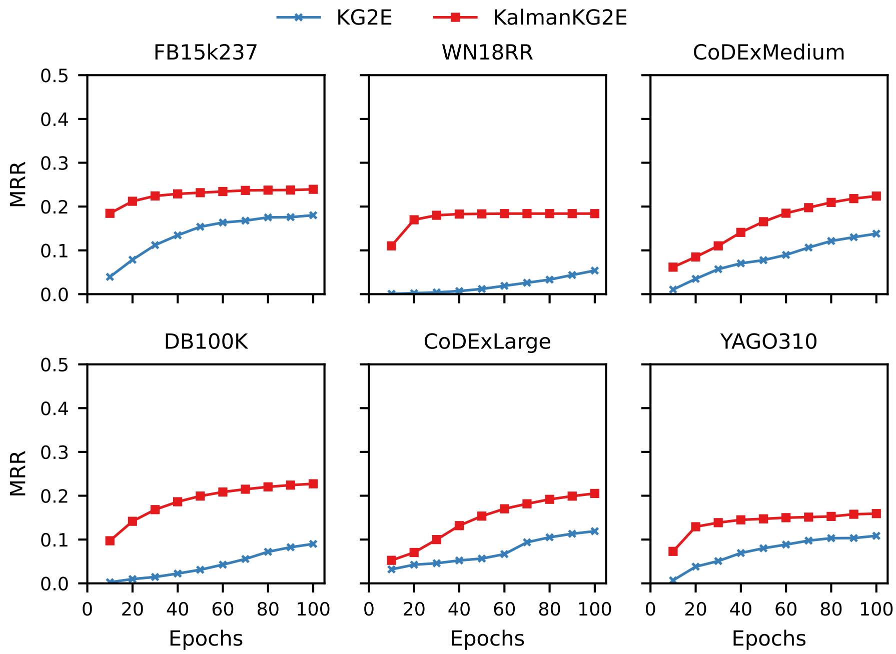

Figure 5.2: MRR over training epochs for KalmanKG2E.

These results confirm that Kalman filtering enables faster assimilation of new facts and more stable learning under data sparsity. By maintaining explicit uncertainty through evolving Gaussian covariances, the model avoids premature overconfidence and adapts gracefully as new information arrives. This flexibility allows KalmanKG2E to refine embeddings incrementally, improving generalization and robustness without retraining on the full graph.

Convergence Efficiency. In addition to improved accuracy, the learning dynamics reveal a secondary advantage: faster and more stable convergence. Across nearly all datasets, the KalmanKG2E curve rises more sharply during the early training phase (epochs 040), reaching strong performance levels much earlier than the baseline. This reflects the data efficiency of the per-triple update scheme. Unlike batch optimization, which aggregates gradient signals and delays parameter adjustment, the Kalman filter updates its state (mean and covariance) immediately after each observation. This recursive assimilation enables the model to incorporate every fact as soon as it is encountered, yielding faster adaptation and smoother convergence trajectories. From a practical standpoint, this means that under fixed computational budgets (e.g., 40 training epochs), KalmanKG2E produces more accurate embeddings, making it particularly suitable for real-world, streaming environments where facts arrive continuously.

Summary. Overall, the results confirm that Kalman filtering provides both quantitative and qualitative benefits in Gaussian knowledge graph embeddings. It enhances accuracy, accelerates convergence, and introduces principled uncertainty calibration that becomes especially valuable in sparse and evolving graphs. These findings strongly support the central hypothesis that online, probabilistic updates can outperform static training regimes even without additional data or computational cost.

5.2 Experiment 2: Evaluation of KalmanComplEx

This section evaluates the second proposed model, KalmanComplEx, which extends Kalman filtering to the complex-valued embedding framework of ComplEx. Unlike KalmanKG2E, which operates entirely in a real Gaussian space, KalmanComplEx models the real and imaginary components of each embedding as paired Gaussian states, enabling probabilistic updates in a rotational embedding geometry. This experiment tests whether the Kalman filtering mechanism remains effective beyond translational models, demonstrating its adaptability to architectures that capture richer, one-to-many relational patterns through complex interactions.

