Future Directions and Open Questions

Overview

This document explores open questions and future research directions in box embeddings. While the mathematical foundations are well-established, several frontiers remain for exploration and development.

Open Theoretical Questions

Scaling to Large Knowledge Graphs

Question: How do box embeddings scale to very large knowledge graphs (millions of entities, billions of triples)?

Current State: Most work focuses on medium-scale KGs (thousands to hundreds of thousands of entities). Large-scale evaluation is limited.

Complexity Analysis:
- **Volume computation**: $O(d)$ per box using the Bessel function formula (see the Gumbel-Box Volume document). For N boxes, total cost is $O(N * d)$.
- **Memory requirements**: Each box requires $2d$ parameters (min and max coordinates), giving total memory $O(N * d)$ for N boxes. For $N = 10^6$ entities and $d = 100$, this requires approximately 800 MB (assuming 32-bit floats).
- **Training time**: Gradient computation scales as $O(B * d)$ per batch, where B is batch size. For BoxE, scoring a fact requires $O(n * d)$ time where n is relation arity, giving total training time $O(T * B * n * d)$ for T training steps.
- **Intersection computation**: Computing box intersections is $O(d)$ per pair, but with many boxes, this becomes $O(N^2 d)$ for all pairs. Spatial data structures (e.g., R-trees) can reduce this to $O(N \log N * d)$ for range queries.

Challenges:
- **Quadratic intersection queries**: Computing all pairwise intersections scales as $O(N^2 d)$, which becomes prohibitive for large N.
- **Memory bandwidth**: Accessing $2d$ parameters per box for millions of boxes can create memory bandwidth bottlenecks.
- **Numerical precision**: High-dimensional volumes (product of d small numbers) can underflow, requiring careful log-space computation.

Future Directions:
- **Efficient volume approximation**: Methods beyond the Bessel function approximation that trade accuracy for speed (e.g., Monte Carlo with variance reduction).
- **Sparse box representations**: Only store non-zero coordinates or use compressed representations for boxes with many zero or near-zero dimensions.
- **Hierarchical box structures**: Multi-scale representation where coarse boxes contain fine boxes, enabling efficient approximate queries.
- **Distributed training strategies**: Partition knowledge graphs across multiple machines, with communication-efficient aggregation methods.
- **Spatial indexing**: Use R-trees or similar data structures to accelerate intersection queries from $O(N^2)$ to $O(N \log N)$.

Connection to Foundations: The volume calculation (see the Gumbel-Box Volume document) and numerical approximation methods provide a starting point, but more efficient methods may be

needed for very large scales. The $O(d)$ per-box complexity is already efficient; the challenge is scaling to millions of boxes rather than high dimensions.

Optimal Temperature Scheduling

Question: What is the optimal temperature schedule for Gumbel boxes during training?

Current State: Most work uses fixed or simple annealing schedules (linear, exponential decay). The optimal schedule is not well-understood.

Challenges:
- Temperature β controls the "softness" of boundaries (see the Log-Sum-Exp and Gumbel Intersection document)
- Too high: boxes become too soft, losing geometric structure
- Too low: boxes become too hard, losing gradient signal (see the Local Identifiability document)
- Optimal schedule likely depends on dataset, task, and model architecture

Future Directions:
- Adaptive temperature scheduling based on training dynamics
- Task-specific temperature optimization
- Theoretical analysis of temperature's effect on learning dynamics
- Connection to curriculum learning and annealing strategies

Connection to Foundations: The temperature parameter appears throughout the mathematical foundations (volume calculations, log-sum-exp, local identifiability), but its optimal scheduling remains an open question.

Expressiveness Comparison

Question: How do box embeddings compare to hyperbolic embeddings in practice? When is each preferable?

Current State: RegD (2025) shows that Euclidean box embeddings can achieve hyperbolic-like expressiveness, but empirical comparisons are limited.

Challenges:
- Hyperbolic embeddings excel at tree-like hierarchies
- Box embeddings excel at DAGs and partial orders
- Direct comparison requires careful experimental design
- Expressiveness depends on data structure (tree vs DAG vs general graph)

Future Directions:
- Large-scale empirical comparison on diverse datasets
- Theoretical analysis of expressiveness for different graph structures
- Hybrid approaches combining boxes and hyperbolic embeddings
- Task-specific recommendations (when to use boxes vs hyperbolic)

Connection to Foundations: The max-stability property (see the Gumbel Max-Stability document) and volume calculations provide theoretical advantages, but empirical validation is needed.

