Modeling Transitivity and Cyclicity in Directed Graphs via Binary Code Box Embeddings

Dongxu Zhang, Michael Boratko, Cameron Musco, Andrew McCallum
University of Massachusetts Amherst
{dongxuzhang, mboratko, cmusco, mccallum}@cs.umass.edu

Abstract

Modeling directed graphs with differentiable representations is a fundamental requirement for performing machine learning on graph-structured data. Geometric embedding models (e.g. hyperbolic, cone, and box embeddings) excel at this task, exhibiting useful inductive biases for directed graphs. However, modeling directed graphs that both contain cycles and some element of transitivity, two properties common in real-world settings, is challenging. Box embeddings, which can be thought of as representing the graph as an intersection over some learned supergraphs, have a natural inductive bias toward modeling transitivity, but (as we prove) cannot model cycles. To this end, we propose *binary code box embeddings*, where a learned binary code selects a subset of graphs for intersection. We explore several variants, including global binary codes (amounting to a union over intersections) and per-vertex binary codes (allowing greater flexibility) as well as methods of regularization. Theoretical and empirical results show that the proposed models not only preserve a useful inductive bias of transitivity but also have sufficient representational capacity to model arbitrary graphs, including graphs with cycles.

1 Introduction

Many real-world networks, such as social media interactions, paper citations, web links, and ontologies, are naturally represented as directed graphs [14, 6]. Two common properties of these graphs are transitivity and cyclicity. A *cycle* in a directed graph is a directed path starting from a vertex and traversing back to itself. For example, “organic matter” $\rightarrow$ “worm” $\rightarrow$ “fish” $\rightarrow$ “cat” $\rightarrow$ “organic matter” is a food cycle. *Transitivity* in a directed graph is the property that if there exists a directed path from u to v, then edge (u, v) also exists. For example, if “cat is mammal” and “mammal is animal” are true, “cat is animal” is true.

In the age of deep learning, it is necessary to determine a way to capture the salient information from a graph via some differentiable parameterization. To this end, various graph embedding methods have been proposed. Some work, such as DeepWalk [17] and Node2vec [8], maps each vertex to a vector in Euclidean space. These methods perform a low-rank factorization of the adjacency matrix or the graph Laplacian [18] and are designed to model *undirected* graphs. These can be extended to capture edge asymmetry in *directed* graphs by using two separate representations - one for source, and one for target - either unconstrained or related with one another via some function. It has been proven that using a dot product or distance-based energy function, separate source and target vectors can encode any graph (given sufficient dimension) [1], including graphs with cycles. When the source and target representations live in separate spaces, however, relationships which result from following directed paths (e.g., transitivity) are harder to encode. For example, given edge (i, j) and (j, k), a vector model can learn to make $|s_i - t_j|_2 \approx 0$ and $|s_j - t_k|_2 \approx 0$, where s_i, s_j are source vectors and t_j, t_k are target vectors. However, the condition $|s_i - t_k|_2 \approx 0$ which would represent the transitive edge (i, k) has no encouragement to hold, as the source and target spaces are entirely disconnected.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Transitivity cannot be trivially injected via symbolic rules, e.g., adding all transitive edges to a directed graph. This is because most real world edges are not *strictly* transitive: the degree of transitivity is "soft", and may hold locally but not globally, or vary for different edge types or sets of nodes in the graph. Some work, such as HOPE [15] and APP [29], attempts to capture transitivity by factorizing high-order proximity signals instead of the graph Laplacian. This branch of work is limited by the imperfection of high-order proximity scores (for example, these scores do not model cycles well). ATP [25] resolves this by breaking cycles in the graph, accepting a loss of graph information.

An alternative approach is to represent nodes in an embedding space with additional geometric structure. For example, hyperbolic embeddings leverage the negative curvature of hyperbolic space to provably model trees with less distortion [22]. Region-based embeddings such as box embeddings [28, 11, 3] and hyperbolic entailment cones [13, 7, 21, 9] have a natural bias toward modeling transitivity. These region-based embeddings can capture the transitivity in a directed graph without relying on high-order proximity scores, using only the original adjacency matrix as supervision. Previous work [2] proved that box embeddings can model any directed acyclic graph (DAG). A natural question, therefore, is whether box embeddings can capture graphs with cycles. As we will show in Section 2, this is not the case.

In this work, we propose *binary code box embeddings* [1], a generalization of box embeddings to represent arbitrary directed graphs. The model is motivated by the intuition that a given directed graph can be regarded as a union of multiple sub-graphs, where each sub-graph is acyclic, and therefore can be represented using boxes. We introduce the concept of a "binary code" which selects these sub-graphs.

Our contributions lie in three folds:

- We propose global (GBC-BOX) and per-vertex (VBC-BOX) binary code box models, a generalization of box embeddings capable of representing arbitrary directed graphs.
- We analyze theoretically the limitations of existing box embedding models when representing cycles. We also prove that, given sufficient dimensions, both binary code box models can model any directed graph. This establishes that, in theory, the representational capacity of these models is not limited.
- We evaluate our model on graph reconstruction and link prediction tasks with various synthetic graphs and real world graphs, and observe that our proposed methods perform the best in almost all scenarios, especially when a graph has strong cyclicity and transitivity.

2 Background

Given a simple[2] directed graph G with vertices and edges $(\mathcal{V}, \mathcal{E})$, we seek to represent the vertices using some mapping $\phi : \mathcal{V} \to \mathbb{R}^d$, and an energy function $\mathrm{E} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_+$ providing a score (based on ϕ and, perhaps, some hyperparameters $\boldsymbol{\lambda}$) which is interpreted as the negative log probability of edge existence, $\mathrm{E}(u, v; \phi, \boldsymbol{\lambda}) = -\log P((u, v) \in \mathcal{E})$. We can view these probabilities as a weighted graph, however in practice it is often necessary to make a hard decision on edge existence, which is done by choosing a (global) threshold T and binarizing the output. We denote the energy function for a particular model M with a subscript, i.e. E_{M}, and say that M is capable of modeling a graph $G = (\mathcal{V}, \mathcal{E})$ if there is some ϕ, $\boldsymbol{\lambda}$, and T such that $\mathcal{E} = \{(u, v) \mid \mathrm{E}_{\mathrm{M}}(u, v; \phi, \boldsymbol{\lambda}) < T\}$.

2.1 Boxicity

Let $\mathbf{I}$ be the set of closed and bounded intervals in $\mathbb{R}$. An *interval graph* is an undirected graph $G = (\mathcal{V}, \mathcal{E})$ such that there exists a mapping $\varphi : \mathcal{V} \to \mathbf{I}$ for which

$$\{u, v\} \in \mathcal{E} \quad \iff \quad \varphi(u) \cap \varphi(v) \neq \emptyset.$$

[1] Our code is available at `https://github.com/iesl/geometric_graph_embedding`.

[2] A *simple* directed graph is one without multiple edges or self-loops, i.e. the adjacency matrix contains only 0s and 1s, with 0s on the diagonal. We also synthetically remove self-loops from the graph modeled by the learned representations.

More generally, we can consider $\mathbf{B}^d$, the set of d-dimensional "boxes", which are Cartesian products of intervals,

$$\prod_{i=1}^{d} [x_i^-, x_i^+] = [x_1^-, x_1^+] \times \cdots \times [x_d^-, x_d^+] \subseteq \mathbb{R}^d. \tag{1}$$

where x_i^- and x_i^+ are min and max coordinates in dimension i. As defined by Roberts [20], the *boxicity* of an undirected graph G is the smallest dimension d such that there exists a mapping $\varphi : \mathcal{V} \to \mathbf{B}^d$ for which $\{u, v\} \in \mathcal{E}$ if and only if $\varphi(u) \cap \varphi(v) \neq \emptyset$. Equivalently, the boxicity is the minimal number of interval graphs whose intersection is G.

2.2 Box Embeddings

Vilnis et al. [28] provide a way of using boxes to represent directed graphs by defining an asymmetric energy function based on box volumes, and subsequent work has introduced various improvements and extensions of this idea Dasgupta et al. [4], Boratko et al. [2]. In this section we will define the energy function of box embedding models under a common framework, in preparation to motivate our extension to binary code box embeddings.

The energy function for all box embedding variants has the form

$$\mathrm{E}(u, v; \phi, \boldsymbol{\lambda}) = -\log \prod_{i=1}^{d} F(\phi(u)_i, \phi(v)_i; \boldsymbol{\lambda}), \tag{2}$$

where $\phi(u)_i$ are the parameters associated with node u in dimension i and $F(\phi(u)_i, \phi(v)_i; \boldsymbol{\lambda}) \in [0, 1]$ is a per-dimension score representing the probability of edge existence. [3] The model originally defined in Vilnis et al. [28] represented each node using a box, as in (1). The per-dimension parameters are the endpoints of an interval, $\phi(u)_i = [\phi(u)_i^-, \phi(u)_i^+]$, and the score function is defined as

$$F_{\text{Box}}([x^-, x^+], [y^-, y^+]) = \frac{|[x^-, x^+] \cap [y^-, y^+]|}{|[y^-, y^+]|} = \frac{\max(\min(x^+, y^+) - \max(x^-, y^-), 0)}{\max(y^+ - y^-, 0)}.$$

This encourages the box for a given vertex to contain (or overlap highly with) the boxes for it's children. Boxes which are disjoint or contained in one another can present problems for training, however. Dasgupta et al. [3] addressed this by modeling the endpoints of intervals using Gumbel random variables. The per-dimension score can be written as[4]

$$F_{\text{G-Box}}((x^-, x^+), (y^-, y^+); (\tau, \nu)) = \frac{\mathrm{LSE}_\nu\left(-\mathrm{LSE}_\tau(-x^+, -y^+) - \mathrm{LSE}_\tau(x^-, y^-), 0\right)}{\mathrm{LSE}_\nu(y^+ - y^-, 0)}, \tag{3}$$

where LSE_t denotes the following continuous extension of $\mathrm{LogSumExp}$ with temperature $t \geq 0$:

$$\mathrm{LSE}_t(\mathbf{x}) = \begin{cases} t \log(\sum_i e^{x_i/t}) & \text{if} \quad t > 0, \\ \max(\mathbf{x}) & \text{if} \quad t = 0. \end{cases} \tag{4}$$

In practice, the temperatures τ and ν are tuned separately as global hyperparameters, however when they are equal the parameters x^-, y^- (resp. x^+, y^+) can be interpreted as the mean of the $\mathrm{GumbelMax}$ (resp. $\mathrm{GumbelMin}$) random variables with scale $\nu = \tau$, and $F_{\text{G-Box}}$ approximates a ratio of expected box volumes. Note that for any $(x^-, x^+), (y^-, y^+) \in \mathbb{R}^2$, $F_{\text{G-Box}}$ is continuous with respect to $\tau, \nu \in \mathbb{R}_{\geq 0}$, and $F_{\text{G-Box}}((x^-, x^+), (y^-, y^+); 0, 0) = F_{\text{Box}}([x^-, x^+], [y^-, y^+])$.