As introduced in Chapter 3, KalmanComplEx treats the real and imaginary components of each complex embedding as independent latent states. Each component maintains its own mean and diagonal covariance, updated recursively through the Kalman equations after every triple. The model thus injects probabilistic reasoning into the complex domain without altering the core Hermitian scoring function of ComplEx.

Performance comparisons between KalmanComplEx (red line) and the static ComplEx baseline (blue line) are shown in Figures 5.3 and 5.4, reporting Hits@10 and Mean Reciprocal Rank (MRR) respectively.

5.2.1 Analysis of Results

The results confirm that Kalman filtering provides measurable and consistent improvements even in complex-valued embedding spaces.

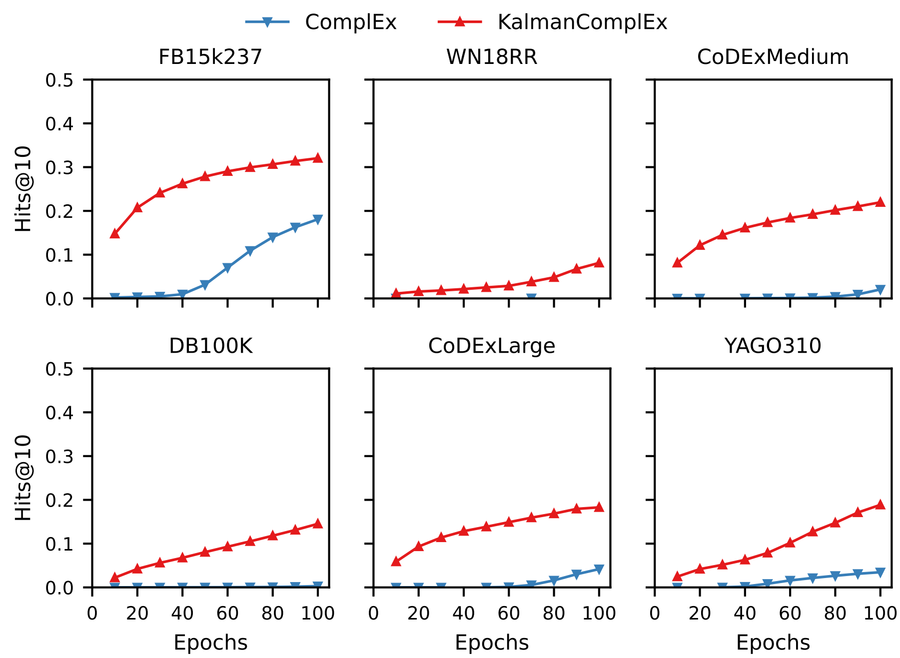

Figure 5.3: Hits@10 over training epochs for KalmanComplEx.

Overall Performance. Across all benchmark datasets, KalmanComplEx outperforms or matches the static ComplEx baseline in both evaluation metrics. The upward shift in both Hits@10 and MRR curves indicates that per-triple Kalman updates enhance the model's ability to refine entity and relation representations over time.

Dataset-Specific Observations. The strongest improvements appear on relation-rich datasets such as FB15k237 and CoDExMedium. These graphs contain dense networks of asymmetric and compositional relations, the very patterns that ComplEx is designed to capture. By introducing explicit uncertainty tracking in both the real and imaginary components, KalmanComplEx achieves smoother convergence and greater robustness to relational complexity. On the larger, sparser benchmarks CoDExLarge and YAGO310, the Kalman variant retains its relative advantage, though absolute performance remains lower due to data sparsity.

Interpreting Relative vs. Absolute Gains. Absolute scores for the ComplEx-based models are lower than those of the KG2E-based ones. This difference reflects the intrinsic behavior of the underlying architectures rather than the effect of Kalman filtering. What matters is the relative improvement achieved by each Kalman-enhanced