Practical Research Directions

Uncertainty Quantification

Question: How can we better quantify and utilize uncertainty in box embeddings?

Current State: Gumbel boxes provide probabilistic boundaries, but uncertainty quantification is not fully exploited.

Future Directions:
- Uncertainty-aware training objectives
- Calibrated uncertainty estimates for downstream tasks
- Connection to Bayesian neural networks and variational inference
- Applications in active learning and few-shot learning

Connection to Foundations: The probabilistic formulation (see the Containment Probability document) and Gumbel distributions provide a natural foundation for uncertainty quantification, but more work is needed to fully exploit this.

Multi-Modal Box Embeddings
Question: Can box embeddings be extended to multi-modal settings (text + images, text + knowledge graphs)?

Current State: Most work focuses on single-modal settings (text or knowledge graphs). Multi-modal extensions are limited.

Future Directions:
- Joint embedding of text and images using boxes
- Cross-modal containment relationships (e.g., image contains text description)
- Integration with vision-language models (CLIP, BLIP)
- Applications in multi-modal retrieval and reasoning

Connection to Foundations: The geometric interpretation of containment (see the Subsumption document) could naturally extend to multi-modal settings, but the mathematical framework needs extension.

Temporal Knowledge Graphs
Question: How can box embeddings handle temporal dynamics in knowledge graphs?

Current State: BoxTE (2022) introduces temporal box embeddings, but temporal dynamics are not fully explored.

Future Directions:
- Time-aware box embeddings (boxes that evolve over time)
- Temporal containment relationships (A contains B at time t)
- Integration with temporal reasoning and forecasting
- Applications in event prediction and temporal link prediction

Connection to Foundations: The max-stability property (see the Gumbel Max-Stability document) could enable efficient temporal updates, but temporal dynamics need more theoretical development.

Integration with Large Language Models
Question: How can box embeddings be integrated with large language models (LLMs) for enhanced reasoning?

Current State: LLMs excel at language understanding but struggle with structured knowledge. Box embeddings could provide structured knowledge representation.

Future Directions:
- LLM-generated box embeddings for knowledge extraction
- Box embeddings as external knowledge for LLM reasoning
- Hybrid architectures combining LLMs and box embeddings
- Applications in knowledge-augmented language models

Connection to Foundations: The subsumption relationship (see the Subsumption document) could enable LLMs to reason about hierarchical knowledge, but integration methods need development.

Implementation Challenges

Numerical Stability at Scale

Question: How can we ensure numerical stability for box embeddings in high dimensions or with extreme parameter values?

Current State: The log-sum-exp trick (see the Log-Sum-Exp and Gumbel Intersection document) and Bessel function approximation (see the Gumbel-Box Volume document) provide stability, but edge cases remain.

Future Directions:
- Improved numerical approximations for extreme cases
- Adaptive precision strategies
- Robust volume computation methods
- Connection to numerical analysis and floating-point arithmetic

Efficient Intersection Computation

Question: Can we compute box intersections more efficiently, especially for many boxes?

Current State: Intersection computation is O(d) per pair, but with many boxes, this becomes expensive.

Future Directions:
- Spatial data structures for efficient intersection queries
- Approximate intersection methods for large-scale settings
- Parallel intersection computation strategies
- Connection to computational geometry algorithms

Connection to Foundations: The log-sum-exp function (see the Log-Sum-Exp and Gumbel Intersection document) provides the mathematical foundation, but computational efficiency needs improvement.

Applications and Use Cases

Healthcare and Bioinformatics

Potential: Medical ontologies (SNOMED CT, UMLS) are natural applications for box embeddings.

Future Directions:
- Embedding of medical ontologies for clinical decision support
- Integration with electronic health records
- Applications in drug discovery and protein function prediction
- Validation of logical consistency in medical knowledge

Connection to Foundations: The subsumption relationship (see the Subsumption document) naturally models medical hierarchies, and formal ontology embedding (TransBox 2024) demonstrates feasibility.