Boratko et al. [2] takes this one step further, using a mapping which learns 4 parameters per dimension: $\phi(u)_i = (\phi(u)_i^-, \phi(u)_i^+, \phi(u)_{i,\tau}, \phi(u)_{i,\nu})$. The per-dimension score function is then defined as

$$F_{\text{T-Box}}((x^-, x^+, x_\tau, x_\nu), (y^-, y^+, y_\tau, y_\nu)) = F_{\text{G-Box}}((x^-, x^+), (y^-, y^+); (\tfrac{x_\tau + y_\tau}{2}, \tfrac{x_\nu + y_\nu}{2})). \tag{5}$$

[3] As in boxicity, we can interpret these box embedding models as representing a graph as an intersection of interval graphs, one for each dimension.

[4] In Dasgupta et al. [4] they interpret $\phi(u)_i^\pm$ as the "location" parameters of the distribution, resulting in a slightly different form of the score function, however as we show in Appendix A our score (3) leads to an equivalent model.

3 Existing Representational Capacity and Limitations

3.1 Representational Capacity

We would like to know each model's representational capacity (the set of graphs capable of being modeled) as well as how this may change depending on hyperparameter settings. It was proven in Boratko et al. [2] that BOX can model any DAG, and of course since T-BOX is equivalent to BOX when $\phi(u)_{i,\tau} = \phi(u)_{i,\nu} = 0$ this also holds for T-BOX. As defined in Section 2, we can also say that G-BOX is capable of modeling any DAG, since it is equivalent to BOX when we set the temperature hyperparameters to zero (i.e. $\tau = \nu = 0$), however this is practically quite different – the temperatures are not trainable as in T-BOX, and would never be set to 0 in order to avoid the training difficulties of BOX. Hence, the existing proof of representational capacity in [2] says very little about the practical representational capacity of G-BOX.[5] Thankfully, more can be proven.

Theorem 1. *Given a threshold T, temperature hyperparameters τ, ν, there exists and a bijection ψ on the set of parameterizations $\{\mathcal{V} \to \mathbb{R}^{2d}\}$ such that for all $u, v \in \mathcal{V}$,*

$$\mathrm{E}_{\text{G-Box}}(u, v; \psi(\phi), (\tau, \nu)) < T \qquad \Longleftrightarrow \qquad \mathrm{E}_{\text{Box}}(u, v; \phi) < T.$$

In other words, for *any* temperature hyperparameters, G-BOX can represent any graph representable by BOX. For the proof, see Appendix B. Proposition 3 in Boratko et al. [2] states that any DAG can be modeled by BOX in $\mathcal{O}((\Delta + 2) \log |\mathcal{V}|)$ dimensions with $\mathcal{O}(D(\Delta + 2) \log^2 |\mathcal{V}|)$ bits of precision per box, where $D \leq |\mathcal{V}|$ is the depth of G and Δ is the maximum degree. Combining this with Theorem 1 we have the following:

Corollary 2. *Let G be any DAG. Given any $\tau, \nu \in \mathbb{R}_{\geq 0}$, there exists a mapping $\phi : \mathcal{V} \to \mathbb{R}^{2d}$ and a threshold $T > 0$ such that $\mathrm{E}_{\text{G-Box}}(u, v; \phi, (\tau, \nu)) < T$ if and only if $(u, v) \in \mathcal{E}$, where $d = \mathcal{O}((\Delta + 2) \log |\mathcal{V}|)$, and Δ is the maximum degree in G.*

In other words, for any setting of temperature hyperparamters, G-BOX can model any DAG.

3.2 Limitation on Modeling Cycles

In this section, we point out that G-BOX cannot model any graph containing a (directed) *chordless cycle*, which is a cycle such that no two vertices are connected by an edge which does not belong to the cycle. Furthermore, we show that while G-BOX can model certain graphs with cycles, it may be sensitive to a perturbation of the parameters.

Theorem 3. *If E is such that $\mathrm{E}(u, v) - \mathrm{E}(v, u) = g(u) - g(v)$ for some function $g : \mathcal{V} \to \mathbb{R}$ then it cannot model any graph containing a chordless cycle with more than 2 nodes.*

Proof. Suppose the vertices $1, 2, ..., N$ comprise a chordless cycle, the edges of which are $D = \{(1, 2), (2, 3), ..., (N - 1, N), (N, 1)\}$. Suppose further that we can model the graph containing this cycle using energy E and threshold T. In particular, we have $\mathrm{E}(u, v) < T$ and $\mathrm{E}(v, u) \geq T$ for $(u, v) \in D$. This implies that $g(u) - g(v) = \mathrm{E}(u, v) - \mathrm{E}(v, u) < 0$, and thus $g(u) < g(v)$ for each $(u, v) \in D$. Hence $g(1) < g(2) < \cdots < g(N) < g(1)$, which is a contradiction. □

Corollary 4. G-BOX *cannot model any graph containing a chordless cycle on more than 2 nodes.*

Proof. Theorem 3 applies to $E_{\text{G-Box}}$ with $g(u) = -\log \prod_{i=1}^{d} \mathrm{LSE}_\nu(\phi(u)_i^+ - \phi(u)_i^-, 0)$. □

It is possible to avoid the contradiction in Theorem 3 by the introduction of one reverse edge, and in this case we observe that it is theoretically possible for G-BOX to represent a graph.

Proposition 1. *If G is a graph which is the union of a chordless cycle and one reverse edge,* G-BOX *can model G in 2 dimensions.*

The proof of this statement is contained in Appendix C, where we also prove that, while it is possible to represent such a graph, it is highly sensitive to perturbation of the parameters in a way which is dependent on the length of the cycle, and therefore may be challenging to learn.

[5] This limitation was acknowledged in Boratko et al. [2] just before Section 4.3 as part of the motivation for the trainable temperatures of T-BOX.

4 Method

In this section, we will introduce the binary code box embedding concept, which includes a family of models whose shared feature is the use of learned binary codes to select subsets of dimensions. Two we will focus on in particular include GBC-BOX, which uses "global" binary code vectors, and VBC-BOX, which uses per-vertex binary code vectors. The following topics will be covered: the motivation for binary codes, the definition of GBC-BOX and VBC-BOX, their representational capacity to model arbitrary directed graphs, our learning objectives and regularization, the models' inductive biases and strengths, and some discussion about their limitations and alternative variants.

4.1 Motivation

The idea of binary code boxes is to allow more flexibility than simply taking an intersection over interval graphs, as captured by boxicity (see Section 2.1). Recall, in the undirected case, the boxicity of a graph $G = (\mathcal{V}, \mathcal{E})$ was equivalent to the smallest number d such that for some set of interval graphs $S = \{G_i = (\mathcal{V}, \mathcal{E}_i)\}_{i=1}^{d}$, we have $\mathcal{E} = \cap_{i=1}^{d} \mathcal{E}_i$, i.e.

$$\{u, v\} \in \mathcal{E} \quad \iff \quad \forall \mathcal{F} : (\mathcal{V}, \mathcal{F}) \in S, \quad \{u, v\} \in \mathcal{F}.$$

There are various ways to increase the flexibility of this representation. For example, we could consider allowing a union over intersections by specifying k subsets of S, $\{S_i\}_{i=1}^{k}$, for which $\mathcal{E} = \cup_{i=1}^{k} \cap_{(\mathcal{V},\mathcal{F}) \in S_k} \mathcal{F}$, i.e.

$$\{u, v\} \in \mathcal{E} \quad \iff \quad \exists i : \forall \mathcal{F} : (\mathcal{V}, \mathcal{F}) \in S_i, \quad \{u, v\} \in \mathcal{F}. \tag{6}$$

To allow for even greater flexibility, we could allow each vertex to select a subset of graphs to intersect. Formally, we allow the specification of a function $\psi : \mathcal{V} \to 2^S$ which assigns a subset of interval graphs to each vertex, for which

$$\{u, v\} \in \mathcal{E} \quad \iff \quad \forall \mathcal{F} : (\mathcal{V}, \mathcal{F}) \in \psi(u) \cap \psi(v), \quad \{u, v\} \in \mathcal{F}. \tag{7}$$

For the undirected case the advantage is minimal. Increasing the flexibility in these ways may allow us to represent an undirected graph in smaller dimension (or, equivalently, using a smaller number of interval graphs), however as mentioned previously we know any undirected graph can be represented as an intersection of interval graphs. For directed graphs, however, this is not the case, and (as we prove in Section 4.3) this generalization allows for any directed graph to be represented.

4.2 Definition

In order to capture the idea of a "union of intersections" specified in (6) we consider learning k "binary code" vectors $\mathbf{b}_j \in [0, 1]^d$. Each binary vector corresponds to a selection of which dimensions to include - if the i^{th} component is 0 the scores for edges in this dimension should be ignored, and if it is 1 they should be included. For convenience, we will represent these as the columns of a $d \times k$ matrix $B \in [0, 1]^{d \times k}$. The energy function in this case is

$$\mathrm{E}_{\text{GBC-Box}}(u, v; (\phi, B), (\tau, \nu, k)) := \min_{j \in \{1, \ldots, k\}} \left(-\log \prod_{i=1}^{d} F_{\text{G-Box}}(\phi(u)_i, \phi(v)_i; (\tau, \nu))^{B_{i,j}} \right) \tag{8}$$

In order to capture the notion of per-vertex subset selection in (7), we learn 3 parameters per dimension, which we denote as $\phi(u)_i = (\phi(u)_i^-, \phi(u)_i^+, \phi(u)_i^\diamond) \in \mathbb{R} \times \mathbb{R} \times [0, 1]$. The binary code $\phi(u)_i^\diamond$ indicates whether this dimension should be taken into account when calculating the probability of edge existence for edges involving this node - if it is 0, the dimension should be ignored, and if it is 1 it may be included. We incorporate this at the level of the per-dimension score function as follows:

$$F_{\text{VBC-Box}}((x^-, x^+, x^\diamond), (y^-, y^+, y^\diamond); (\tau, \nu)) := F_{\text{G-Box}}((x^-, x^+), (y^-, y^+); (\tau, \nu))^{x^\diamond y^\diamond}. \tag{9}$$

Using the product in the exponent is a relaxation of the intersection $\psi(u) \cap \psi(v)$ from (7). When computing the probability of an edge (u, v), the binary codes can learn to ignore certain dimensions by making $\phi(u)_i^\diamond$ or $\phi(v)_i^\diamond$ equal to 0.