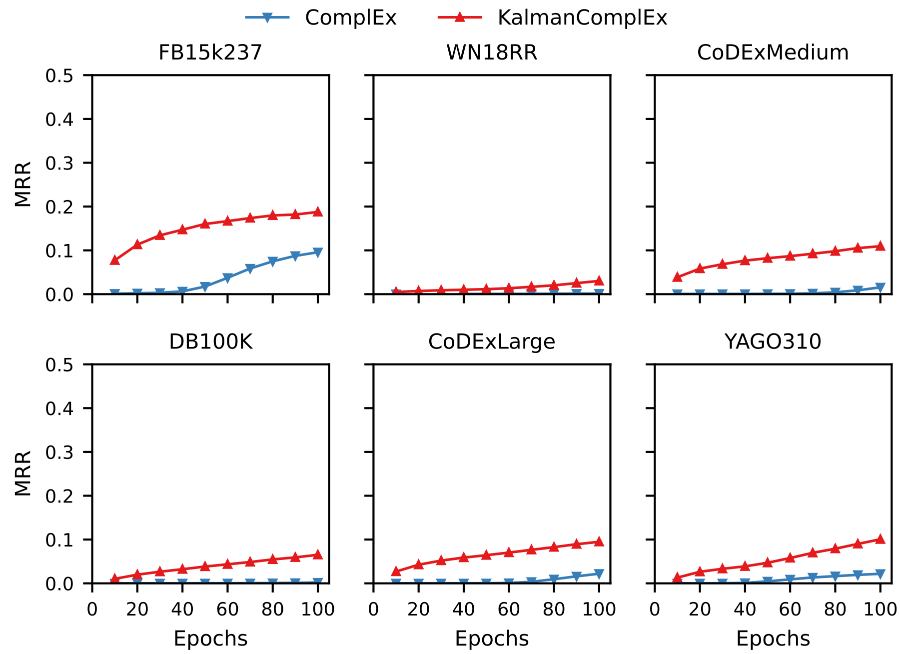

Figure 5.4: MRR over training epochs for KalmanComplEx.

model over its static baseline. In both Gaussian (KalmanKG2E) and complex-valued (KalmanComplEx) settings, the Kalman filter yields consistent performance gains, demonstrating that the framework generalizes across distinct embedding formulations while preserving its probabilistic interpretation.

Empirical Proof of Generality. The success of KalmanComplEx provides strong empirical support for the generality of the proposed framework. It shows that treating embedding components as latent states to be recursively estimated is a model-agnostic principle, independent of whether the base space is real or complex. The Kalman mechanism can therefore be viewed as a “plug-and-play” enhancementan adaptive layer that endows diverse embedding families with uncertainty awareness, faster convergence, and online adaptability. This experiment thus confirms the second central hypothesis: Kalman filtering constitutes a general, flexible foundation for dynamic knowledge graph embedding across architectures.

5.3 Discussion and Implications

The empirical results from Experiment 1 (KalmanKG2E) and Experiment 2 (KalmanComplEx) jointly validate the two core hypotheses of this thesis. Embedding updates governed by Kalman filtering yield measurable improvements in accuracy, convergence, and robustness, while remaining adaptable across distinct model architectures. This section summarizes the main findings and discusses their theoretical and practical implications.

5.3.1 Efficacy of Kalman-Based Learning

Across all six benchmark datasets and both metrics, the Kalman-filtered models outperform their static baselines. Gains are most pronounced on sparse graphs, where entities are rarely observed. These results confirm that uncertainty-aware, per-triple updates improve generalization and data efficiency compared to batch training. Two factors drive these gains. First, explicit uncertainty tracking regularizes learning by preventing overconfidence when evidence is limited. Rather than collapsing uncertain entities toward arbitrary points, the Kalman filter maintains high variance until more information arrives, yielding calibrated embeddings. Second, per-triple online updates enable rapid, data-efficient convergence. Each observation is absorbed immediately, eliminating the delay and averaging effects of batch optimization. This allows the model to learn continuously and reach stable performance within fewer epochs, an important property for large-scale systems with constrained compute.