Semantic Web and Knowledge Graphs

Potential: Large-scale knowledge graphs (DBpedia, Wikidata) could benefit from box embeddings.

Future Directions:
- Scalable embedding of web-scale knowledge graphs

- Integration with semantic web standards (RDF, OWL)
- Applications in knowledge graph completion and reasoning
- Hybrid approaches combining boxes with other geometric structures

Connection to Foundations: The mathematical foundations provide the theoretical basis, but scalability and efficiency need improvement.

Natural Language Processing
Potential: Hyponym-hypernym relationships in natural language are natural applications.

Future Directions:
- Embedding of WordNet and other lexical resources
- Integration with language models for semantic understanding
- Applications in question answering and semantic parsing
- Cross-lingual box embeddings for multilingual knowledge

Connection to Foundations: The subsumption relationship (see the Subsumption document) naturally models semantic hierarchies, but NLP-specific extensions may be needed.

Theoretical Limitations and Constraints

Expressivity Limitations
While BoxE proves **full expressivity** for knowledge graph completion with dimensionality $d = |E|^{\{n-1\}}|R|$, this dimensionality grows rapidly with the number of entities and relation arities. For large knowledge bases, this suggests that full expressivity may not translate to practical expressivity with reasonable parameter budgets.

Quantitative Analysis: Consider a knowledge graph with $|E| = 10,000$ entities and $|R| = 100$ binary relations ($n = 2$). BoxE's full expressivity theorem requires dimension $d = |E|^{n-1} * |R| = 10,000^1 * 100 = 1,000,000$. This is computationally prohibitive for most applications. In practice, much smaller dimensions (typically $d = 50$ to 200) are used, trading full expressivity for computational efficiency. The VC dimension bound of $2d$ provides a theoretical limit: with $d = 100$, we can represent at most 200 distinct containment patterns in the worst case, but real-world hierarchies have structure that allows much more efficient representation.

Practical Expressiveness: Empirical results show that box embeddings achieve strong performance on knowledge graph completion tasks with dimensions $d = 50$ to 200, far below the theoretical requirement for full expressivity. This suggests that real-world knowledge graphs have structure that allows efficient representation—the worst-case exponential growth in dimension is not realized in practice. The gap between theoretical full expressivity ($d = |E|^{\{n-1\}}|R|$) and practical expressivity ($d \approx 100$) is substantial, indicating that box embeddings can effectively represent real-world knowledge structures without requiring exponential dimension.

Formal Limitations: Research has established formal limitations of box embeddings for description logic ontologies. Box embeddings cannot faithfully represent certain description logic axioms even with appropriate parameter tuning, particularly complex intersection patterns and qualified role restrictions. These limitations stem from geometric constraints of axis-aligned boxes: arbitrary collections of axis-aligned hyperrectangles cannot exactly represent all possible concept hierarchies.

Connection to Foundations: The intersectional closure property (see the Subsumption document) ensures that intersections remain boxes, but this very property limits expressiveness for complex logical patterns that require non-box geometric structures.

Closure Properties

Boxes are **closed under intersection** but **not under union** in the classical sense. While the intersection of two axis-aligned boxes is always an axis-aligned box (or empty), the union of two boxes may require many boxes to represent exactly. This asymmetry limits expressiveness for certain types of relational structures requiring symmetric treatment of union and intersection operations.

Connection to Foundations: The max-stability property (see the Gumbel Max-Stability document) ensures algebraic closure for intersection operations, but union operations lack this closure property, creating fundamental limitations for certain logical operations.

Optimization Challenges

Box embeddings present significant optimization challenges distinct from point-based embeddings. Large flat regions of local minima occur when disjoint boxes should remain separated, yet optimization discovers minimal-energy configurations where boxes approach boundaries without crossing. These flat regions substantially slow convergence and can trap optimization in suboptimal configurations.

Connection to Foundations: The local identifiability problem (see the Local Identifiability document) addresses gradient sparsity, but flat loss landscapes remain a challenge even with Gumbel boxes, particularly in high-dimensional settings.

VC Dimension and Expressiveness Bounds

Question: What are the theoretical limits on expressiveness for box embeddings?

Current State: BoxE proves full expressivity with dimensionality $d = |E|^{\{n-1\}}|R|$, but this grows rapidly with the number of entities. For large knowledge bases, this suggests practical expressiveness may be limited.