In the following, we point out two perspectives which provide further intuition behind these models:

Generalization: Both GBC-BOX and VBC-BOX are a generalization of G-BOX, and revert back to it when all binary codes are 1, in which case all dimensions are used for volume calculation. As we will show in Section 4.3 these models are strictly more expressive, as when some binary codes are less than 1 these models can represent more complex graphs.

Projection: If $B_{i,j} \in \{0, 1\}$ we can think of this defining a set of projections $\{P_j\}_{j=1}^k$ where P_j projects the boxes parameterized by ϕ into the $\sum_{i=1}^d B_{i,j}$ dimensional subspace where $B_{i,j} = 1$. Similarly, for VBC-BOX, if $\phi(u)_i^\diamond \in \{0, 1\}$ we can think of this determining the dimensions $D_u \coloneqq \{i : \phi(u)_i^\diamond = 1\}$ which the box will be projected in. Given an edge (u, v), we project into dimensions $D_u \cap D_v$ before determining the edge existence.

4.3 Representational Capacity

The energy functions for GBC-BOX and VBC-BOX were constructed such that Theorem 3 would not apply, thus making it possible that they may be capable of representing some graphs with cycles. In this section, we prove that both can model *any* directed graph.

Theorem 5. *Given a directed graph $G = (\mathcal{V}, \mathcal{E})$ and any $\tau, \nu \geq 0$ and $k \geq 2$, there exists a threshold $T > 0$, parameters $\phi : \mathcal{V} \to \mathbb{R}^{2d}$, and binary codes $B \in [0, 1]^{d \times k}$ for which*

$$\mathrm{E}_{\text{GBC-Box}}(u, v; (\phi, B), (\tau, \nu)) < T \qquad \iff (u, v) \in \mathcal{E}.$$

Proof. Given a directed graph $G = (\mathcal{V}, \mathcal{E})$, let $(<, \mathcal{V})$ be an arbitrary strict total order on the vertices. Then define subgraphs $D_1 = (\mathcal{V}, \mathcal{E}_1)$ and $D_2 = (\mathcal{V}, \mathcal{E}_2)$ where $\mathcal{E}_1 = \{(u, v) \mid u < v\} \cap \mathcal{E}$ and $\mathcal{E}_2 = \{(u, v) \mid u > v\} \cap \mathcal{E}$. Observe that D_1 and D_2 are directed acyclic graphs, and thus Corollary 2 implies that for $j \in \{1, 2\}$ there exists a threshold T_j, dimension $d_j = \mathcal{O}((\Delta + 2) \log |\mathcal{V}|)$, and mapping $\phi_j : \mathcal{V} \to \mathbb{R}^{2d_j}$ such that $\mathrm{E}_{\text{G-Box}}(u, v; \phi, (\tau, \nu)) < T_j \iff (u, v) \in \mathcal{E}_j$. Let $d = d_1 + d_2$, $T = \min(T_1, T_2)$, and define $\phi : V \to \mathbb{R}^{2d}$ and $B \in [0, 1]^{d \times k}$ as follows:

$$\begin{aligned} \forall i \in \{1, \ldots, d_1\}, \quad \phi(u)_i^\pm = \phi_1(u)_i^\pm, \quad B_{i,1} = \tfrac{T}{T_1}, \quad B_{i,2} = 0, \\ \forall i \in \{d_1 + 1, \ldots, d_1 + d_2\}, \quad \phi(u)_i^\pm = \phi_2(u)_i^\pm, \quad B_{i,1} = 0, \quad B_{i,2} = \tfrac{T}{T_2}. \end{aligned}$$

Then we have $\mathrm{E}_{\text{GBC-Box}}(u, v; (\phi, B), (\tau, \nu)) = \min_{j \in \{1,2\}} \frac{T}{T_j} \mathrm{E}_{\text{G-Box}}(u, v; \phi_j, (\tau, \nu))$ which completes the proof with $k = 2$, and therefore implies the result for all $k > 2$. □

While motivated by a similar idea, note that VBC-BOX is not a generalization of GBC-BOX, and thus an independent proof of representational capacity is required.

Theorem 6. *Given a directed graph $G = (\mathcal{V}, \mathcal{E})$ and any $\tau, \nu \geq 0$, there exists a threshold $T > 0$ and parameters $\phi : \mathcal{V} \to \mathbb{R}^{2d} \times [0, 1]^d$ for which $\mathrm{E}_{\text{VBC-Box}}(u, v; \phi, (\tau, \nu)) < T$ if and only if $(u, v) \in \mathcal{E}$.*

Proof. Given a graph $G = (\mathcal{V}, \mathcal{E})$ let $H = \{\{u, v\} \mid u, v \in \mathcal{V}, u \neq v\}$. We will construct a VBC-BOX model in $d = |H|$ dimensions. For convenience, index the dimensions using $h \in H$. Then let $\phi(u)_h^\diamond = 1$ if $u \in h$, and 0 otherwise. This means when evaluating the edge (u, v) or (v, u) we simply need to compare in the 1-d space obtained by projecting the boxes to dimension $h = \{u, v\}$, and furthermore that this dimension will not be used when considering any other edges. Any directed graph on 2 nodes can be embedded using boxes in 1-dimension (observable by direct construction), which completes the proof. □

While Theorem 5 and Theorem 6 are helpful in establishing that, unlike all prior box embedding models and many other geometric embeddings, GBC-BOX and VBC-BOX are capable of modeling any directed graph, the implied dimensionality bounds are far from optimal. In general, both models tend to require fewer dimensions than alternatives, which we analyze theoretically in Appendix D and observe empirically in Section 5.

4.4 Learning

We fit geometric embeddings by optimizing a binary cross entropy objective. Given some edges in a training set $\mathcal{T}$, the loss is defined as

$$L_{BCE}(\phi; \boldsymbol{\lambda}) = \sum_{(u,v)\in\mathcal{T}} \left[\mathrm{E}(u, v; \phi, \boldsymbol{\lambda}) - \sum_{(u',v')\in N(u,v)} \log\left(1 - e^{-\,\mathrm{E}(u',v';\phi,\boldsymbol{\lambda})}\right) \right] \tag{10}$$

where $N(u, v)$ is the set of negative samples for each positive edge (u, v) within one batch. We sample minibatches of positive edges in $\mathcal{T}$, and for each positive edge we sample 32 edges not in $\mathcal{T}$ by randomly corrupting either the source or target node. We also use a self-adversarial negative weight, as described in [26].

4.5 Limitations and Regularization

There are a few limitations of Binary Code Box Embeddings: 1) **Transitivity and Flexibility** In order to model cycles with separate sub-spaces, the inductive bias of asymmetric transitivity might be weakened. 2) **Inefficiency of Parameterization** For VBC-Box, the number of parameters is increased by 50% compared with a G-Box model in equal dimension. In addition, during inference, a large portion of box co-ordinates are "dead" when binary codes are near zero. Here, we introduce a regularization method and a tunable binary code size to resolve these concerns.

Regularization We can regularize the sparsity of binary codes to penalize dimension drop-off using the lasso with a regularization weight w_r, leading to a loss function $L = L_{BCE} + w_r * \|1 - B\|_{\ell^1}$.

Restricted Binary Code Size We can constrain the number of trainable binary codes to the last d_{bin} dimensions, setting the first $d - d_{\text{bin}}$ dimensions to 1.

Both w_r and d_{bin} can provide a handle on mitigating issues mentioned above. One can increase w_r or decrease d_{bin} to preserve more transitivity and make full use of the model's parameters.

5 Experiments

5.1 Graph Reconstruction

While GBC-Box and VBC-Box can provably model any directed graph, the extent to which they can be trained to do so via gradient descent is another matter. In this section, we compare the reconstruction performance of various geometric embedding methods on a number of synthetic graphs, including simple directed cycles, trees and scale-free networks.

Baselines We compare our model with different baselines, including: *Vector**: We implement a vector baseline where each node is parameterized by a source and target vector, and the energy function is measured by $\mathrm{E}(u, v) = -\log\sigma(\phi(u)_{\text{source}} \cdot \phi(v)_{\text{target}})$. * indicates it uses source and target vectors. *Lorentzian*: It has been shown that hyperbolic space can embed undirected trees with arbitrarily low distortion [22], therefore we also compare with the baseline of squared Lorentzian distance on the hyperboloid [10, 2]. *Hyperbolic Entailment Cones* [7] model vertices as cones in hyperbolic space, combining the bias of hyperbolic space to represent tree-like graphs with the transitivity bias of region-based representations. We also compare with G-Box and T-Box, as defined in Section 2.2. We use Bayesian hyperparameter tuning based on the optimal F1 score for reconstruction.

Capacity over Cycles We evaluate each model's capacity to represent cycles, where simple directed cycles are generated with an increasing number of vertices ($|\mathcal{V}| = 2^2$, 2^4, 2^8, 2^{12}). In addition, we analyze the effect of adding one reverse edge to the cycle, where standard box embedding can model it (though high precision is required). Results are shown in Table 1. For a fair comparison, all methods use 12 parameters per vertex [6]. VBC-Box shows the best reconstruction performance. Most other geometric baselines cannot model cycles. Surprisingly, VBC-Box even outperforms the *Vector* baseline when $|\mathcal{V}| = 2^{12}$, indicating our model's high expressivity and surprising ease of training. Results also show that, in concordance with Proposition 1, G-Box can model a cycle containing a reverse edge when $|\mathcal{V}|$ is small. We also see that T-Box can model certain cycles when $|\mathcal{V}|$ is small, which is not surprising as Theorem 3 does not apply to T-Box.

[6] 12 is the least common multiple of 2, 3, 4 which are the minimum number of parameters per node for G-Box, t-box and VBC-Box

Table 1: Reconstruction performance (F1 score) on directed cycles. All embeddings use 12 parameters per vertex. Different columns show results as we increase the number of vertices in the cycle.