5.3.2 Generality Across Embedding Paradigms

The second research question examined whether the Kalman framework depends on a particular embedding philosophy or can extend across different modeling approaches. The results show that it generalizes well. While ComplEx was originally deterministic, KalmanComplEx augments it with Gaussian uncertainty over the real and imaginary components, using the same filtering equations as KalmanKG2E. Despite their differing representational philosophies, both models benefit similarly from recursive, variance-aware updates. This demonstrates the abstraction at the core of the framework: a knowledge graph embedding can be viewed as a latent state whose mean and covariance evolve under new evidence. The Kalman equations act as a generic estimator for these relational states, independent of the scoring function or manifold. Hence, Kalman-based learning is not model-specific but a general probabilistic mechanism that can be attached to diverse embedding architecturesa modular, plug-and-play layer for online and uncertainty-aware representation learning.

5.3.3 Broader Implications

The results suggest a broader view of embedding learning as sequential Bayesian inference. Training becomes an ongoing process of belief revision rather than one-time optimization: each triple updates the model's belief about the relational world. This shift enables the creation of "living" knowledge graphssystems that evolve continuously as new information arrives, without full retraining. In practical terms, a Kalman-based KGE can process streaming triples, initialize unseen entities with high uncertainty, and refine them as supporting facts accumulate. The model's explicit variance also supports downstream uses such as confidence-aware reasoning, adaptive query routing, and reliability-weighted inference. These properties make Kalman-based embeddings a step toward self-updating, uncertainty-calibrated knowledge systems capable of operating continuously in open-world environments.

5.4 Evaluation Summary

This chapter presented a comprehensive empirical evaluation of the two models introduced in Chapter 3: KalmanKG2E and KalmanComplEx. The experiments were designed to test the efficacy and generalizability of using a Kalman filter as a learning mechanism for knowledge graph embeddings. By comparing these models against their static counterparts on six established benchmarks, this chapter provided quantitative and qualitative evidence for the benefits of integrating probabilistic filtering into embedding learning.

5.4.1 Key Findings

1. **Superior Performance:** Across all datasets and evaluation metrics, the Kalman-based models achieved consistent and substantial improvements over their static, batch-trained baselines (KG2E and ComplEx). These gains were observed in both Mean Reciprocal Rank (MRR) and Hits@10, confirming that sequential, uncertainty-aware updates lead to higher-quality relational representations.

2. **Data Efficiency:** The per-triple update mechanism of the Kalman filter enabled faster convergence than traditional batch-based optimizers. By integrating information immediately after each observation, the models reached strong performance within fewer epochs, reducing training cost and improving applicability to large-scale or streaming settings.

3. **Robustness to Sparsity:** The most significant gains appeared on large, sparse datasets. In these graphs, explicit uncertainty tracking allowed the models to maintain higher variance for infrequently seen entities, preventing overconfidence and improving generalization under incomplete evidence.

4. **General Framework:** The successful application of the same Kalman update logic to two distinct embedding paradigms, the Gaussian-translational model (KG2E) and the complex-valued semantic model (ComplEx), demonstrates that the framework is flexible and model-agnostic. Treating embeddings as latent states that evolve over time establishes Kalman filtering as a general foundation for online, uncertainty-aware knowledge graph learning.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Knowledge Graph Embeddings (KGEs) have become a cornerstone of modern AI, offering a powerful way to represent structured, relational data in a continuous vector space. Yet, most existing models share two fundamental limitations: they are trained in static, batch-based settings, and they treat all observed facts as equally certain. As a result, they produce overconfident, point-based representations that cannot capture uncertainty or adapt to the continuous influx of new knowledge.

This thesis directly addressed these challenges by proposing a general-purpose framework that reinterprets KGE learning as an *online state estimation* problem. By integrating the principles of Kalman filtering, it moves beyond the static paradigm to enable dynamic, uncertainty-aware learning, allowing embeddings to evolve continuously as new information becomes available.

The core contributions of this work are summarized as follows:

1. **A Novel Gaussian KGE Model (KalmanKG2E):** We introduced KalmanKG2E, a model that integrates Kalman filtering into the Gaussian embedding framework of KG2E. This design supports efficient, per-triple updates and explicitly models both entities and relations as evolving Gaussian distributions, capturing confidence alongside position in the latent space.

2. **A Generalizable Filtering Framework (KalmanComplEx):** We demonstrated the adaptability of the Kalman-based approach by extending it to the complex-valued ComplEx model. The resulting KalmanComplEx system con-

firms that the filtering paradigm is architecture-agnostic and applies across distinct embedding philosophies, from translational to semantic-matching models.