VC Dimension of Axis-Aligned Boxes: The class of axis-aligned boxes (hyperrectangles) in $\mathbb{R}^d$ has VC dimension exactly $2d$. This classical result from statistical learning theory provides a fundamental bound on the complexity of box embeddings:

- **Lower bound**: One can construct a set of $2d$ points that is shattered by axis-aligned boxes, establishing VCdim $\geq 2d$.
- **Upper bound**: For any set of $2d + 1$ points, at least two points share an extremal coordinate (min or max in some dimension). Any axis-aligned box containing all extremal points must include at least one additional point, so no set of size $2d + 1$ can be shattered, giving VCdim $< 2d + 1$.
- **Exact value**: Combining these bounds gives VCdim $= 2d$ exactly.

This result has practical implications: for a d-dimensional embedding space, box embeddings can represent at most $2d$ distinct containment patterns in the worst case. However, this is a worst-case bound—in practice, the structure of real-world hierarchies allows much more efficient representation.

Theoretical Results: Query2Box (Ren et al., 2020) proves a negative result: embedding EPFO queries as single points or boxes requires dimension proportional to the number of KG entities. Specifically, for M conjunctive queries with disjoint answer sets, the VC dimension of the distance function class must be at least M to model any EPFO query. For real-world KGs like FB15k with $M \approx 13,365$ disjoint queries, this implies dimensionality $\Theta(|V|)$, which is not low-dimensional. This motivates the DNF transformation strategy: by transforming queries to Disjunctive Normal Form, each conjunctive query can be embedded in low-dimensional space, then aggregated by taking the minimum distance to any box.

Computational Complexity: BoxE achieves time complexity $O(n * d)$ for scoring facts and applying updates, where n is the maximal relation arity and d is the embedding dimension. Space complexity is $O((|E| + n * |R|) * d)$, which is comparable to standard bilinear/translational KGE models up to the factor n. This linear complexity in dimension makes box embeddings computationally efficient, even when high-dimensional embeddings are required for expressiveness.

Connection to Foundations: The VC dimension argument establishes fundamental limits on expressiveness, showing that certain query structures cannot be efficiently represented in low-dimensional box space. The DNF transformation provides a practical workaround, but understanding these theoretical limits helps guide model design and hyperparameter selection. The $2d$ VC dimension bound provides a concrete measure of representational capacity, while the $O(n * d)$ computational complexity shows that this capacity comes at reasonable computational cost.

Box Offset and Relation Cardinality

Question: How do box sizes relate to relation patterns and answer set cardinalities?

Current State: Query2Box demonstrates that box offsets (sizes) correlate with the number of answer entities. One-to-many relations tend to have larger offset embeddings, while one-to-one relations have smaller offsets.

Empirical Findings: On FB15k, relations with the largest box sizes include `/common/.../topic` (3616.0 average answers, box size 147.0) and `/people/.../spouse` (889.8 average answers, box size 114.3). Relations with the smallest box sizes include `/architecture/.../owner` (1.0 average answers, box size 2.3) and `/base/.../dog_breeds` (2.0 average answers, box size 4.0). This correlation demonstrates that the adaptive box size mechanism naturally captures relation cardinality: relations that connect entities to many other entities require larger boxes to contain all answer entities.

Practical Implications: This adaptive sizing provides interpretability—box sizes directly indicate relation cardinality. It also suggests that volume regularization should be relation-aware: one-to-many relations may need different regularization strengths than one-to-one relations.

Connection to Foundations: The volume calculations (see the Gumbel-Box Volume document) provide the mathematical foundation, but the adaptive sizing mechanism demonstrates how the probabilistic interpretation naturally captures semantic properties of relations.

Volume Regularization: Thresholds and Strategies

Question: What are optimal volume regularization thresholds and strategies for different tasks and dimensions?

Current State: Most work uses fixed thresholds (typically $\tau = 0.1$ to 0.5) with uniform regularization across all boxes. Relation-aware and dimension-aware regularization are not well-explored.