Methods	**Simple cycle**				**+ One bidirectional edge**			
$\|\mathcal{V}\| =$	2^2	2^4	2^8	2^{12}	2^2	2^4	2^8	2^{12}
Vector*	**1.0**	**1.0**	**1.0**	0.676	**1.0**	**1.0**	**1.0**	0.666
Lorentzian	0.857	0.75	0.679	0.671	**1.0**	0.839	0.693	0.665
Hyperbolic Entailment Cones	0.75	0.667	0.662	0.635	0.75	0.692	0.654	0.645
G-Box	0.857	0.762	0.695	0.648	**1.0**	0.914	0.689	0.630
t-Box	**1.0**	**1.0**	0.996	0.685	**1.0**	**1.0**	0.992	0.659
GBC-Box	**1.0**	**1.0**	0.992	0.957	**1.0**	**1.0**	0.993	0.967
VBC-Box ($d_{bin} = d$)	**1.0**	**1.0**	**1.0**	**0.973**	**1.0**	**1.0**	**1.0**	**0.978**

Capacity over Trees It is known that Box space can naturally capture asymmetric transitivity, and hyperbolic space is suitable for un-directed trees. In this experiment, we evaluate whether BC-Box maintains the inductive bias of Box Embeddings. Therefore, we generated four balanced (out-)trees, each has 2^{13} vertices and the number of branches are choosing from [2, 3, 5, 10]. And we also generated another four graphs with full transitive closures. We compare each methods with 12, 24 and 48 parameters per vertex. All results are average performances over four graphs. The results are shown in Table 2. As expected, G-Box performs the best on transitively closed trees, while Lorentzian distance embedding performs well on balanced trees. It is shown that, GBC-Box performs equally well as G-Box on transitively closed trees, while the performance of VBC-Box is slightly lower. In contrast, the latter performs similarly or better than former in the transitive reduction trees. This suggests that more transitive bias is preserved in GBC-Box, while per vertex binary codes provide more representational flexibility. In addition, we get similar observation as [2], where Binary Code Boxes and t-Box outperform Lorentzian model on balanced trees when dimension size is increased, while performances of other geometric-based embeddings saturate on larger dimension sizes.

Table 2: Average reconstruction performance (F1 score) on balanced trees with $|\mathcal{V}| = 2^{13}$ and branching factors of $2, 3, 5, 10$ using $12, 24$, and 48 parameters per vertex.

Methods	**Balanced tree**			**w. transitive closures**		
# Parameters / vertex =	12	24	48	12	24	48
Vector*	0.453	0.992	**1.0**	0.863	0.999	**1.0**
Lorentzian	**0.929**	0.935	0.951	0.975	0.979	0.995
Hyperbolic Entailment Cones	0.828	0.834	0.838	0.977	0.982	0.987
G-Box	0.832	0.830	0.842	**1.0**	**1.0**	**1.0**
t-Box	0.800	0.957	**1.0**	0.952	0.997	**1.0**
GBC-Box	0.901	0.961	0.983	**1.0**	**1.0**	**1.0**
VBC-Box ($d_{bin} = d$)	0.866	**0.994**	**1.0**	0.987	0.999	**1.0**

Capacity over Random Graphs Finally, we conduct experiments on scale-free networks, a simulation to real-world graphs, where the edge distribution follows preferential attachment. In order to analyze how the cyclicity of graphs affects each model's performance, we randomly sampled nearly three-thousand graphs using a wide range of parameters used for graph generation. Then we split the generated graphs into five bins by our proposed measure *cyclicity*: The proportion of vertices in a given graph involved in at least one cycle. In order to analyze models' effectiveness of modeling cycles instead of the density of graphs, we randomly sample 5 graphs from each bin where the average degree is in the range between 3 and 4. Results are shown in Figure 1. From the chart, we can see that our proposed model outperforms standard G-Box in all scenarios. And there exists significant gap when cyclicity is high. It also clearly shows that VBC-Box provides more representational capacity overall, while GBC-Box is less expressive in modeling cyclic graphs.

5.2 Link Prediction

We apply Binary Code Box on link prediction tasks to evaluate model's generalization ability. We employ following real world graphs: Google hyperlink network, Epinions trust network, CORA

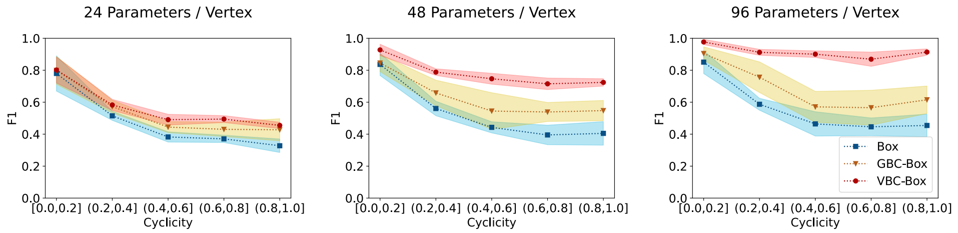


Figure 1: Reconstruction performance on Scale-free networks with $|\mathcal{V}| = 2^{13}$. We plot the F1 scores for G-Box, GBC-Box and VBC-Box using 24, 48, 96 parameters per vertex.

citation network, Twitter network, and DREAM5 gene regulatory networks. For more data statistics, see Appendix E. During training, all hyper-parameters are tuned via 10% cross-validation and then the test set results are averaged over 10 models of different random seeds, which trained on the full training set with the best hyper-parameters found during corss validation. We also tune w_r for all BC-Box models on the link prediction task.

In Table 3, we follow [30] and compare with several baselines including vector based baselines such as DeepWalk [17], LINE [27], node2vec [8], HOPE [15], APP [29], DGGAN [30] and our implementation of vector-based model with source and target parameters, and region based baselines such as G-Box [3] and t-Box [2]. Models are evaluated using the Area Under ROC Curve (AUC). For fair comparison, we following [30] and use 128 parameters per vertex for all our models. Results show that our Binary Code Box models out-performs other baselines in most cases. And there is an clear trend that VBC-Box performs the best when graphs are highly cyclic (on the left side of the table), then VBC-Box with less binary code dimensions and GBC-Box model start to perform better when graphs are less cyclic. In the case where a graph is mostly acyclic (on the right side of the table), G-Box performs equally well. We can see that box geometry is superior than vector inner-product models in all scenario, even if graphs are less transitive, or has a lot of cycles, showing the strength of box geometry in modeling directed graphs. In addition, we also compared with another recent work from Sim et al. [23] over DREAM5 datasets, where we observe that Box embedding-based model out-performs their baselines significantly in most cases (On *In Silico* dataset, our model has an average precision of 68.8%, out-performs their best result 61.0%). Detailed results are in Appendix F.

Table 3: **Link prediction on real-world graphs** We use AUC as evaluation metric. Vector-based methods (upper), and box embedding variants (bottom). We evaluated over two negative sampling strategies for testing, *unif.*: uniformly sampled negatives; *corr.*: randomly corrupting source or target node in each positive edge in the test set. All methods use 128 parameters per vertex. Bold numbers perform the best, and underscored numbers perform the best in all non-box models

Methods	**Google**		**Epinions**		**CORA**		**Twitter**	
Cyclicity =	*0.96*		*0.88*		*0.23*		*0.01*	
Transitivity =	*0.40*		*0.09*		*0.22*		*0.01*	
	unif.	corr.	unif.	corr.	unif.	corr.	unif.	corr.
DeepWalk 2014	83.6	-	76.6	-	84.9	-	50.4	-
LINE-1 2015	89.7	-	78.8	-	84.7	-	53.1	-
node2vec 2016	84.3	-	89.7	-	85.3	-	50.6	-
HOPE* 2016	87.5	-	79.6	-	77.6	-	98.0	-
APP* 2017	92.1	-	70.5	-	76.6	-	71.6	-
DGGAN 2021	92.3	-	96.1	-	85.1	-	99.7	-
Vector(Ours)*	94.0	94.2	93.0	88.9	78.9	76.7	99.8	84.1
G-Box	99.2	98.2	95.1	90.0	93.9	89.6	99.8	**86.3**
t-Box	97.1	95.8	96.4	89.6	87.3	79.6	99.8	84.4
GBC-Box	99.0	98.3	96.8	**92.7**	92.7	**90.3**	**99.9**	86.2
VBC-Box ($0 < d_{bin} < d$)	99.3	98.3	97.6	92.2	**94.1**	89.7	**99.9**	86.1
VBC-Box ($d_{bin} = d$)	**99.5**	**98.6**	**98.0**	92.4	93.2	88.8	99.8	85.7

6 Conclusion

In this paper, we introduced binary code box embeddings, a generalized box embedding method to model directed graphs. We provide both theoretical and empirical results showing the capacity of our model for modeling directed graphs. We demonstrated that this model can maintain a useful bias of transitivity while also modeling graphs with cycles.

Acknowledgements

This work supported in part by the Center for Data Science and in part the Center for Intelligent Information Retrieval, and in part by the National Science Foundation under Grants No. 1763618, University of Southern California subcontract no. 123875727 under Office of Naval Research prime contract no. N660011924032, and in part by IBM Research AI through AI Horizons Network, and in part by the Chan Zuckerberg Initiative under the project "Scientific Knowledge Base Construction". The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Some of the work reported here was performed using high performance computing equipment obtained under a grant from the Collaborative R&D Fund managed by the Massachusetts Technology Collaborative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

[1] Robi Bhattacharjee and Sanjoy Dasgupta. What relations are reliably embeddable in euclidean space? *Conference on Algorithmic Learning Theory (ALT)*, 2020.

[2] Michael Boratko, Dongxu Zhang, Nicholas Monath, Luke Vilnis, Kenneth Clarkson, and Andrew McCallum. Capacity and bias of learned geometric embeddings for directed graphs. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] Shib Sankar Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Lorraine Li, and Andrew McCallum. Improving local identifiability in probabilistic box embeddings. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[4] Shib Sankar Dasgupta, Xiang Lorraine Li, Michael Boratko, Dongxu Zhang, and Andrew McCallum. Box-to-box transformations for modeling joint hierarchies. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 277–288, 2021.

[5] Munmun De Choudhury, Yu-Ru Lin, Hari Sundaram, Kasim Selcuk Candan, Lexing Xie, and Aisling Kelliher. How does the data sampling strategy impact the discovery of information diffusion in social media? In *Fourth international AAAI conference on weblogs and social media*, 2010.

[6] Leo Egghe and Ronald Rousseau. *Introduction to informetrics: Quantitative methods in library, documentation and information science*. Elsevier Science Publishers, 1990.

[7] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. *arXiv preprint arXiv:1804.01882*, 2018.

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

[9] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. *International Conference on Learning Representations (ICLR)*, 2019.

[10] Marc Law, Renjie Liao, Jake Snell, and Richard Zemel. Lorentzian distance learning for hyperbolic representations. *International Conference on Machine Learning (ICML)*, 2019.

[11] Xiang Li, Luke Vilnis, Dongxu Zhang, Michael Boratko, and Andrew McCallum. Smoothing the geometry of probabilistic box embeddings. *International Conference on Learning Representations (ICLR)*, 2019.

[12] Daniel Marbach, James C Costello, Robert Küffner, Nicole M Vega, Robert J Prill, Diogo M Camacho, Kyle R Allison, Manolis Kellis, James J Collins, and Gustavo Stolovitzky. Wisdom of crowds for robust gene network inference. *Nature methods*, 9(8):796–804, 2012.

[13] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. *International Conference on Machine Learning (ICML)*, 2018.

[14] Evelien Otte and Ronald Rousseau. Social network analysis: a powerful strategy, also for the information sciences. *Journal of information Science*, 28(6):441–453, 2002.