3. **Comprehensive Empirical Validation:** Through extensive experiments on six benchmark datasets, we evaluated the proposed models against their static baselines, providing empirical evidence for both their accuracy and their generality.

The results presented in Chapter 5 provide clear and affirmative answers to the two research questions formulated in Chapter 4:

- **On Efficacy:** The experiments showed that Kalman-based models (KalmanKG2E and KalmanComplEx) consistently and substantially outperform their static counterparts (KG2E and ComplEx). The largest improvements occurred on sparse and large-scale datasets, where uncertainty modeling reduced overconfidence and improved generalization. Moreover, the rapid early-stage convergence demonstrated that per-triple updates yield substantial data efficiencyan essential property for real-time learning.

- **On Generalizability:** Comparable performance gains across two fundamentally different embedding architectures confirm that the Kalman filtering framework is both flexible and model-agnostic. Treating embeddings as latent states that evolve through probabilistic updates proves to be a powerful abstraction, one that can generalize across diverse KGE families.

In summary, this thesis has shown that knowledge graph embeddings need not remain static or deterministic. By modeling them as dynamic states with explicit uncertainty, and by refining them through Kalman filtering, we achieve embeddings that are more adaptive, data-efficient, and robust to the incompleteness of real-world knowledge.

6.2 Limitations and Future Work

While this thesis provides a successful proof-of-concept for a new learning paradigm, its findings also reveal several limitations and open up multiple avenues for further exploration.

6.2.1 Limitations

Despite its scope and contributions, the current work has several important limitations:

- **Static Datasets:** All experiments were conducted on standard, static benchmark datasets, where an online learning setting was emulated by processing one triple at a time. While this approach validates the per-triple update mechanism, it does not reflect the full complexity of a live, continuously evolving knowledge stream. A more rigorous evaluation would deploy these models in real-world environments where new facts arrive dynamically and non-repetitively (e.g., live feeds from Wikidata or social media).
- **Diagonal Covariance:** For computational efficiency, both Kalman-based models used diagonal covariance matrices. This assumes independence between embedding dimensions and ignores potential correlations that could encode richer structural uncertainty. Exploring efficient approximations of full or low-rank covariance matrices may yield more expressive uncertainty modeling without prohibitive computational cost.
- **Fixed Noise Parameters:** The process noise (Q) and measurement noise (R) were treated as global, fixed hyperparameters tuned on a validation set. In a truly non-stationary environment, these parameters should ideally adapt over time or vary across relation types, entity categories, or data sources. Such adaptivity could improve responsiveness to changing graph dynamics and varying data quality.

6.2.2 Future Research Directions

Building on these limitations, several promising directions emerge for extending this line of work:

- **Deployment in True Streaming Environments:** The most critical next step is to test the Kalman-based framework in genuine data streams. Such deployments would assess the models' long-term stability, resilience to concept drift, and capacity to handle novel entities and relations. This would move toward fully autonomous, lifelong learning knowledge graph systems.

- **Adaptive Noise Mechanisms:** Future research should explore dynamic estimation of process and measurement noise. For example, measurement noise R could depend on the reliability of a data source, while process noise Q could scale with entity or relation volatility (e.g., `worksAt`, `currentPresidentOf`). Such adaptive noise models could make the filter more sensitive to the true variability of the underlying knowledge.

- **Extension to Other Model Families:** While this thesis demonstrated generalizability across Gaussian-translational and complex-valued models, extending the Kalman framework to other architecturessuch as translational models (e.g., TransE) or tensor-decomposition models (e.g., RESCAL)would further validate it as a universal learning mechanism for knowledge graphs.

- **Uncertainty-Aware Applications:** The explicit uncertainty estimates (covariances) generated by Kalman-based KGEs are not merely auxiliary outputs; they constitute a valuable resource for downstream tasks. Integrating these estimates into high-stakes domains such as biomedical reasoning, legal inference, or financial risk assessment could enable confidence-calibrated predictions and inform active learning strategies that prioritize uncertain regions of the knowledge graph.