Empirical Findings:
- **WordNet hypernym prediction**: Optimal thresholds $\tau = 0.1$ to 0.2 for boxes in $[0, 1]^d$ with $d = 50$ to 100
- **Knowledge graph completion**: Thresholds $\tau = 0.3$ to 0.5 work well for FB15k-237 and WN18RR
- **High dimensions**: Thresholds should scale as $\tau \propto (0.5)^d$ to account for volume decay in high dimensions (proportional to $(0.5)^d$)
- **Relation-specific**: One-to-many relations may need looser thresholds ($\tau = 0.5$) while one-to-one relations need tighter thresholds ($\tau = 0.1$)

Regularization Strategies:
1. **Fixed threshold**: $L_{\text{reg}} = \sum_j 1_{\text{Vol}(\alpha^{(j)}) > \tau} * \text{Vol}(\alpha^{(j)})$ — simple but may be too rigid
2. **Adaptive threshold**: $\tau(t) = \tau_0 * \left(1 - \frac{t}{T}\right)$ where t is training step and T is total steps — allows boxes to grow early, then shrink
3. **Relation-aware**: Different thresholds τ_r for different relation types r — accounts for varying cardinalities
4. **Soft regularization**: Replace indicator with sigmoid: $L_{\text{reg}} = \sum_j \sigma\left(\frac{\text{Vol}(\alpha^{(j)}) - \tau}{s}\right) * \text{Vol}(\alpha^{(j)})$ where s is a smoothing parameter — provides smooth gradients

Future Directions:
- Learnable thresholds: Make τ a learnable parameter optimized jointly with box embeddings
- Hierarchical regularization: Enforce volume conservation (parent volume $\geq \sum$ of children volumes)
- Task-specific optimization: Develop automatic threshold selection based on validation performance
- Theoretical analysis: Derive optimal thresholds from first principles based on dataset statistics

Connection to Foundations: Volume regularization builds on the volume calculations (see the Gumbel-Box Volume document) and containment probability (see the Containment Probability document). The threshold parameter τ controls the trade-off between expressiveness (larger boxes can model more entities) and precision (smaller boxes provide more specific containment relationships).

Theoretical Extensions

Beyond Axis-Aligned Boxes
Question: Can we extend box embeddings to rotated or non-axis-aligned boxes?

Current State: All work focuses on axis-aligned boxes (hyperrectangles). Rotated boxes would increase expressiveness but complicate computation.

Future Directions:
- Rotated box embeddings with learnable rotation angles
- Theoretical analysis of expressiveness gains
- Efficient computation methods for rotated boxes
- Applications where rotation is semantically meaningful

Connection to Foundations: The current mathematical framework assumes axis-alignment. Extensions would require new volume calculations and intersection methods.

Higher-Order Relationships
Question: Can box embeddings handle relationships beyond binary (e.g., ternary, n-ary)?

Current State: Most work focuses on binary relationships (head, relation, tail). Higher-order relationships are limited.

Future Directions:
- Box embeddings for n-ary relationships
- Compositional reasoning with multiple boxes
- Applications in event extraction and relation extraction
- Theoretical analysis of expressiveness for higher-order structures

Connection to Foundations: The intersection operation (see the Log-Sum-Exp and Gumbel Intersection document) could be extended, but the mathematical framework needs development.

Summary

The mathematical foundations of box embeddings are well-established, but many frontiers remain:

Theoretical: Scaling, expressiveness, optimal scheduling **Practical**: Uncertainty quantification, multi-modal extensions, temporal dynamics **Implementation**: Numerical stability, efficient computation **Applications**: Healthcare, semantic web, natural language processing **Extensions**: Rotated boxes, higher-order relationships

The future of box embeddings is bright, with active research addressing these challenges and exploring new directions. The mathematical foundations we've established provide a solid base for future developments.

Connection to Mathematical Foundations

All future directions build on the mathematical foundations:

- **Subsumption**: Enables logical reasoning and hierarchical modeling
- **Gumbel Box Volume**: Provides computational tractability
- **Containment Probability**: Enables probabilistic reasoning
- **Gumbel Max-Stability**: Ensures algebraic closure and efficiency
- **Log-Sum-Exp**: Provides numerical stability
- **Local Identifiability**: Enables effective learning

These foundations provide the theoretical basis for addressing open questions and exploring new directions.

Next Steps

For implementation details and practical guidance, see the code documentation. For recent applications, see the Applications document. For mathematical foundations, see the previous documents in this series.