[15] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114, 2016.

[16] Gergely Palla, Illés J Farkas, Péter Pollner, Imre Derényi, and Tamás Vicsek. Directed network modules. *New journal of physics*, 9(6):186, 2007.

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

[18] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 459–467, 2018.

[19] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. Trust management for the semantic web. In *International semantic Web conference*, pages 351–368. Springer, 2003.

[20] Fred S Roberts. On the boxicity and cubicity of a graph. *Recent progress in combinatorics*, 1969.

[21] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Re. Representation tradeoffs for hyperbolic embeddings. *International Conference on Machine Learning (ICML)*, 2018.

[22] Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. *International Symposium on Graph Drawing*, pages 355–366, 2011.

[23] Aaron Sim, Maciej L Wiatrak, Angus Brayne, Páidí Creed, and Saee Paliwal. Directed graph embeddings in pseudo-riemannian manifolds. In *International Conference on Machine Learning*, pages 9681–9690. PMLR, 2021.

[24] Lovro Šubelj and Marko Bajec. Model of complex networks based on citation dynamics. In *Proceedings of the 22nd international conference on World Wide Web*, pages 527–530, 2013.

[25] Jiankai Sun, Bortik Bandyopadhyay, Armin Bashizade, Jiongqian Liang, P Sadayappan, and Srinivasan Parthasarathy. Atp: Directed graph embedding with asymmetric transitivity preservation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 265–272, 2019.

[26] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *International Conference on Learning Representations (ICLR)*, 2019.

[27] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

[28] Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. Probabilistic embedding of knowledge graphs with box lattice measures. *Association for Computational Linguistics (ACL)*, 2018.

[29] Chang Zhou, Yuqiong Liu, Xiaofei Liu, Zhongyi Liu, and Jun Gao. Scalable graph embedding for asymmetric proximity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[30] Shijie Zhu, Jianxin Li, Hao Peng, Senzhang Wang, and Lifang He. Adversarial directed graph embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4741–4748, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work does not introduce any particular new potential for negative societal impacts, but it provides inductive biases (transitivity / cyclicity) that could have negative downstream use cases as features for models, in search (for example). However, we provide handles to control and trade-off such bias, therefore downstream users should be aware and control the bias as necessary.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] We include complete proofs in Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] We obtained code and data for real world graphs that was publicly available.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The data we use does not have any personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Equivalence of G-Box and Dasgupta et al. [4]

In our notation, the model in Dasgupta et al. [4] would have score function

$$F_{\text{D-Box}}((x^-, x^+), (y^-, y^+); (\tau, \nu)) := \frac{\operatorname{LSE}_\nu\left(-\operatorname{LSE}_\tau(-x^+, -y^+) - \operatorname{LSE}_\tau(x^-, y^-) - 2\nu\gamma, 0\right)}{\operatorname{LSE}_\nu(y^+ - y^- - 2\nu\gamma, 0)}$$

where γ is the Euler-Mascheroni constant.

As presented in Dasgupta et al. [4], D-Box used a single temperature $\beta = \tau = \nu$, and derived this score function as an approximation to a ratio of expected volumes of intervals whose endpoints were modeled by Gumbel random variables. Gumbel random variables are typically parameterized by a *location* and *scale*, and Dasgupta et al. [4] interpreted the parameters x^-, y^- (resp. x^+, y^+) as the location parameters for $\operatorname{GumbelMax}$ (resp. $\operatorname{GumbelMin}$) distributions with scale β.

Remark 1. *Although the model was proposed and analyzed in Dasgupta et al. [4] using a single temperature parameter β, the authors do use separate τ and ν parameters when implementing the model, and so we adopt that formulation when defining $F_{\text{D-Box}}$ above.*

We claim D-Box and G-Box are equivalent, in the following sense.

Proposition 2. *Given any $\nu \geq 0$ there exists a bijection ψ on the set of functions $\{V \to \mathbb{R}^{2d}\}$ such that*

$$E_{\text{G-Box}}(u, v; \psi(\phi), (\tau, \nu)) = E_{\text{D-Box}}(u, v; \phi, (\tau, \nu)), \tag{11}$$

and, being a bijection,

$$E_{\text{G-Box}}(u, v; \phi, (\tau, \nu)) = E_{\text{D-Box}}(u, v; \psi^{-1}(\phi), (\tau, \nu)). \tag{12}$$

Proof. Observe that for any $a, b, c \in \mathbb{R}$ and $t \geq 0$,

$$\operatorname{LSE}_t(a + c, b + c) = \operatorname{LSE}_t(a, b) + c.$$

Then

$$\begin{aligned}
&F_{\text{G-Box}}((x^- + \nu\gamma, x^+ - \nu\gamma), (y^- + \nu\gamma, y^+ - \nu\gamma); (\tau, \nu)) \\
&\quad = \frac{\operatorname{LSE}_\nu\left(-\operatorname{LSE}_\tau(-x^+ + \nu\gamma, -y^+ + \nu\gamma) - \operatorname{LSE}_\tau(x^- + \nu\gamma, y^- + \nu\gamma), 0\right)}{\operatorname{LSE}_\nu(y^+ - \nu\gamma - y^- - \nu\gamma, 0)} \\
&\quad = \frac{\operatorname{LSE}_\nu\left(-\operatorname{LSE}_\tau(-x^+, -y^+) - \operatorname{LSE}_\tau(x^-, y^-) - 2\nu\gamma, 0\right)}{\operatorname{LSE}_\nu(y^+ - y^- - 2\nu\gamma, 0)} \\
&\quad = F_{\text{D-Box}}((x^-, x^+), (y^-, y^+); (\tau, \nu)).
\end{aligned}$$

Therefore, as introduced in Section 2, if we label the output of ϕ using d pairs $\phi(u)_i = (\phi(u)_i^-, \phi(u)_i^+)$ and define $\psi(\phi)$ to be a mapping from $V \to \mathbb{R}^{2d}$ such that

$$\psi(\phi)(u)_i = (\phi(u)_i^- + \nu\gamma,\ \phi(u)_i^+ - \nu\gamma), \tag{13}$$

the calculations above prove (11), and the proof of (12) is similar. □

Remark 2. *Note that the mean of $X \sim \operatorname{GumbelMax}(\mu, \beta)$ is $\mu + \beta\gamma$, and similarly the mean of $X \sim \operatorname{GumbelMin}(\mu, \beta)$ is $\mu - \beta\gamma$. As mentioned above, in the setting where $\beta = \tau = \nu$ the parameters of* D-Box *can be interpreted as the location parameters for Gumbel distributions, and thus* (13) *simply takes the location parameters to their mean. Hence, in the case where $\tau = \nu$, the* G-Box *model can simply be interpreted as using the mean of the Gumbel distributions as opposed to the location parameter. This leads to a slight simplification in the score function by removing the $2\nu\gamma$, which has minor computational and mathematical benefits. The more conceptual benefit, however, is that it unifies* Box, G-Box, *and* T-Box.

B Representational Capacity of G-Box

In this section we prove any graph capable of being represented by Box is also capable of being represented by G-Box, regardless of the temperature hyperparameters. The proof involves two components:

1. The representational capacity of G-Box depends not on the absolute values of τ and ν, but rather their ratio.
2. The energy of Box can be approximated by G-Box using small enough τ and ν.

Proposition 3. *Let $\tau_1, \nu_1 > 0$ and $\phi : V \to \mathbb{R}^{2d}$ be given. Then for any $\tau_2, \nu_2 > 0$ such that $\frac{\tau_2}{\nu_2} = \frac{\tau_1}{\nu_1}$ the function $\psi(\phi) : V \to \mathbb{R}^{2d}$ for which $\psi(\phi)(u)_i^{\pm} = \frac{\nu_2}{\nu_1}\phi(u)_i^{\pm}$ is such that*

$$\mathrm{E}_{\text{G-Box}}(u, v; \phi, (\tau_1, \nu_1)) = \mathrm{E}_{\text{G-Box}}(u, v; \psi(\phi), (\tau_2, \nu_2)). \tag{14}$$

Proof. The proof is by direct calculation. First, note that for any $t, c > 0$ we have for any vector $\mathbf{x} \in \mathbb{R}^n$

$$\mathrm{LSE}_t(c\mathbf{x}) = t \log \left(\sum_{i=1}^{n} \exp\left(\tfrac{cx_i}{t}\right) \right) = c(t/c) \log \left(\sum_{i=1}^{n} \exp\left(\tfrac{x_i}{t/c}\right) \right) = c \, \mathrm{LSE}_{t/c}(\mathbf{x}).$$

In particular, for $c = \frac{\nu_2}{\nu_1} = \frac{\tau_2}{\tau_1}$ (where the latter equality follows from the premise of the proposition) we have for any $a, b \in \mathbb{R}$

$$\mathrm{LSE}_{\tau_2}(ca, cb) = c \, \mathrm{LSE}_{\tau_1}(a, b) \quad \text{and} \quad \mathrm{LSE}_{\nu_2}(ca - cb, 0) = c \, \mathrm{LSE}_{\nu_1}(a - b, 0).$$

Thus

$$\begin{aligned}
&F_{\text{G-Box}}((cx^-, cx^+), (cy^-, cy^+); (\tau_2, \nu_2)) \\
&= \frac{\mathrm{LSE}_{\nu_2}\left(-\mathrm{LSE}_{\tau_2}(-cx^+, -cy^+) - \mathrm{LSE}_{\tau_2}(cx^-, cy^-), 0\right)}{\mathrm{LSE}_{\nu_2}(cy^+ - cy^-, 0)} \\
&= \frac{\mathrm{LSE}_{\nu_2}\left(-c\,\mathrm{LSE}_{\tau_1}(-x^+, -y^+) - c\,\mathrm{LSE}_{\tau_1}(x^-, y^-), 0\right)}{\mathrm{LSE}_{\nu_2}(cy^+ - cy^-, 0)} \\
&= \frac{c\,\mathrm{LSE}_{\nu_1}(-\mathrm{LSE}_{\tau_1}(-x^+, -y^+) - \mathrm{LSE}_{\tau_1}(x^-, y^-), 0)}{c\,\mathrm{LSE}_{\nu_1}(y^+ - y^-, 0)} \\
&= F_{\text{G-Box}}((x^-, x^+), (y^-, y^+); (\tau_1, \nu_1)),
\end{aligned}$$

which proves (14).

The following lemma will be helpful in proving the next part.