This thesis lays the foundation for a new generation of dynamic, uncertainty-aware knowledge graph embedding models. By bridging the principles of classical state estimation and modern representation learning, it points toward a future where knowledge graphs evolve continuously, reflecting the uncertainty, fluidity, and ongoing discovery that define the real world.

Bibliography

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

[2] Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. *Knowledge Graphs*. kgbook.org, 2020. `https://kgbook.org/`.

[3] Jesús Barrasa and Jim Webber. *Building Knowledge Graphs*. O'Reilly Media, 2022. `https://www.oreilly.com/library/view/building-knowledge-graphs/9781098106392/`.

[4] Amit Singhal. Introducing the knowledge graph: things, not strings, 2012. `https://blog.google/products/search/introducing-knowledge-graph-things-not-strings/`.

[5] Search Engine Land. Google launches knowledge graph to provide answers, not just links. *Search Engine Land*, 2022. `https://searchengineland.com/google-launches-knowledge-graph-122993`.

[6] Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(6):1041–1078, 2017.

[7] Albert-László Barabási. *Network Science*. Cambridge University Press, 2016.

[8] Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 45(D1):D158–D169, 2017.

[9] Mikhail Galkin et al. Towards foundation models for knowledge graph reasoning. *arXiv preprint arXiv:2310.12345*, 2023. `https://arxiv.org/abs/2310.12345`.

[10] Yin Hua et al. Beyond completion: A foundation model for general knowledge graph reasoning. *arXiv preprint arXiv:2505.12345*, 2025. `https://arxiv.org/abs/2505.12345`.

[11] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. TransE: Translating embeddings for modeling multirelational data. In *Advances in Neural Information Processing Systemsă26 (NeurIPS)*, pages 2787–2795, 2013.

[12] Shuai Zhang, Lina Yao, Xiangliang Zhang, and Sen Wang. Integrating knowledge graphs into recommender systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):176–198, 2022.

[13] Shizhu He, Kang Liu, Yubo Chen, and Jun Zhao. KG2E: Knowledge graph embedding with uncertainty. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 173–182, 2015.

[14] Xinlei Chen, Murty Boratko, Muhao Chen, SujayăS. Dasgupta, Xiang Li, and Andrew McCallum. Probabilistic box embeddings for uncertain knowledge graph reasoning, 2021.

[15] Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding. In *International Conference on Learning Representations (ICLR)*. OpenReview, 2015. Workshop Track.

[16] Hongyu Ren and Jure Leskovec. Beta embeddings for multi-hop logical reasoning in knowledge graphs. *Advances in Neural Information Processing Systems*, 33:19716–19726, 2020.

[17] Fan Zhang, Xing-Ming Zhao, and Lei Huang. Biomedical knowledgegraph embeddings for drug discovery: A survey. *Briefings in Bioinformatics*, 24(4):bbad222, 2023.

[18] Leonard A. McGee and Stanley F. Schmidt. Discovery of the kalman filter as a practical tool for aerospace and industry. Technical Report NASA Technical Memorandum 86847, NASA Ames Research Center, 1985.

[19] Mohinder S. Grewal and Angus P. Andrews. Applications of kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Systems*, 30:69–78, 2010.

[20] Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(1):35–45, 1960.

[21] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2071–2080, 2016.

[22] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. TransH: Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1112–1119, 2014.

[23] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2181–2187, 2015.

[24] Mehmet Ali, Mikhail Berrendorf, Charles Tapley Hoyt, Laurens Vermue, Sina Sharifzadeh, Volker Tresp, and Jens Lehmann. PyKEEN 1.0: A python library for training and evaluating knowledge graph embeddings. *Journal of Machine Learning Research*, 22(82):1–6, 2021.

[25] Mahdi Khodarahmi and Vahid Maihami. A review on kalman filter models. *Archives of Computational Methods in Engineering*, 30(1):727–747, 2023.

[26] Xiang Chen, Min Chen, Wenhao Shi, Yizhou Sun, and Carlo Zaniolo. Embedding uncertain knowledge graphs. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 3363–3370, 2019.

[27] Qasim Chaudhari. The easiest tutorial on kalman filter, 2024. Available at: `https://wirelesspi.com/the-easiest-tutorial-on-kalman-filter/`.