Lemma 1. *For all $y > 0$, given $\varepsilon > 0$ and some $M \in \mathbb{R}$, there exists $\delta > 0$ such that for all $0 < \nu < \delta$, for all $x < M$ we have*

$$\left| \frac{\mathrm{LSE}_\nu(x, 0)}{\mathrm{LSE}_\nu(y, 0)} - \frac{\max(x, 0)}{y} \right| < \varepsilon.$$

Proof. Note that $\mathrm{LSE}_\nu(x, 0)$ is monotonically increasing in x for any $\nu \geq 0$, and is always greater than $\max(x, 0)$. Furthermore,

$$|\,\mathrm{LSE}_\nu(x, 0) - \max(x, 0)| = \mathrm{LSE}_\nu(x, 0) - \max(x, 0) \leq \nu \log 2$$

as it obtains it's maximum when $x = 0$ (which can be observed by inspection of the signs of the derivative). Then for all $x < M$ we have

$$\begin{aligned}
\left| \frac{\mathrm{LSE}_\nu(x, 0)}{\mathrm{LSE}_\nu(y, 0)} - \frac{\max(x, 0)}{y} \right| &< \left| \frac{\mathrm{LSE}_\nu(x, 0)}{\mathrm{LSE}_\nu(y, 0)} - \frac{\mathrm{LSE}_\nu(x, 0)}{y} \right| + \left| \frac{\mathrm{LSE}_\nu(x, 0)}{y} - \frac{\max(x, 0)}{y} \right|. \\
&< \mathrm{LSE}_\nu(M, 0) \left| \frac{1}{\mathrm{LSE}_\nu(y, 0)} - \frac{1}{y} \right| + \frac{\nu \log 2}{y}.
\end{aligned} \tag{15}$$

Now (15) does not depend on x, and tends to 0 as $\nu \to 0$, which completes the proof. □

Proposition 4. *Given a mapping $\phi : V \to \mathbb{R}^{2d}$ where $\phi(u)_i^+ > \phi(u)_i^-$ for each $u \in V$, $i \in \{1, \ldots, d\}$, we have that for all $u, v \in V$,*

$$\lim_{(\tau,\nu)\to(0,0)} \mathrm{E}_{\text{G-Box}}(u, v; \phi, (\tau, \nu)) = \mathrm{E}_{\text{Box}}(u, v; \phi). \tag{16}$$

Proof. Given fixed $x^- < x^+, y^- < y^+$, let $f(\tau) = -\operatorname{LSE}_\tau(-x^+, -y^+) - \operatorname{LSE}(x^-, y^-)$, and $z = \min(x^+, y^+) - \max(x^-, y^-) = \lim_{\tau\to 0} f(\tau)$. Then

$$\begin{aligned}
&\left|F_{\text{G-Box}}((x^-, x^+), (y^-, y^+); (\tau, \nu)) - F_{\text{Box}}((x^-, x^+), (y^-, y^+))\right| \\
&= \left|\frac{\operatorname{LSE}_\nu(f(\tau), 0)}{\operatorname{LSE}_\nu(y^+ - y^-, 0)} - \frac{\max(z, 0)}{y^+ - y^-}\right| \\
&< \left|\frac{\operatorname{LSE}_\nu(f(\tau), 0)}{\operatorname{LSE}_\nu(y^+ - y^-, 0)} - \frac{\max(f(\tau), 0)}{y^+ - y^-}\right| + \left|\frac{\max(f(\tau), 0)}{y^+ - y^-} - \frac{\max(z, 0)}{y^+ - y^-}\right|. \qquad (17)
\end{aligned}$$

Given $\varepsilon > 0$, choose δ_1 such that $0 < \tau < \delta_1$ implies the second summand in (17) is bounded by $\varepsilon/2$. Then $f(\tau)$ is bounded, and we can apply Lemma 1 to choose δ_2 such that $0 < \nu < \delta_2$ implies the first summand is less than $\varepsilon/2$. Thus taking $\delta = \min(\delta_1, \delta_2)$ completes the proof on the level of the per-dimension score functions, and thus (16) follows by continuity. □

We are now ready to prove the main theorem.

□

Theorem 1. *Given a threshold T, temperature hyperparameters τ, ν, there exists and a bijection ψ on the set of parameterizations $\{\mathcal{V} \to \mathbb{R}^{2d}\}$ such that for all $u, v \in \mathcal{V}$,*

$$\mathrm{E}_{\text{G-Box}}(u, v; \psi(\phi), (\tau, \nu)) < T \quad\iff\quad \mathrm{E}_{\text{Box}}(u, v; \phi) < T.$$

Proof. Let $\varepsilon > 0$ be a number we will specify later. Then by Proposition 4, for each $(u, v) \in V^2$ we have some $\delta_{(u,v)} > 0$ such that

$$\tau, \nu \in (0, \delta_{(u,v)}) \quad\implies\quad |\,\mathrm{E}_{\text{G-Box}}(u, v; \phi, \tau, \nu) - \mathrm{E}_{\text{Box}}(u, v; \phi)| < \varepsilon. \qquad (18)$$

Let

$$\delta = \min_{(u,v)\in V^2} \delta_{(u,v)}, \quad \tau' = \frac{\delta\tau}{2\max(\tau, \nu)}, \quad \nu' = \frac{\delta\nu}{2\max(\tau, \nu)}.$$

Since $\frac{\tau'}{\nu'} = \frac{\tau}{\nu}$ we can apply Proposition 3, which guarantees the existence of a function ψ such that

$$\mathrm{E}_{\text{G-Box}}(u, v; \psi(\phi), (\tau', \nu')) = \mathrm{E}_{\text{Box}}(u, v; \phi, (\tau, \nu)).$$

Noting that $\tau', \nu' \in (0, \delta)$, we can combine this with (18), and find

$$\mathrm{E}_{\text{Box}}(u, v; \phi) - \varepsilon < \mathrm{E}_{\text{G-Box}}(u, v; \psi(\phi), (\tau, \nu)) < \mathrm{E}_{\text{Box}}(u, v; \phi) + \varepsilon.$$

Let

$$T_1 = \max_{(u,v)\in\mathcal{E}} \mathrm{E}_{\text{Box}}(u, v; \phi), \quad T_2 = \min_{(u,v)\notin\mathcal{E}} \mathrm{E}_{\text{Box}}(u, v; \phi),$$

and set $\varepsilon = \min(T - T_1, T_2 - T)$. Then if $(u, v) \in \mathcal{E}$ we have

$$\mathrm{E}_{\text{G-Box}}(u, v; \psi(\phi), (\tau, \nu)) < T_1 + T - T_1 = T,$$

and if $(u, v) \notin \mathcal{E}$ we have

$$\mathrm{E}_{\text{G-Box}}(u, v; \psi(\phi), (\tau, \nu)) > T_2 - (T_2 - T) = T,$$

which completes the proof. □

C Representing Cycles with Box Embeddings

Proposition 1 *If G is a graph which is the union of a chordless cycle and one reverse edge,* G-BOX *can model G in 2 dimensions.*

Proof. Given a $G = \{\mathcal{V}, \mathcal{E}\}$, $\mathcal{E} = \{(1, 2), (2, 3), (3, 4), (4, 5), , (N - 1, N), (N, 1)\} \cup \{(1, N)\}$ When $N = 2$, it is trivial since two boxes can be equal. When $N > 2$ and N is an even number (odd number), we can construct 2-d BOX in Figure 2 left (right). Let the area of $\phi(i)$ be $V(\phi(i))$, $i \le N$, the area of intersection box between two nodes that are connected $\delta_i = V(\phi(i) \cap \phi(i + 1))$, $i \le N - 1$, and let $V(\phi(N) \cap \phi(1)) = C\delta_{N-1}$ where $C > 1$. It can be observed from Figure 2

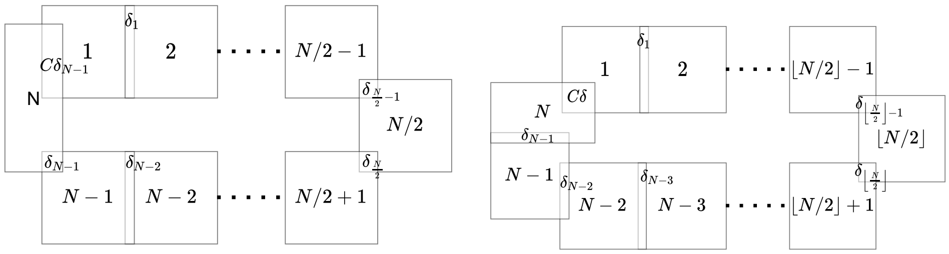


Figure 2: Visualization of 2D box to represent a graph with chordless cycles and one reverse edge. Diagram on the left is when there are even number of nodes in the graph, while the one on the right is when the number is odd.

that there exists arrangement of ϕ, threshold T and a large enough α where $0 < \alpha < 1$, such that for any $N > 2$, $V(\phi(i+1)) = \alpha * V(\phi(i))$, $\delta_{i+1} = \alpha * \delta_i$, $i \leq N-1$ and $C = \frac{1}{\alpha^{N-1}}$. Then, there exists t, for any pair of node $(i, i+1) \in G$, $E(i, i+1) = E(N, 1) = -log\frac{\delta_1}{\phi(1)\alpha} < T$, $E(1, N) = -log\frac{\delta_1}{\alpha^N \phi(1)} < T$, and for $(u, v) \notin \mathcal{E}$, $E(u, v) \geq T$. Given Appendix B, this proof also applies to G-BOX. □

Proposition 2 *Given box embeddings that model a directed graph $G \in (\mathcal{V}, \mathcal{E})$ which is the union of a chordless cycle and one reverse edge. Let $P(u, v) = e^{-E(u,v;\phi,\lambda)} = \prod_{i=1}^{d} F(\phi(u)_i, \phi(v)_i; \boldsymbol{\lambda})$, $P(u) = \prod_{i=1}^{d} \max(\phi(u)_i^+ - \phi(u)_i^-, 0)$, $\mu = e^{-T}$ and $\gamma = \min_{(u,v)\in\mathcal{E}} P(u, v) - \max_{(u',v')\notin\mathcal{E}} P(u', v')$. Then, γ and μ need to satisfy:*

$$\gamma < \mu(1 - \mu^{\frac{1}{N-1}})$$

Proof. Given $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{E} = \{(1, 2), (2, 3), ..., (N-1, N), (N, 1), (1, N)\}$ and $N = |\mathcal{V}|$. According to Proposition 1, that there exist G-BOX that can reconstruct G with threshold $\mu = e^{-T}$: a directed edge (u, v) exists iff $P(u, v) > \mu$ and (u, v) does not exist iff $P(u, v) \leq \mu - \gamma$. γ is a positive margin. We would like to understand the relationship between N and γ, where a smaller γ will be more sensitive to perturbation on the box parameters.

If $(u, v) \in \mathcal{E}$ and $(v, u) \notin \mathcal{E}$, we have $P(v)\mu < P(u, v) \leq P(u)(\mu - \gamma)$. Thus $P(u)/P(v) > \mu/(\mu - \gamma)$. By extending this inequality from $u = 1$,$v = 2$ to $u = N-1$, $v = N$, we can derive a lower bound of marginal ratio between $P(N)$ and $P(1)$:

$$P(1)/P(N) > \mu^{N-1}/(\mu - \gamma)^{N-1} \tag{19}$$

Since $(N, 1) \in \mathcal{E}$, we have

$$P(1, N)/\mu > P(1) \tag{20}$$

Given Eq. 19 and Eq. 20, we have

$$P(1, N)/P(N) > \mu^N/(\mu - \gamma)^{N-1} \tag{21}$$

Since $P(1, N)/P(N) \leq 1$, $\mu^N < (\mu - \gamma)^{N-1}$, therefore

$$\gamma < \mu(1 - \mu^{\frac{1}{N-1}}) \tag{22}$$

Since $\gamma > 0$, according to sandwich theorem, Eq. 22 indicates that γ will be close to zero when $N \to +\infty$. □

D Dimensionality Bounds for Binary Code Models

We start by proving the following lemma, which implies that any directed graph on 2 nodes can be embedded using boxes in 1-dimension with any threshold $T > 0$.

Lemma 2. *Given $T > 0$, there exist $x^-, x^+, y^-, y^+ \in \mathbb{R}$ such that*

$$-\log F_{\text{Box}}((x^-, x^+), (y^-, y^+)) < T < -\log F_{\text{Box}}((y^-, y^+), (x^-, x^+)).$$

Proof. The proof is a direct construction; take

$$y^- = x^- = 0, \quad x^+ = 1, \quad \text{and} \quad y^+ = e^{-2T}.$$

Then

$$\max(\min(1, e^{-2T}) - \max(0, 0), 0) = e^{-2T},$$

and thus

$$F_{\text{Box}}((0, 1), (0, e^{-T/2})) = 1, \quad \text{and} \quad F_{\text{Box}}((0, e^{-2T}), (0, 1)) = e^{-2T}$$

which implies the desired result. □

We then strengthen the statement of Theorem 6 to apply to an arbitrary threshold.

Theorem 7. *Given a directed graph $G = (\mathcal{V}, \mathcal{E})$, any temperatures $\tau, \nu \geq 0$, and any threshold $T > 0$, there exists a parameterization $\phi : \mathcal{V} \to \mathbb{R}^{2d} \times [0, 1]^d$ with $d = \mathcal{O}(|\mathcal{V}|^2)$ for which*

$$\mathrm{E}_{\text{VBC-Box}}(u, v; \phi, (\tau, \nu)) < T \quad \iff \quad (u, v) \in \mathcal{E}.$$

Proof. Given a graph $G = (\mathcal{V}, \mathcal{E})$ let $H = \{\{u, v\} \mid u, v \in \mathcal{V}, u \neq v\}$. We will construct a VBC-Box model in $d = |H|$ dimensions. For convenience, index the dimensions using $h \in H$, and let $\phi(u)^{\diamond}_{h} = 0$ if $u \notin h$. Thus when evaluating edge (u, v) or (v, u), dimension $h = \{u, v\}$ is the only dimension whose score may not be equal to 1.

Lemma 2 implies any graph on 2 nodes can be embedded using threshold T in one dimension, and Theorem 1 implies this is also true for G-Box for any setting of temperatures, completing the proof. □

Theorem 8. *Let $G = (\mathcal{V}, \mathcal{E})$, and let $\tau, \nu \geq 0$ be given temperature hyperparameters. Let $\mathcal{V}_F$ be the minimum feedback vertex set, $\mathcal{E}_F = \mathcal{E} \cap \mathcal{V}_F^2$, and $G_F = (\mathcal{V}_F, \mathcal{E}_F)$. Then for any temperatures τ, ν there exists a threshold $T > 0$ and a parameterization $\phi : \mathcal{V} \to \mathbb{R}^{2d} \times [0, 1]^d$ such that*

$$\mathrm{E}_{\text{VBC-Box}}(u, v; \phi, (\tau, \nu)) < T \quad \iff \quad (u, v) \in \mathcal{E},$$

where $d = \mathcal{O}((\Delta_F + 2)\log(|\mathcal{V}_F|) + |\mathcal{V}_C|^2)$, with Δ_F the maximum degree of G_F, and

$$\mathcal{V}_C = \{u \mid (u, v) \in \mathcal{E}, u \notin \mathcal{V}_F \text{ or } v \notin \mathcal{V}_F\}.$$

Proof. Theorem 2 implies that G_F can be embedded using G-Box with the given temperature hyperparameters τ, ν in dimension at most $d_F = \mathcal{O}((\Delta_F + 2)\log|\mathcal{V}_F|)$. Let $\phi_F : \mathcal{V}_F \to \mathbb{R}^{2d_F}$ be the parameterization for this embedding, and T the threshold on the energy. Now let

$$\mathcal{E}_{\neg F} = \{(u, v) \in \mathcal{V}^2 \mid u \notin \mathcal{V}_F \quad \text{or} \quad v \notin \mathcal{V}_F\},$$

and define $\mathcal{E}_1 = \mathcal{E}_F \cup \mathcal{E}_{\neg F}$ and $G_1 = (\mathcal{V}, \mathcal{E}_1)$. We can extend ϕ_F to a VBC-Box parameterization $\phi_1 : \mathcal{V} \to \mathbb{R}^{2d_F} \times [0, 1]^{2d_F}$ as follows:

$$\phi_1(u)_i = \begin{cases} (\phi_F(u)_i^-, \phi_F(u)_i^+, 1) & \text{if} \quad u \in \mathcal{V}_F, \\ (0, 1, 0) & \text{otherwise.} \end{cases}$$

This parameterization is such that

$$\mathrm{E}_{\text{VBC-Box}}(u, v; \phi_F, (\tau, \nu)) \begin{cases} = 0 & \text{if} \quad (u, v) \in \mathcal{E}_{\neg F}, \\ < T & \text{if} \quad (u, v) \in \mathcal{E}_F, \\ > T & \text{otherwise.} \end{cases} \tag{23}$$

Now let $\mathcal{E}_C = \mathcal{E} \cap \mathcal{E}_{\neg F}$, and note that these edges are only between nodes in $\mathcal{V}_C$. Define $G_C = (\mathcal{V}_C, \mathcal{E}_C)$, which, by Theorem 7, can be embedded with threshold T in $d_C = \mathcal{O}(|\mathcal{V}_C|^2)$ dimensions. Let $\phi_C : \mathcal{V}_C \to \mathbb{R}^{2d_C}$ be the associated parameterization.

Now let $G_2 = (\mathcal{V}, \mathcal{E}_C \cup (\mathcal{V}_F \times \mathcal{V}_F))$, then extend ϕ_C to a parameterization $\phi_2 : \mathcal{V} \to \mathbb{R}^{2d_C}$ as follows:

$$\phi_2(u)_i = \begin{cases} (\phi_C(u)_i^-, \phi_C(u)_i^+, 1) & \text{if} \quad u \in \mathcal{V}_C, \\ (0, 1, 0) & \text{otherwise.} \end{cases}$$

For this parameterization, we have

$$\mathrm{E}_{\text{VBC-Box}}(u, v; \phi_2, (\tau, \nu)) \begin{cases} = 0 & \text{if} \quad (u, v) \in \mathcal{V}_F \times \mathcal{V}_F, \\ < T & \text{if} \quad (u, v) \in \mathcal{E}_C, \\ > T & \text{otherwise.} \end{cases} \tag{24}$$

The desired VBC-Box embedding ϕ for G with $d = d_F + d_C = \mathcal{O}((\Delta_F + 2)\log(|\mathcal{V}_F|) + |\mathcal{V}_C|^2)$ dimensions now be created by concatenating ϕ_1 and ϕ_2, for which

$$\mathrm{E}_{\text{VBC-Box}}(u, v; \phi, (\tau, \nu)) = \mathrm{E}_{\text{VBC-Box}}(u, v; \phi_1, (\tau, \nu)) + \mathrm{E}_{\text{VBC-Box}}(u, v; \phi_2, (\tau, \nu)). \tag{25}$$

Since $\mathcal{E}_F \cap \mathcal{E}_C = \emptyset$, inspecting (23) and (24) we have that

$$\mathrm{E}_{\text{VBC-Box}}(u, v; \phi(\tau, \nu)) < \mathcal{T} \quad \iff \quad (u, v) \in (\mathcal{E}_C \cap \mathcal{E}_{\neg F}) \cup (\mathcal{E}_F \cap (\mathcal{V}_F \times \mathcal{V}_F)) = \mathcal{E}.$$

□

E Data Statistics

Google [16] (15,763 nodes and 171,206 edges) is a hyperlink network from pages within Google's sites. Nodes represent pages and directed edges represent hyperlink between pages. **Epinions** [19] (75,879 nodes and 508,837 edges) is a trust network from the online social network Epinions. Nodes represent users and directed edges represent trust between users. **CORA** [24] (23,166 nodes and 91,500 edges) is a citation network of academic papers. Nodes represent papers and directed edges represent the citation relationships between papers. **Twitter** [5] (465,017 nodes and 834,797 edges) is a social network. Nodes represent users and directed edges represent following relationships between users. **DREAM5** [12] is a gene regulatory networks across organisms. *In silico* network has 1,565 nodes and 4,012 edges. *E. Coli* network has 1,081 nodes and 2,066 edges. *S. cerevisiae* network has 1,994 nodes and 3,940 edges. These networks contain a relatively small number number of cycles.

F Link Prediction on DREAM5 Datasets

In Table 4, we compared with a recent work that embedding graphs into Pseudo-Riemannian manifolds [23], along with other baselines such as Euclidean and Hyperboloid embeddings on DREAM5 datasets. Models are evaluated using Average Precision (AP). Results show that Binary Code Box significantly out-performs baseline methods on *In Silico* and *S. Cerevisiae* datasets, while show competitive performance on the *E. Coli* graph. It can also be observed that Gumbel Box performs competitively on these graphs, which are almost acyclic.

Table 4: **Link prediction on Experiments on DREAM5 Datasets**. Following Sim et al. [23], we use median Average Precision among 5 test sets with different negative samples as evaluation metric, and we sample 4 times the negatives by randomly corrupting one of the node in each positive edges in the test set. We compare our model with other baselines using 10, 50, 100 number of parameters per vertex. Cyclicity and transitivity of each graph are shown in the table. Bold numbers perform the best, and underscored numbers perform the best in all non-box models. For more details of baselines models, please refer to [23].

Methods	**In Silico**			**E. Coli**			**S. Cerevisiae**		
Cyclicity =	*0.01*			*0.01*			*0.01*		
Transitivity =	*0.25*			*0.40*			*0.17*		
# parameters / vertex	10	50	100	10	50	100	10	50	100
Euclidean + FD	39.7	39.8	34.8	40.2	44.5	49.0	40.2	44.5	49.0
Hyperboloid + FD	50.8	50.9	52.5	52.7	53.6	50.6	46.5	48.8	47.9
Minkowski + TFD	51.2	57.7	58.0	$\underline{63.4}$	$\underline{67.7}$	$\underline{\mathbf{68.2}}$	46.4	52.7	54.0
Anti de-Sitter + TFD	51.9	55.6	56.0	61.8	63.3	63.0	44.9	47.5	49.4
Cylindrical Minkowski + TFD	56.3	58.9	$\underline{61.0}$	62.3	65.8	63.2	46.8	53.4	54.6
Vector(Ours)*	$\underline{56.7}$	$\underline{59.2}$	59.8	56.0	58.1	59.6	$\underline{51.2}$	$\underline{55.2}$	$\underline{55.2}$
G-Box	**62.3**	66.1	66.6	65.1	66.5	68.0	55.0	58.6	59.5
GBC-Box	62.0	66.4	**68.8**	**65.4**	**68.3**	65.9	55.1	59.4	**59.7**
VBC-Box ($d_{bin} < d$)	58.4	**66.5**	66.4	62.6	67.3	66.1	52.1	**59.6**	58.3
VBC-Box ($d_{bin} = d$)	55.3	66.0	64.9	58.1	65.8	65.3	**55.3**	57.5	57.7

G Hyper-parameter Search

We follow the setting from Boratko et al. [2] for hyper-parameter search strategies in graph reconstruction experiments. Table 5 shows ranges of Bayesian hyper-parameter search for our link prediction experiments.

Table 5: Hyper-parameter range of Bayesian optimization for link prediction.

Hyper-Parameter	Range
learning rate	1e-5 $\sim$ 1e-2
batch size	1024 (Table 3), 64 (Table 4)
max epochs	16, 32, 64, 128
τ	0.001 $\sim$ 0.1
ν	0.1 $\sim$ 10.0
w_r	$10^{-8} \sim 1.0$
k	$1 \sim 10$
d_{bin}	$0 \sim d$

H Case Study

In this section, we visualize how binary codes work to preserve transitivity and cyclicity together.

As shown in Figure 3, our analysis is over two synthetic graphs. For fair comparison, we embed both graphs into 3 dimensional G-Box and 2 dimensional VBC-Box. The graph on the top is formed by a 7-node directed chain (vertex 0 to vertex 6) with full transitive closures and one additional node connecting the chain into a cycle. It shows that Gumbel Box can model transitive closures well, but cannot model the cycle while Binary Code Box can handle both. The latter models transitive closures by sharing full box space from node 0 to node 5, and then models the cycle by selecting sub-dimensions in node 6 and 7. The graph on the bottom is formed by a chain of triangle cycles. It shows that Gumbel Box cannot handle cycles, and generates an acyclic graph. In contrast, binary code boxes can handle this graph nicely with much lower errors by alternately switching among sub-spaces within each triangle cycles.

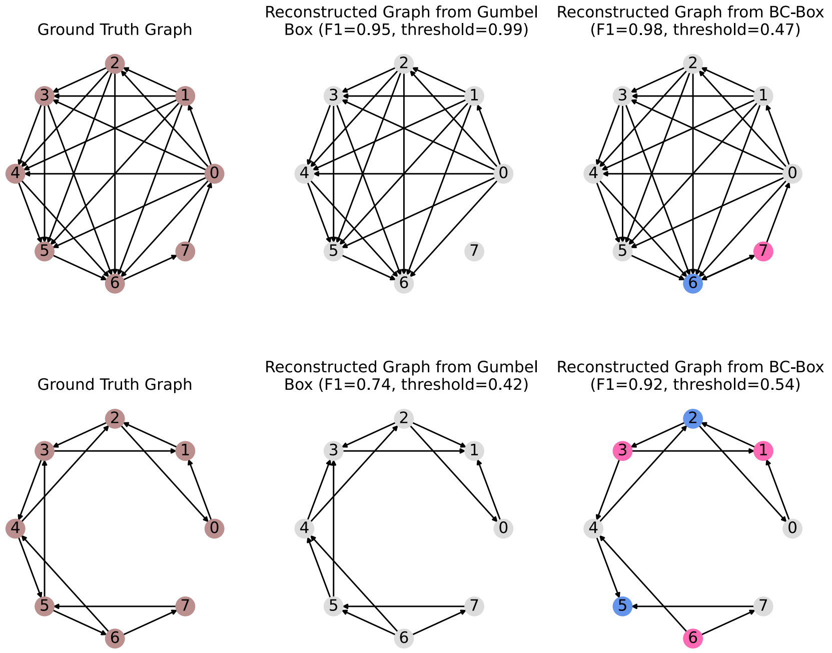
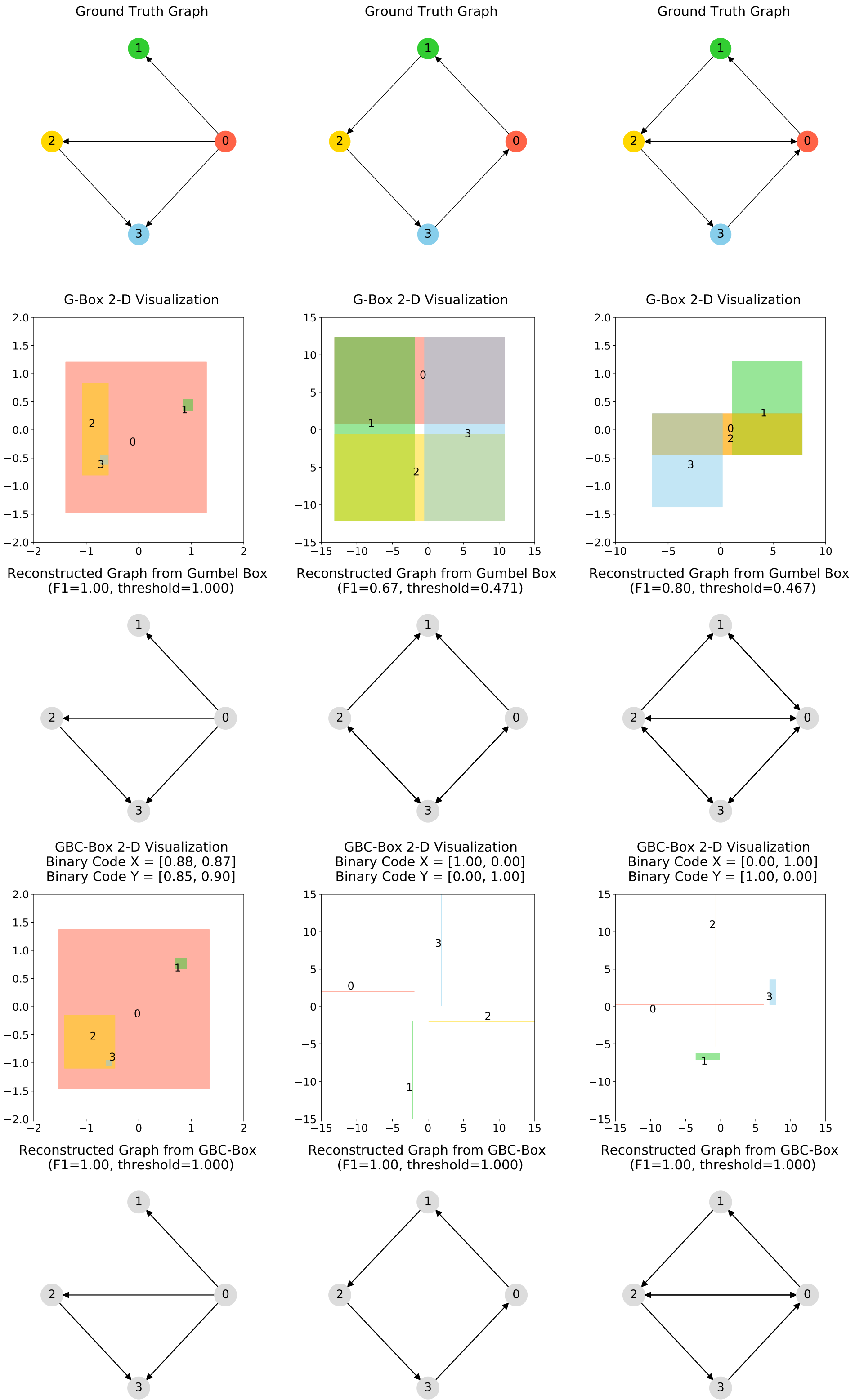

Ground Truth Graph
Reconstructed Graph from Gumbel Box (F1=0.95, threshold=0.99)
Reconstructed Graph from BC-Box (F1=0.98, threshold=0.47)
Ground Truth Graph
Reconstructed Graph from Gumbel Box (F1=0.74, threshold=0.42)
Reconstructed Graph from BC-Box (F1=0.92, threshold=0.54)
0
1
2
3
4
5
6
7

Figure 3: Two directed graphs and reconstructed graphs by G-Box (3D) and VBC-Box (2D). The upper graph is a directed cycle with almost all transitive closures. The bottom graph is a chain of directed triangle cycles. Blue colored vertices have binary codes of [1,0], pink colored vertices have binary codes of [0,1], and grey colored vertices have binary codes of [1, 1].

In Fig. 4, we visualize how GBC-Box handles cycles and transitive edges. We compare GBC-Box with 2 dimensional G-Box. Fig. 4a shows that when representing a DAG, GBC-Box learns to utilize all dimensions in both binary code vectors and leverages box containment to model edge directions. Fig. 4b shows that given a pure cycle, GBC-Box learns "skinny" boxes to model $0 \to 1, 2 \to 3$ using the vertical axis, and the rest of the edges in the horizontal axis. Fig. 4c shows a more complicated graph and our model also learns to split the graph into $0 \to 1, 1 \to 2, 0 \to 2$ in the horizontal axis, and $2 \to 3, 3 \to 0, 2 \to 0$ in the vertical axis. In comparison, the original G-Box struggles with cycles and cannot reconstruct ground truth graphs from 4b and 4c [7].

[7] The 0th and 2nd boxes cover almost same regions in the second row of Figure 4c



(a) (b) (c)

Figure 4: Visualization of graph reconstruction using 2 dimensional G-Box and GBC-Box. In figure 4a, the ground truth graph is a DAG (an out-tree with transitive closure). In figure 4b, we have a pure cycle. In figure 4c, we have three cycles nested together, forming two 2-hop transitive closures. We visualized 2-D boxes trained via gradient descent and binary cross entropy loss with $\tau = 0.001$ and $\nu = 0.5$, and learning rate 0.01.